



Universidad
Nacional
de Loja

Universidad Nacional de Loja

**Facultad de la Energía, Las Industrias y los Recursos Naturales
No Renovables**

Carrera de Computación

**Integración de reconocimiento facial y sistema de Control de Acceso para el
registro de asistencia en los laboratorios de software de la Carrera de
Computación de la Universidad Nacional de Loja**

**Integration of facial recognition and Access Control system
for the registration of attendance in the software laboratories
of the Computing Career of the National University of Loja**

**Trabajo de integración Curricular,
previo a la obtención del título en
Ciencias de la Computación.**

AUTOR:

Luis Alfredo Jumbo Herrera

DIRECTOR:

Ing. Edison Leonardo Coronel Romero, Mg.Sc.

Loja – Ecuador

2025

Certificación

Loja, 20 de enero del 2025

Ing. Edison Leonardo Coronel Romero, Mg.Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Integración Curricular denominado: **Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja**, previo a la obtención del título de **Ingeniero en Ciencias de la Computación**, de la autoría del estudiante **Luis Alfredo Jumbo Herrera**, con **cédula de identidad Nro. 1105946014**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.



Firmado electrónicamente por:
EDISON LEONARDO
CORONEL ROMERO

Ing. Edison Leonardo Coronel Romero, Mg.Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Yo, **Luis Alfredo Jumbo Herrera**, declaro ser autor del presente Trabajo de Integración Curricular y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular, en el Repositorio Digital Institucional – Biblioteca Virtual.



Firmado electrónicamente por:
**LUIS ALFREDO JUMBO
HERRERA**

Firma

Cedula de identidad: 1105946014

Fecha: 20 de enero de 2025

Correo electrónico: luis.a.jumbo.h@unl.edu.ec

Teléfono: 0989349047

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo del Trabajo de Integración Curricular.

Yo, **Luis Alfredo Jumbo Herrera**, declaro ser el autor del Trabajo de Integración Curricular denominado: **Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Integración Curricular que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veinte días del mes de enero de dos mil veinticinco.



Firma

Autor: Luis Alfredo Jumbo Herrera

Cédula de identidad: 1105946014

Dirección: Loja, Aztecas y Av. Eugenio Espejo

Correo electrónico: luis.a.jumbo.h@unl.edu.ec

Teléfono: 0989349047

DATOS COMPLEMENTARIOS:

Director del Trabajo de Integración Curricular: Ing. Edison Leonardo Coronel Romero, Mg.Sc.

Dedicatoria

Dedico este trabajo a mis queridos y amados padres, Luis Alberto Jumbo Sánchez y Sandra Guadalupe Herrera Herrera, por su apoyo incondicional y todo su amor brindado. Por todo su sacrificio realizado para poder darme una educación y una calidad de vida digna, sacrificando su propio bienestar por el mío y el de nuestro hogar. Son mi principal fuente de motivación y modelo a seguir en la vida. Son una parte fundamental para poder culminar con mi etapa universitaria y poder seguir mi camino hacia lo que quiero ser y lograr

A mi querido hermano, el cual mediante sus enseñanzas y valores inculcados desde que era pequeño, me han permitido desenvolverme en la vida de la mejor manera posible. El cariño y respeto que tengo por él será un sentimiento que lo llevaré conmigo por el resto de mi vida.

A mis sobrinas por brindarme esa tranquilidad y amor que me permite seguir adelante y afrontar los problemas que se me presenten.

A mi familia en general, los cuales han estado para mí cuando lo he necesitado, dándome apoyo y ánimos para poder superar cualquier dificultad que se me presente.

A mis amigos, los cuales han representado un sustento anímico y emocional muy grande durante mi etapa universitaria, brindándome consejos, apoyo, y momentos divertidos que han permitido forjar vínculos que perdurarán por el resto de mi vida.

Reitero mis más sinceros y profundos agradecimientos a todas aquellas personas que me han brindado su apoyo y ayuda cuando lo he necesitado. Son una parte fundamental en mi desarrollo como personal y profesional.

Luis Alfredo Jumbo Herrera

Agradecimiento

Expreso mis más sinceros agradecimientos a la Universidad Nacional de Loja, y, sobre todo, a su prestigiosa Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables. Donde tanto su personal administrativo, como sus docentes, han sido fundamentales en mi desarrollo humano y profesional, brindándome una guía con su profesionalismo y calidad humana en mi carrera universitaria.

Mis profundos agradecimientos al Ing. Edison Leonardo Coronel Romero, Mg.Sc. por su tiempo, dedicación, y orientación que fueron esenciales en el desarrollo del presente Trabajo de Integración Curricular.

Agradezco a mi familia y amigos, los cuales, gracias a su constante presencia y apoyo incondicional, fueron un pilar fundamental para poder culminar mi etapa de vida universitaria.

Luis Alfredo Jumbo Herrera

Índice de contenidos

Portada	i
Certificación	ii
Autoría.....	iii
Carta de autorización.....	iv
Dedicatoria	v
Agradecimiento.....	vi
Índice de contenidos	vii
Índice de tablas:	x
Índice de figuras:	xi
Índice de anexos:.....	xiii
1 Título	1
2 Resumen	2
3 Introducción	4
4 Marco Teórico.....	7
4.1 Antecedentes.....	7
4.1.1 Situación actual.....	7
4.1.2 Muestra.....	7
4.1.3 Laboratorio de software.....	8
4.2 Fundamentación teórica.....	8
4.2.1 Integración de software.....	8
4.3 Metodologías	9
4.3.1 Metodología XP	9
4.3.2 Metodología TAM.....	10
4.4 Tecnologías	12
4.4.1 Python.....	12
4.4.2 ODOO	13
4.4.3 PostgreSQL	14

4.5	Trabajos relacionados	15
5	Metodología	17
5.1	Área de Estudio.....	17
5.2	Procedimiento	18
5.2.1	Objetivo Específico 1. Desarrollar un software que permita integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso para el registro de asistencias en los laboratorios de software en Python, mediante la metodología XP	18
5.2.2	Objetivo Específico 2. Determinar el nivel de aceptación tecnológica del software para el registro de asistencia en los laboratorios de software mediante el modelo TAM	20
5.3	Recursos.....	21
5.3.1	Recursos científicos	21
5.3.2	Recursos técnicos	22
5.4	Participantes.....	24
6	Resultados.....	25
6.1	Objetivo Específico 1. Desarrollar un software que permita integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso para el registro de asistencias en los laboratorios de software en Python, mediante la metodología XP	25
6.1.1	Fase de planificación	25
6.1.2	Fase de diseño	38
6.1.3	Fase de codificación	45
6.1.4	Fase de pruebas.....	53
6.1.5	Fase de lanzamiento	56
6.2	Objetivo Específico 2. Determinar el nivel de aceptación tecnológica del software para el registro de asistencia en los laboratorios de software mediante el modelo TAM	60
6.2.1	Fase de elaboración	60
6.2.2	Fase de medición.....	62

6.2.3 Fase de análisis.....	63
7 Discusión	71
7.1 Objetivo Específico 1. Desarrollar un software que permita integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso para el registro de asistencias en los laboratorios de software en Python, mediante la metodología XP	71
7.2 Objetivo Específico 2. Determinar el nivel de aceptación tecnológica del software para el registro de asistencia en los laboratorios de software mediante el modelo TAM	73
8 Conclusiones	76
9 Recomendaciones	78
9.1 Trabajos futuros.....	78
10 Bibliografía	80
11 Anexos	84

Índice de tablas:

Tabla 1. Trabajos relacionados.....	15
Tabla 2. Informe de análisis a la observación participante efectuada al encargado de los laboratorios de software de la carrera de Computación	26
Tabla 3. Informe de análisis a la entrevista efectuada al director de la carrera de Computación	27
Tabla 4. Informe de revisión de documentos, para el TIC, orientado al control de acceso a los laboratorios de software de la carrera de Computación de la UNL .	29
Tabla 5. Informe de revisión de documentos, para el TIC, orientado al registro local de asistencias mediante reconocimiento facial, a los laboratorios de software	30
Tabla 6. Descripción de las funcionalidades que pueden realizar los actores del sistema	32
Tabla 7. Requisitos funcionales del sistema.....	32
Tabla 8. Requisitos no funcionales del sistema	34
Tabla 9. Formato de historia de usuario.....	35
Tabla 10. Historia de usuario más relevante, denominada “Registrar asistencia Face ID”	36
Tabla 11. Plan de iteración.....	37
Tabla 12. Resumen general de las pruebas unitarias realizadas	54
Tabla 13. Resumen general de las pruebas de aceptación realizadas	55
Tabla 14. Cuestionario para evaluar el nivel de aceptación tecnológica (TAM)...	60
Tabla 15. Ponderaciones para cada respuesta del cuestionario	62
Tabla 16. Matriz de resultados en base a cada constructo.....	62

Índice de figuras:

Figura 1. Fases de la Metodología XP [17]	10
Figura 2. Modelo de Aceptación Tecnológica (TAM) [24]	12
Figura 3. Ubicación de Laboratorios de FEIRNNR de la UNL	18
Figura 4. Diagrama de despliegue.....	39
Figura 5. Diagrama de componentes	40
Figura 6. Diagrama de clases.....	41
Figura 7. Diagrama Modelo Relacional.....	42
Figura 8. Diagrama de casos de uso.....	43
Figura 9. Diagrama de actividades	44
Figura 10. Prototipo de Interfaz Gráfica más relevante del sistema	45
Figura 11. Prototipo de la estructura general del sistema.	46
Figura 12. Estructura general del sistema.	47
Figura 13 Esquema general de los modelos desarrollados.	48
Figura 14. Muestra del modelo desarrollado denominado “EquipoInformatico”.49	
Figura 15. Esquema general de las vistas desarrolladas.....	50
Figura 16. Muestra de la vista desarrollada denominada “EquipoInformatico”. 51	
Figura 17. Esquema general de las vistas desarrolladas.....	51
Figura 18. Muestra del controlador desarrollado denominado “Registro Asistencias Face Id” para “Validar Horario”......	52
Figura 19. Reporte de asistencias en la demostración en vivo.	53
Figura 20. Conexión a los servidores de la carrera de Computación mediante SSH.	56
Figura 21. Actualización del código fuente del módulo “control_acceso” en el servidor.	57
Figura 22. Actualización del código fuente del módulo “control_acceso” en el servidor.	57
Figura 23. Instalación de los requerimientos que necesita el sistema.....	57
Figura 24. Página de administración en Odoon de la carrera de Computación.....	58
Figura 25. Actualización de las aplicaciones en el servidor.	58
Figura 26. Actualización del módulo “control_acceso”.	59
Figura 27. Módulo “control_acceso” funcionando correctamente luego de la actualización.....	59

Figura 28. Resultados obtenidos por cada ítem del constructo de Utilidad Percibida (UP) de acuerdo a la escala de Likert	63
Figura 29. Resultados obtenidos por cada ítem del constructo de Facilidad de Uso Percibida (FUP) de acuerdo a la escala de Likert	65
Figura 30. Resultados obtenidos por cada ítem del constructo de Actitud hacia el uso (AU) de acuerdo a la escala de Likert.....	67
Figura 31. Resultados obtenidos por cada ítem del constructo de Intención de uso (IU) de acuerdo a la escala de Likert	69

Índice de anexos:

Anexo 1. Entrevista realizada al encargado de los laboratorios de software de la carrera de computación, Ing. Luis Sinche, para la elicitación de requisitos	84
Anexo 2. Entrevista realizada al director de la carrera de computación, Ing. Pablo Ordoñez, para la elicitación de requisitos	91
Anexo 3. Historias de Usuario	97
Anexo 4. Implementación de metodología XP	116
Anexo 5. Resultados de la aceptación tecnológica del sistema mediante TAM	198
Anexo 6. Manual de usuario para administrador	209
Anexo 7. Manual de usuario para docente.....	239
Anexo 8. Manual de usuario para estudiante	253
Anexo 9. Pruebas de validación operativas del sistema en un entorno controlado.....	267
Anexo 10. Pruebas de aceptación realizadas al director de la carrera de computación, Ing. Pablo Ordoñez	273
Anexo 11. Pruebas de aceptación realizadas al encargado de los laboratorios de software de la carrera de computación, Ing. Luis Sinche	275
Anexo 12. Reporte de plagio en la herramienta Copyleaks.....	277
Anexo 13. Documentos firmados electrónicamente	284
Anexo 14. Certificado de traducción del resumen.....	286

1 Título

Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja

Integration of facial recognition and Access Control system for the registration of attendance in the software laboratories of the Computing Career of the National University of Loja

2 Resumen

En la actualidad, para poder registrar asistencia en los laboratorios de software en la carrera de Computación de la Universidad Nacional de Loja (UNL), se lo realiza de manera manual, es decir, no existe una aplicación dedicada al manejo de asistencia en dichas instalaciones. Por tal motivo, el presente Trabajo de Integración Curricular (TIC) tiene como objetivo, desarrollar un sistema que integre el reconocimiento facial como método biométrico para el registro de asistencias en dichos laboratorios mediante la actualización del sistema actual de control de acceso. Para poder alcanzar el objetivo definido, se plantea dos objetivos específicos. El primero se centra en el desarrollo del sistema que integra el reconocimiento facial como método biométrico para el registro de asistencia. Esto, mediante la metodología de desarrollo XP, y las tecnologías de: Python, JavaScript, Odoo V17 y PostgreSQL. Luego del desarrollo, se implementó este software en el laboratorio de software de la carrera de Computación de la UNL. El segundo se centra en la evaluación de del sistema mediante el Modelo de Aceptación de Tecnología (TAM). Esto, mediante la evaluación de las cuatro variables recomendadas por el modelo a los usuarios que harán uso del software. Los resultados obtenidos muestran que la aceptación del sistema se sitúa entre medio y alto. Este nivel de aceptación confirma que la solución es una opción viable y eficiente para el control de asistencias en los laboratorios de la carrera de Computación de la UNL:

Palabras claves: Software, Metodología XP, TAM, Integración de Software, Odoo.

Abstract

Currently, attendance tracking in the software laboratories of the Computer Science degree program at the National University of Loja (UNL) is conducted manually, meaning there is no dedicated application for managing attendance in these facilities. For this reason, the present Curricular Integration Project (CIP) aims to develop a system that integrates facial recognition as a biometric method for attendance tracking by upgrading the current access control system. To achieve this goal, two specific objectives are defined. The first focuses on developing the system that integrates facial recognition as a biometric method for attendance tracking. This was accomplished using the XP development methodology and the following technologies: Python, JavaScript, Odoo V17, and PostgreSQL. After development, the software was implemented in the software laboratory of the Computer Science degree program at UNL. The second objective focuses on evaluating the system using the Technology Acceptance Model (TAM). This evaluation was conducted by assessing the four variables recommended by the model with the software's end users. The results indicate that the system's acceptance level ranges from medium to high. This level of acceptance confirms that the solution is a viable and efficient option for managing attendance control in the laboratories of the Computer Science degree program at UNL:

Keywords: Software, XP Methodology, TAM, Software Integration, Odoo.

3 Introducción

La integración se refiere al proceso de combinar diferentes elementos, sistemas o componentes para que funcionen juntos de manera armoniosa y eficiente. En el contexto de la tecnología y el software, la integración implica la unificación de diferentes sistemas o aplicaciones para que puedan intercambiar datos y trabajar de manera conjunta [10]. En el mundo de la computación, la integración de software se refiere al proceso de combinar diferentes sistemas o componentes de software para que funcionen juntos como un solo sistema. Implica diseñar, desarrollar y probar interfaces que permitan que los diferentes componentes de software se comuniquen e intercambien datos sin problemas. El objetivo de la integración de software es crear un sistema unificado que proporcione una funcionalidad, rendimiento y experiencia de usuario mejores que los componentes individuales [11]. Esto evidencia que integrar sistemas que funcionan por separado, pero que, poseen un fin común, puede ayudar a mejorar la experiencia de usuario, pues, se mejora considerablemente tanto la funcionalidad como el rendimiento de dichos sistemas.

Esto da paso a la problemática central identificada en la Universidad Nacional de Loja, puesto que, en la actualidad, para poder ingresar a los laboratorios de software, se hace uso del sistema de control de acceso, el cual controla qué usuarios pueden o no ingresar a los mismos. Esto, mediante el dispositivo RFID que permite la autenticación de los usuarios a través de una tarjeta, la misma que posee su respectivo registro en la base de datos de la carrera de computación, además, este dispositivo controla la temperatura de los usuarios, es decir, si el usuario presenta una temperatura elevada, no podrá ingresar a los laboratorios de software [1]. Por otro lado, existe el sistema de reconocimiento facial, el cual permite autenticar a un usuario mediante su modelo de detección de vida, lo que brinda una autenticación bastante segura [2]. No obstante, este sistema se encuentra ejecutándose en local (servidor local), por lo que no se encuentra integrado con el sistema de control de acceso, pero, que, de integrarse, ayudaría en gran medida a mantener el control del registro de asistencias de los usuarios que hagan uso de los laboratorios de software. En base a la situación problemática planteada anteriormente, se propone la siguiente pregunta de investigación: **¿Cuál es el nivel de aceptación tecnológica que se podría obtener al integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja, mediante un software que permita el registro de asistencia en los mismos?**

La problemática de integrar nuevas tecnologías más avanzadas al sistema de control de acceso actual de los laboratorios de software de la Universidad Nacional de Loja, es una

problemática importante y muy relevante, ya que brindaría muchos beneficios tanto a la comunidad universitaria, como a la sociedad en general que lleguen a hacer uso de estas instalaciones en un futuro. Esto, ya que posibilita tanto a los docentes como a los estudiantes, marcar asistencia de manera más rápida y eficiente, eliminando así los procesos manuales actuales que se emplean para el registro de dichas asistencias (hojas de cálculo de Excel y hojas de papel), reduciendo en gran medida los retrasos para poder dar inicio a las respectivas clases impartidas en estos laboratorios. Además, dicho sistema pretende mejorar de manera notable la experiencia de usuario de las personas que hagan uso de estas instalaciones, esto, mediante la medición de la aceptación tecnológica que dichos usuarios presenten al hacer uso del software a desarrollar.

Con el fin de responder la pregunta de investigación, se propuso el siguiente objetivo general: “Evaluar el nivel de aceptación tecnológica de un software que permita integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso para el registro de asistencias en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja”. Y para ejecutarlo, se desarrollaron dos objetivos específicos, los cuales son: “Desarrollar un software que permita integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso para el registro de asistencias en los laboratorios de software en Python, mediante la metodología XP”, abarcando todas las fases de dicha metodología, que compete las fases de: planificación, diseño, codificación, pruebas y lanzamiento [17]; y “Determinar el nivel de aceptación tecnológica del software para el registro de asistencia en los laboratorios de software mediante el modelo TAM”, siguiendo las fases propuestas por el modelo que abarcan las fases de: elaboración, medición y análisis, así como el enfoque a sus cuatro variables, las cuales son: utilidad percibida (UP), facilidad de uso percibida (FUP), actitud hacia el uso (AU), e intención de uso (IU) [24].

El alcance de este proyecto se enfoca en la integración y desarrollo de un sistema web que permita facilitar el proceso actual para el registro de asistencia dentro de los laboratorios de software. Dicho sistema será desplegado y evaluado en los laboratorios de software de la carrera de Computación de la Universidad Nacional de Loja. El software será integrado y desarrollado siguiendo la metodología XP, y se evaluará su aceptación de los usuarios mediante el Modelo de Aceptación Tecnológica (TAM). El área de estudio se focaliza en los estudiantes que actualmente hacen uso de los laboratorios de software, proporcionando así un sistema que les facilite considerablemente el registro de asistencia dentro de dichas instalaciones.

El presente TIC se compone por las siguientes secciones: en el **Marco Teórico** se encuentran los antecedentes, tecnologías y metodologías utilizadas, materiales, y trabajos

relacionados; en la **Metodología** está presente el área de estudio, el procedimiento seguido para ambos objetivos del TIC, los recursos científicos y técnicos necesarios para su ejecución, y los participantes; la sección de **Resultados** se divide en los dos objetivos que dictaminan el alcance del TIC; la **Discusión** que expone la interpretación de los resultados obtenidos; la sección de **Conclusiones** que aborda los hitos destacables del proyecto; y, finalmente, las **Recomendaciones** que detallan las sugerencias y trabajos futuros.

4 Marco Teórico

Esta sección presenta la base teórica del Trabajo de Integración Curricular (TIC) desarrollado, junto con una revisión bibliográfica del tema abordado. En la sección **4.1**, se analizan los antecedentes, es decir, la situación actual con respecto al laboratorio de software y los sistemas a integrar. La sección **4.2** profundiza en la fundamentación teórica, explicando los aspectos clave de la integración de software. En la sección **4.3**, se describen las metodologías utilizadas como guía para el desarrollo del proyecto. La sección **4.4** detalla las tecnologías seleccionadas para su implementación. Por último, la sección **4.5** examina los trabajos relacionados, aportando un contexto que resalta la importancia y relevancia del presente trabajo.

4.1 Antecedentes

4.1.1 Situación actual

En la actualidad, para poder ingresar a los laboratorios de software de la carrera de Computación de la Universidad Nacional de Loja, se hace uso del sistema de control de acceso, el cual controla qué usuarios pueden o no ingresar a los mismos. Esto, mediante el dispositivo RFID que permite la autenticación de los usuarios a través de una tarjeta, la misma que posee su respectivo registro en la base de datos de la carrera de computación, además, este dispositivo controla la temperatura de los usuarios, es decir, si el usuario presenta una temperatura elevada, no podrá ingresar a los laboratorios de software [1]. Por otro lado, existe el sistema de reconocimiento facial, el cual permite autenticar a un usuario mediante su modelo de detección de vida, lo que brinda una autenticación bastante segura [2]. No obstante, este sistema se encuentra ejecutándose en local (servidor local), por lo que no se encuentra integrado con el sistema de control de acceso, pero, que, de integrarse, ayudaría en gran medida a mantener el control del registro de asistencias de los usuarios que hagan uso de los laboratorios de software.

La problemática de integrar nuevas tecnologías más avanzadas al sistema de control de acceso actual de los laboratorios de software de la Universidad Nacional de Loja, es una problemática importante y muy relevante, ya que brindaría muchos beneficios tanto a la comunidad universitaria, como a la sociedad en general que lleguen a hacer uso de estas instalaciones en un futuro.

4.1.2 Muestra

Para el presente Proyecto de Integración Curricular, se empleará una muestra conformada por los cursos de Segundo "A", Tercero y Quinto de la carrera de computación de la

Universidad Nacional de Loja. Esto, ya que dichos estudiantes son los que hacen uso de los laboratorios de software que posee la universidad.

Esta muestra permitirá recopilar datos para la identificación y sustento de la situación problemática, las respectivas pruebas de aceptación una vez culminado el software, y, sobre todo, permitirán evaluar el nivel de aceptación tecnológica que estos tendrán con el software funcional (metodología TAM).

Además, cabe recalcar que, todo el proyecto se va a desarrollar en un ambiente controlado en los laboratorios de software, por lo que también se verán involucrados el personal administrativo que gestiona los mismos, como tal es el caso del encargado de estos laboratorios, el cual es el Ing. Luis Sinche

4.1.3 Laboratorio de software

En la actualidad, existen 27 equipos informáticos en el laboratorio de software de la carrera de Computación. Dicha instalación presta sus servicios a los estudiantes de la carrera, ofreciendo muchas herramientas tecnológicas tanto a nivel de software como de hardware que ayudan al aprendizaje práctico de los estudiantes. Debido a ello, la constante mejora, actualización, mantenimiento e implementación de nuevas herramientas tecnológicas es crucial para formar profesionales competentes en el mercado laboral y profesional. Los laboratorios son espacios compartidos esenciales que permiten a los estudiantes adquirir nuevos conocimientos, desarrollar actitudes y aptitudes positivas, y, fortalecen las habilidades obtenidas en la formación profesional de los mismos, aportando un valor agregado considerable a su perfil profesional [3]. Del mismo modo, [4] reafirma que todas las actividades realizadas dentro de un laboratorio de software, potencia considerablemente el perfil profesional de los estudiantes, dándoles la posibilidad de interactuar con nuevas tecnologías competentes dentro de equipos multidisciplinarios. Basándose en este enfoque descrito, es fundamental brindar nuevas y mejores tecnologías a estos espacios para que los estudiantes potencien aún más sus habilidades de trabajo en equipo, promoviendo una comunicación efectiva y crecimiento en sus conocimientos.

4.2 Fundamentación teórica

4.2.1 Integración de software

La integración de software es el proceso de unir varios sistemas, aplicaciones o módulos de software en un único entorno funcional con la idea o la certeza de que todas las partes funcionarán de forma coordinada [5]. Este proceso es crucial en proyectos de desarrollo, ya que

brinda la posibilidad de unir componentes que funcionan de manera independientes, y que estos puedan comunicarse e intercambiar datos, garantizando una funcionalidad completa del sistema [6].

La integración continua es una actividad común en los enfoques ágiles que integra los cambios en código mediante los pipelines y herramientas como Jenkins o GITLAB CI/CD [7]. Por otro lado, la integración vertical vincula módulos basados en procesos de trabajo específicos, mientras que la integración horizontal intenta fusionar subsistemas independientes a través de capas intermedias como APIs [8].

Algunos de los beneficios de integración adecuada son erradicación de errores y la minimización de la duración de los ciclos de efectividad en los procesos y el ambiente del usuario. No obstante, este proceso tiene sus inconvenientes que incluyen: problemas de compatibilidad entre sistemas, la dificultad técnica y los costos que puede imponer la configuración y manejo de la integración [9].

4.3 Metodologías

4.3.1 Metodología XP

La Programación Extrema (Extreme Programming, XP) es una metodología del desarrollo de software ágil cuyo objetivo principal es lograr una mayor calidad. Estuvo promovida por Kent Beck en la década de 1990 como resultado de los problemas asociados con las metodologías ortodoxas en entornos de alta incertidumbre [12]. XP favorece la comunicación continua entre usuarios desarrolladores y, también, ofrece prácticas que aseguran la adaptabilidad y la frecuencia de entregables en un software operativo [13].

Los cuatro valores de XP son la comunicación, la retroalimentación, la simplicidad del diseño, el valor y el respeto. Estos guían la puesta en práctica de aspectos como la programación en parejas, el desarrollo guiado por pruebas (TDD), la integración continua y la reestructuración. Estas prácticas no solo se encargan de aumentar la calidad de las soluciones, sino también disminuir el tiempo de entrega, así como el número de riesgos inherentes en el desarrollo [14].

XP presenta muchos beneficios significativos, entre los que se destacan: adaptabilidad al cambio, alta calidad de software, interacción constante con el cliente, consiguiendo una mayor satisfacción del mismo, entre otros. Sin embargo, también presenta varias limitaciones a considerar, como lo pueden ser: implementación compleja en equipos grandes, que los equipos no estén acostumbrados a trabajar con TDD o programación en parejas [15].

La metodología XP brinda la posibilidad de desarrollar sistemas que necesitan adaptarse rápidamente a cambios, presentando entregables en cada iteración, esto, conjuntamente con el uso de herramientas de versionado e integración continua como lo puede ser GitHub, GitLab, Bitbucket, etc., permiten a los equipos desarrollar sistemas aprovechando al máximo el potencial que puede ofrecer esta metodología ágil [16].

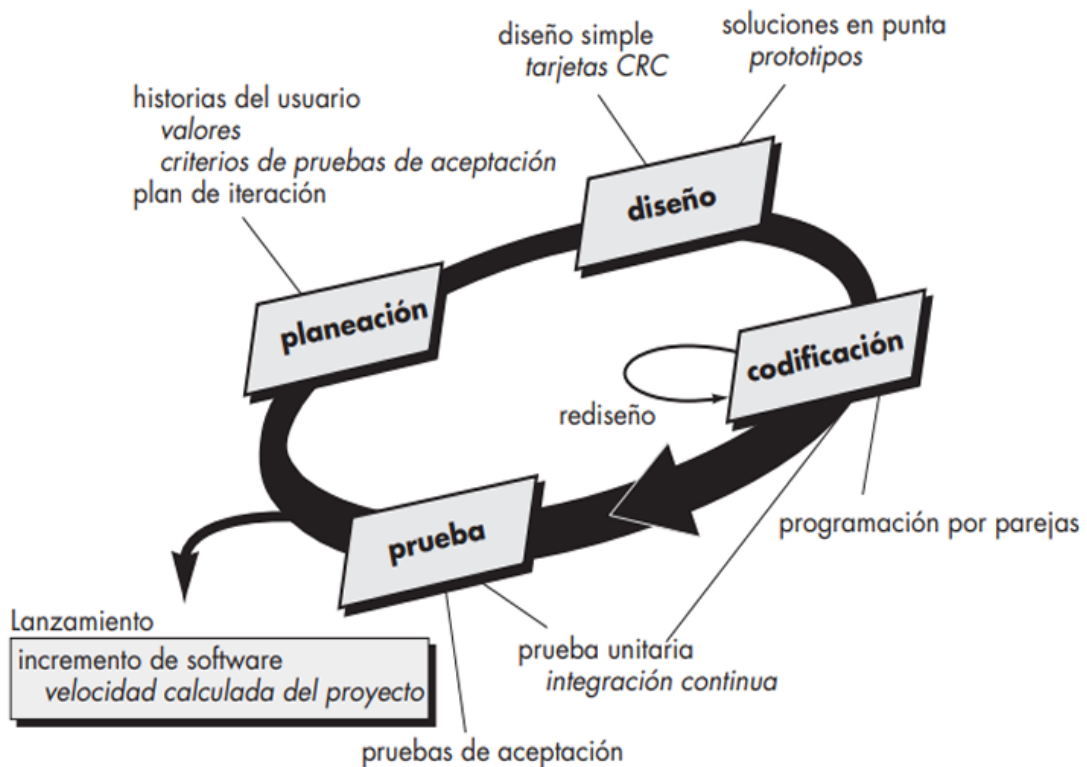


Figura 1: Fases de la Metodología XP [17]

Debido a que XP es una metodología ágil que admite un desarrollo de software iterativo e incremental, es la metodología idónea para ejecutar el presente Trabajo de Integración Curricular. Esto, ya que la metodología permite presentar entregables en cada iteración, que, conjuntamente con la ejecución de las fases propuestas por la metodología, se consigue un producto de alta calidad, garantizando el correcto funcionamiento del mismo.

4.3.2 Metodología TAM

El Modelo de Aceptación Tecnológica (TAM) es un marco teórico, que pretende determinar y predecir el nivel de aceptación y adopción de nuevas tecnologías en usuarios. Este modelo se basa en que mediante la percepción de utilidad (PU) y la percepción de facilidad de uso (PEU), se puede llegar a determinar de manera muy precisa y acertada la intención de uso de una tecnología en concreto [18].

Estas dos percepciones son el pilar para una correcta implementación del modelo. Respecto a la percepción de utilidad (PU), esta hace referencia a la medida cuantificada en que el hacer uso de una nueva tecnología mejorará considerablemente el rendimiento actual del usuario. Por otro lado, la percepción de facilidad de uso (PEU), hace referencia a la medida cuantificada del esfuerzo realizado para poder hacer uso de la nueva tecnología. Es decir, una se centra en determinar cuán útil es la implementación de nueva tecnología, y la otra se centra en determinar el nivel de esfuerzo necesario para hacer uso de dicha tecnología. Al combinar estos dos aspectos fundamentales, se tiene como resultado una medida bastante aproximada sobre la actitud hacia el uso e intención de uso [19]. A nivel de trasfondo, este modelo se fundamenta en algunas teorías psicológicas comprobadas, en donde se destaca la Teoría de la Acción Razonada (TRA). Este aspecto ha catapultado a TAM a ser uno de las principales metodologías para conseguir evaluar el nivel de aceptación tecnológica en muchas investigaciones en distintos campos [20].

Su versatilidad ha conseguido que TAM se utilice para evaluar el nivel de aceptación tecnológica en muchos campos, que van desde tiendas en línea, aplicaciones web, móviles, o sistemas inteligentes. Para potenciar aún más esta versatilidad, TAM ha poseído extensiones como TAM2 y UTAUT, que toma como base el modelo, pero añade variables extras, para aumentar aún más la precisión de la predicción, dichas variables se centran exclusivamente en factores contextuales y sociales [21]. Sin embargo, debido a tener que considerar los elementos culturales en contextos específicos, este modelo es altamente criticado por su complejidad de aplicación, ya que tomar en cuenta factores culturales para la adopción de una nueva tecnología, no es algo que se tome en cuenta en un inicio, por lo que, al hacerlo, su complejidad se vuelve exponencial, lo que ocurre del mismo modo con la extensión de 'pruebas mínimas para poder tener una predicción acertada [22]. Pero, a pesar de todas estas limitaciones, TAM sigue siendo una de las alternativas más viables y útiles para poder determinar el nivel de aceptación tecnológica, brindando la posibilidad de obtener datos cuantificados que dan una idea bastante clara y concisa sobre la adopción de nuevas tecnologías. Dichos datos facilitan notoriamente el análisis, lo que permite desarrollar e implementar estrategias específicas y útiles centradas en conseguir que los usuarios hagan uso (adopten) nuevas tecnologías [23].

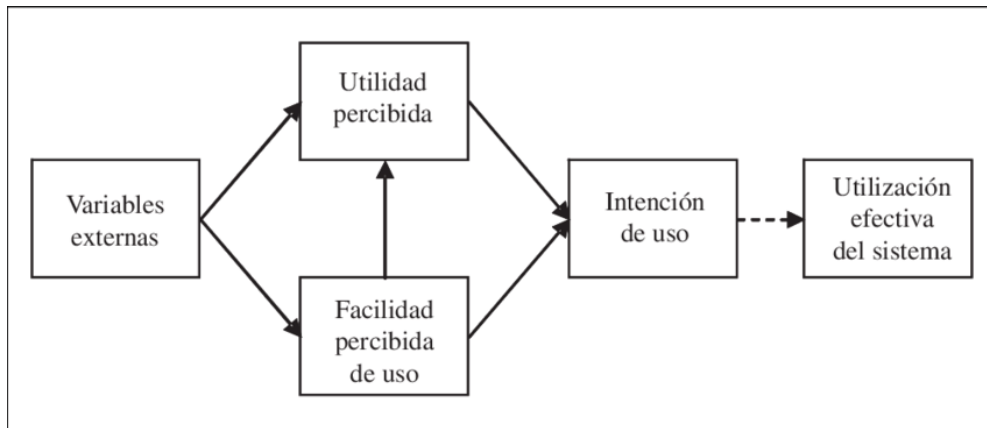


Figura 2: Modelo de Aceptación Tecnológica (TAM) [24]

Debido a todas las características mencionadas anteriormente, se considera que TAM es la metodología ideal para poder determinar el nivel de aceptación y adopción tecnológica del sistema propuesto. Esto, ya que TAM permite tener medidas cuantificadas desde varios aspectos y puntos de vista del software, brindando la posibilidad de determinar el nivel que facilidad de uso e intención de uso de los usuarios en relación al sistema. Poder tener una medida cuantificada de la adopción por parte de los usuarios de una nueva tecnología (software a desarrollar), es un indicador muy relevante para poder dictaminar si el desarrollo e implementación del sistema fue o no acertado.

4.4 Tecnologías

4.4.1 Python

Python es un lenguaje de programación de alto nivel, que pretende brindar a los programadores, una manera sencilla, simple y legible de desarrollar un sistema, mediante buenas prácticas de programación [25]. Además, este es un lenguaje que posibilita implementar gran parte de los paradigmas de programación existentes en la actualidad, tales como: programación orientada a objetos, estructurada o funcional. Este aspecto lo convierte en uno de los lenguajes más flexibles y versátiles para desarrollar cualquier tipo de sistemas en diferentes campos [26].

Python posee una de las mayores comunidades en el mundo Open Source, lo que ha conllevado que de forma nativa posea muchas bibliotecas de uso cotidiano, aunque siempre con la posibilidad de agregar nuevas librerías que permiten a los desarrolladores codificar funcionalidades considerablemente complejas, que abarcan desde una simple página web, hasta módulos complejos como análisis de datos, inteligencia artificial, entre otras [27]. Gracias a

dichas comunidades, Python se ha consolidado como uno de los lenguajes con mayor cantidad de bibliotecas, permitiendo construir aplicaciones web con Django o entrenar modelos con TensorFlow [28]. Así mismo, Python ha integrado herramientas como Cython o PyPy, que le permiten competir directamente con otros lenguajes de programación (como C++) en términos de velocidad de rendimiento, el cual es uno de sus aspectos a mejorar de forma nativa [29]. Todas estas características, han conllevado que, gracias a su simplicidad y efectividad, Python es uno, por no decir el lenguaje preferido tanto para programadores experimentados como para programadores novatos, ya que permite solucionar problemas sencillos o complejos, que van desde automatizaciones de procesos simples, hasta big data que requiere gran cantidad de recursos computacionales [30].

Debido a todas las características mencionadas anteriormente, se considera que Python es el lenguaje ideal para el desarrollo del presente Trabajo de Integración Curricular por su amplio y extenso uso en la Inteligencia Artificial. Esto, ya que se pretende integrar el reconocimiento facial como un método biométrico, para lo cual librerías como opencv, face-recognition, keras, tensorflow, etc., permiten agilizar en gran medida el desarrollo de dicha integración.

4.4.2 ODOO

Odoe es un ERP (Enterprise Resource Planning) de código abierto, que pretende satisfacer las necesidades de todo tipo de negocios, desde negocios pequeños, hasta grandes empresas corporativas, este sistema de gestión empresarial integrado desarrollado en 2005, es muy flexible, fácil de manejar, y brinda la posibilidad de integrar módulos tanto desarrollados por terceros como preinstalados de forma nativa [31].

De forma nativa, Odoe posee más de 30 módulos principales, los cuales abarcan desde contabilidad, inventario, ventas, entre otros. Dichos módulos, permiten la integración con otros módulos, extrapolando en gran medida lo que puede ofrecer a las empresas según sus necesidades. Además, Odoe permite desarrollar módulos personalizados, lo que lo convierte en una herramienta muy versátil y flexible, en donde no solamente se tiene una extensa gama de módulos de forma nativa, si no que se pueden desarrollar nuevos módulos para solventar problemas específicos del modelo de negocio [32].

Odoe integra tecnologías de desarrollo contemporáneas que, al trabajar de manera conjunta, dan como resultado un marco de desarrollo muy potente, para la lógica de negocios se maneja con Python; para el frontend con HTML 5 y JavaScript; y para la base de datos, se hace uso de PostgreSQL. Estas tecnologías garantizan un correcto desempeño, dando la posibilidad de escalar en gran medida. Además, gracias a su comunidad activa de código

abierto, cada vez se desarrollan módulos más completos en estas tecnologías que solventas problemáticas muy específicas, esa es la ventaja de los módulos personalizados de Odoo [33].

Actualmente, Odoo posee una versión de gratuita (pública) y otra de pago (empresarial), por obvias razones, la versión de pago posee muchas más características que favorecen a empresas grandes. Sin embargo, la versión gratuita no se queda atrás, cumple con lo prometido y es una opción muy viable para pequeños y medianos negocios. Por ello, Odoo es una de las alternativas más utilizadas en el mercado actual tanto por negocios pequeños, mediante y grandes empresas gracias a sus complementos y módulos personalizados [34].

Debido a todas las características mencionadas anteriormente, se considera que Odoo es la herramienta ideal para integrar el reconocimiento facial al sistema de control de acceso. Además, el servidor que aloja las aplicaciones de producción en la carrera de computación de la Universidad Nacional de Loja, posee un servidor activo de Odoo, lo que facilitará en gran medida el despliegue del sistema. Adicionalmente, ambos sistemas a integrar fueron desarrollados en Odoo, por lo que integrar ambos sistemas a una misma versión de Odoo sería el objetivo final.

4.4.3 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto. Fue desarrollado en 1986 en la Universidad de California, con la idea de presentar una alternativa fuerte en: consistencia, escalabilidad y conformidad con las especificaciones SQL. Debido a estos tres enfoques, PostgreSQL es en la actualidad una de las principales opciones para las aplicaciones empresariales que buscan ser altamente escalables y manejan datos críticos [35].

Una de las principales características que la diferencian del resto de gestores de bases de datos, es su facilidad para manejar datos complejos, tales como índices de datos o consultas almacenadas. Además, oferta la replicación en tiempo real, el control de concurrencia, y transacciones de base ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) que tienen como finalidad asegurar la integridad y, sobre todo, la precisión de los datos almacenados [36]. Del mismo modo, otra de las principales características, es su gran desempeño cuando se trata de altas cargas de trabajo, es decir, procesar gran volumen de datos, esto, con la facilidad de ser utilizado tanto en ambientes locales como basados en la nube, ha conllevado a que destaque en el mercado actual. Por si fuera poco, PostgreSQL también puede ser programado en varios lenguajes de programación populares, como lo son R, Java y el mismo Python. Esta característica le brinda la posibilidad de una alta estabilidad en varias áreas de conocimiento,

que van desde análisis avanzado de datos, aplicaciones basadas en inteligencia artificial o simplemente servicios web [37].

PostgreSQL representa una competencia directa a soluciones mucho más comerciales y establecidas como lo son Oracle y SQL Server. Esto, gracias a su comunidad muy activa que mejora constantemente esta tecnología, proporcionando versiones, extensiones, y documentación de alto nivel, lo que se traduce en una actividad comunitaria de código abierto considerable, que le permite competir con empresas asentadas en el mercado actual [38].

Debido a todas las características mencionadas anteriormente, se considera que PostgreSQL es el gestor de bases de datos ideal para integrar el reconocimiento facial al sistema de control de acceso. Esto, ya que al tratarse de un sistema que emplea altos volúmenes de datos e inteligencia artificial, esta tecnología cumple de gran manera su trabajo. Además, el servidor de Odoos que se encuentra activo en la carrera de computación de la Universidad Nacional de Loja, hace uso de esta base de datos, al igual que los otros dos sistemas a integrar. Por tanto, se pretende centralizar en una sola base de datos en PostgreSQL, todo el tráfico de datos que exista en los laboratorios de software de la carrera de computación de la Universidad Nacional de Loja posee un servidor activo de Odoos, lo que facilitará en gran medida el despliegue del sistema. Adicionalmente, ambos sistemas a integrar fueron desarrollados en Odoos, por lo que integrar ambos sistemas a una misma versión de Odoos sería el objetivo final.

4.5 Trabajos relacionados

Tabla 1. Trabajos relacionados

Trabajos relacionados	Resumen
Sistema de reconocimiento facial automatizado para el control de asistencia de recursos humanos [39].	Este trabajo desarrolla un sistema que permite registrar asistencia mediante reconocimiento facial a recursos humanos dentro de la Asociación de Desarrollo Comunitario y Medio Ambiente Namanda [39]. Por tanto, este es un trabajo que se relaciona en gran medida al presente TIC, ya que, mediante este, se puede visualizar el uso de la librería OpenCV y el lenguaje JAVA.
Registro de asistencia de alumnos por medio de reconocimiento facial utilizando visión artificial [40].	Este trabajo desarrolla un sistema que permite registrar asistencia mediante reconocimiento facial a los estudiantes de la UTA Ambato [40]. Por tanto, este es un trabajo que se relaciona en gran medida al presente TIC, ya que, mediante este, se puede visualizar el uso y comportamiento de las redes neuronales para poder detectar rostros faciales y, posteriormente, procesar los datos para encontrar similitudes y poder marcar asistencia.
Sistema biométrico de reconocimiento facial para el control de asistencia del personal docente y administrativo de la Uniandes Tulcán [41].	Este trabajo desarrolla un sistema que permite registrar asistencia mediante reconocimiento facial del personal docente y administrativo de la Uniandes Tulcán [41]. Por tanto, este es un trabajo que se relaciona en gran medida al presente TIC, ya que, mediante este, se puede visualizar una aplicación de escritorio completamente funcional que

	tiene la misma finalidad del registro de asistencia mediante reconocimiento facial.
Prototipo de reconocimiento facial para mejorar el control de asistencia de estudiantes en el laboratorio de cómputo, UNIANDES-QUEVEDO [42].	Este trabajo desarrolla un sistema que permite registrar asistencia mediante reconocimiento facial a los estudiantes que hagan uso del laboratorio de cómputo de la Uniandes Quevedo [42]. Por tanto, este es un trabajo que se relaciona en gran medida al presente TIC, ya que, mediante este, se puede visualizar un prototipo desarrollado en una raspberry Pi que permite el registro de asistencias con reconocimiento facial, funcionando correctamente y haciendo uso de la IA.

5 Metodología

Esta sección detalla la metodología empleada para llevar a cabo el desarrollo del trabajo de titulación. Para ello, en la sección **5.1** se indica el área de estudio que sitúa el lugar en donde se implementó el TIC. La sección **5.2** aborda todo el procedimiento que se llevó a cabo para cumplir los dos objetivos específicos planteados. La sección **5.3** presenta los recursos necesarios para el desarrollo del TIC. Finalmente, la sección **5.4** menciona los participantes involucrados en el TIC.

5.1 Área de Estudio

El presente Trabajo de Integración Curricular (TIC) se llevó a cabo durante el período académico: octubre 2023 a marzo 2024 en los laboratorios de Software que posee la carrera de Computación ubicada dentro de la Facultad de Energía, las industrias y los Recursos Naturales no Renovables (FEIRNNR) de la Universidad Nacional de Loja (UNL). Cabe recalcar que, estas instalaciones son utilizadas por diversos tipos de personal universitario, como: estudiantes, docentes, administrativos, personal de limpieza, encargados, entre otras. Sin embargo, el alcance del presente TIC se enfoca al registro de asistencia, por lo que abarca únicamente tres tipos de los mencionados, los cuales son: **estudiantes**, los cuales hacen uso de las instalaciones para recibir clases curriculares; **docentes**, los cuales hacen uso de las instalaciones para impartir clases curriculares; y, **administrador**, el cual es el encargado de brindar el respectivo mantenimiento a los laboratorios, así como controlar la asignación de horarios y personal que ingresa a los mismos. Dichas instalaciones se encuentran actualmente en las coordenadas: -4.029896540034394, -79.19941943678909, tal y como se puede observar en la **Figura 3**.



Figura 3. Ubicación de Laboratorios de FEIRNNR de la UNL

5.2 Procedimiento

La siguiente sección detalla todo el procedimiento realizado para cumplir con los objetivos planteados en un inicio.

5.2.1 Objetivo Específico 1. Desarrollar un software que permita integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso para el registro de asistencias en los laboratorios de software en Python, mediante la metodología XP

Para cumplir con el primer objetivo planteado, se lo realizó en base a la metodología XP, abarcando sus cinco fases establecidas, las cuales son: planificación, diseño, codificación, pruebas y lanzamiento. Cabe recalcar que, los resultados generales obtenidos en cada fase de la metodología se encuentran en el desarrollo del primer objetivo (véase sección de **Resultados, apartado 6.1**). Por otro lado, a esta metodología se le dio un enfoque iterativo e incremental, por lo que, se dividió el desarrollo del proyecto en iteraciones, en donde cada iteración abarca las fases de la metodología XP, dichas iteraciones pueden observarse en el plan de iteración (véase sección de **Resultados, apartado 6.1.1.7**). A continuación, se detalla el procedimiento acorde a cada una de las fases.

5.2.1.1 Fase de planificación

Se realizó el análisis, tanto de la observación participante (véase sección de **Metodología, apartado 5.3.2.4**), como de la entrevista (véase sección de **Metodología, apartado 5.3.2.3**) efectuadas a los clientes directos del sistema. Luego, se revisó las memorias de los Trabajos de Integración Curricular a integrar con su respectivo código fuente, con la

finalidad de conocer la situación actual de ambos sistemas (véase sección de **Metodología, apartado 5.3.2.8**). Estos tres métodos de recopilación de información, permitió identificar los actores involucrados y las funcionalidades que pueden realizar cada uno de ellos (véase sección de **Resultados, apartado 6.1.1.4**). Con las funcionalidades de cada actor y, con el análisis de toda la información recaba hasta el momento, se logró abstraer los requisitos del sistema (véase sección de **Resultados, apartado 6.1.1.5**). Con los requisitos del sistema, se pudo elaborar las historias de usuario que abarcan el alcance del sistema en su totalidad, donde cada requisito posee su respectiva historia de usuario con muchos más detalles y especificaciones del mismo (véase sección de **Resultados, apartado 6.1.1.6**). Con las historias de usuario definidas, se elaboró el plan de iteración, el cual permite agrupar las historias de usuario según se considere más conveniente para el desarrollo del software siguiendo la implementación de la metodología XP (véase sección de **Resultados, apartado 6.1.1.7**).

5.2.1.2 Fase de diseño

Se analizó las historias de usuario del sistema definidas en la **Fase de planificación**. Esto, permitió elaborar un conjunto de diagramas que pretenden graficar las funcionalidades a codificar del sistema, haciendo más fácil de entender el alcance del mismo. En primer lugar, se elaboró la arquitectura de software, la cual pretende demostrar cómo se desplegará el sistema mediante el diagrama de despliegue, y cómo están relacionados sus componentes mediante el diagrama de componentes (véase sección de **Resultados, apartado 6.1.2.1**). Luego, se elaboró los diagramas UML (diagrama de clases, modelo relacional, casos de uso y de actividades), los cuales buscan graficar las funcionalidades que tendrá el sistema a desarrollar (véase sección de **Resultados, apartado 6.1.2.2**). Finalmente, se diseñó los respectivos prototipos en base a cada iteración (véase sección de **Resultados, apartado 6.1.2.3**), esto, ya que, los prototipos pueden representar gráficamente los criterios de aceptación (requisitos a codificar) de más de una historia de usuario, por tanto, se agrupó directamente por iteración y no por historia de usuario.

5.2.1.3 Fase de codificación

Se desarrolló el código fuente utilizando los diseños creados durante la **Fase de diseño** y las historias de usuario elaboradas durante la **Fase de planificación** (véase sección de **Resultados, apartado 6.1.3**). Esto permitió verificar los criterios de aceptación que validan el correcto funcionamiento de cada requisito, así como todas sus observaciones y necesidades detalladas. Es importante mencionar que el desarrollo del sistema se realizó de acuerdo con el plan de iteración (véase sección de **Resultados, apartado 6.1.1.7**), lo que implicó segmentar los entregables en función del número de iteraciones.

5.2.1.4 Fase de pruebas

Se llevó a cabo una prueba de validación operativa del sistema en un ambiente controlado, para simular el funcionamiento del software en un escenario real (véase sección de **Resultados, apartado 6.1.4.1**). Además, se ejecutó pruebas unitarias para validar el correcto funcionamiento del código desarrollado (véase sección de **Resultados, apartado 6.1.4.2**). Así mismo, se efectuaron pruebas de aceptación en las que participaron los clientes directos del sistema, esto, mediante una reunión (véase sección de **Metodología, apartado 5.3.2.6**), donde se presentó el sistema funcional y, una vez validado su correcto funcionamiento, se procedió a la firma de las pruebas de aceptación del software, donde se detalla y valida que el sistema se encuentra finalizado de acuerdo con el alcance definido (véase sección de **Resultados, apartado 6.1.4.3**).

5.2.1.5 Fase de lanzamiento

Se desplegó el sistema completamente funcional y validado por los clientes directos, a los servidores de la carrera de Computación, en el entorno de producción. Durante esta fase, se elaboraron los respectivos manuales de usuario para todos los roles del sistema, que incluyen: manual de usuario para administrador (véase sección de **Resultados, apartado 6.1.5.1**), docente (véase sección de **Resultados, apartado 6.1.5.2**) y estudiante (véase sección de **Resultados, apartado 6.1.5.3**). Además, se detalló el proceso seguido para la instalación y despliegue del sistema mediante el manual de instalación (véase sección de **Resultados, apartado 6.1.5.4**).

5.2.2 Objetivo Específico 2. Determinar el nivel de aceptación tecnológica del software para el registro de asistencia en los laboratorios de software mediante el modelo TAM

Para cumplir con el segundo objetivo planteado, se lo realizó en base al Modelo de Aceptación Tecnológica (TAM), el cual, se dividió en tres fases, las cuales son: elaboración, medición y análisis.

5.2.2.1 Fase de elaboración

Se elaboró un cuestionario que permita determinar la percepción del usuario en relación a cada una de las variables de TAM a evaluar, las cuales son: Utilidad Percibida (UP), Facilidad de Uso Percibida (FUP), Actitud hacia el Uso (AU) y percepción de Intención de Uso (IU), además, es necesario hacer hincapié en que, para cada pregunta, se definió un identificador que asocia a dicha pregunta con la variable (constructo) a evaluar y, para obtener valores cuantificados para cada pregunta, se utilizó la escala de Likert, la cual permitió ponderar las respuestas en una escala de 0 a 4 (véase sección de **Resultados, apartado 6.2.1**).

5.2.2.2 Fase de medición

Se llevó a cabo una encuesta utilizando Google Forms, la cual se basó en el cuestionario desarrollado durante la **Fase de elaboración**. Esta encuesta se dirigió a 42 estudiantes, los cuales representan el 100% de la población a evaluar, la cual corresponde a tres ciclos de la carrera de Computación: Segundo “A”, Séptimo “A” y Octavo “A” (véase sección de **Metodología, apartado 5.3.2.5**). Como resultado, se generó una matriz de resultados que recopila las respuestas de manera clara y organizada (véase sección de **Resultados, apartado 6.2.2**).

5.2.2.3 Fase de análisis

Se analizó la matriz de resultados obtenida en la **Fase de medición**. Este análisis se realizó para cada pregunta, lo que permitió identificar el nivel de aceptación tecnológica para cada constructo (mediante la escala de Likert). Además, es importante destacar que, mediante el análisis individual de cada constructo, se logró determinar el nivel de aceptación tecnológica general del sistema (véase sección de **Resultados, apartado 6.2.3**).

5.3 Recursos

Se hizo uso de recursos tanto científicos como técnicos, los cuales permitieron cumplir con los objetivos planteados, facilitando el desarrollo del proyecto, y sustentándolo con recursos pertinentes.

5.3.1 Recursos científicos

5.3.1.1 Investigación bibliográfica

Esta técnica permitió sustentar el fundamento teórico del TIC, desde: antecedentes, fundamentación teórica, tecnologías, metodologías y trabajos relacionados. Esto, mediante consultas realizadas en fuentes confiables, tales como: libros, revistas científicas, artículos científicos, trabajos de titulación (véase sección **Marco Teórico**).

5.3.1.2 Experimento

Se llevó a cabo un proceso de experimentación, el cual detalla las actividades a ejecutarse para poder evaluar el software desarrollado, dicho proceso, se empleó específicamente en el segundo objetivo planteado, el cual abarca la evaluación de la aceptación tecnológica del software mediante TAM, esto, por parte de todas las partes involucradas, los cuales fueron: estudiantes, docentes y encargado del laboratorio (véase sección de **Metodología, apartado 5.2.2**).

5.3.1.3 Método Científico

Este método fue la base para el desarrollo del TIC. Esto, desde el momento en que se planteó la respectiva propuesta del Proyecto de Integración Curricular (PIC) hasta la culminación del presente Trabajo de Integración Curricular (TIC). Esto, se evidencia mediante el desarrollo de las diferentes actividades planteadas para cada objetivo planteado (véase sección de **Metodología**), los resultados obtenidos en cada fase (véase sección de **Resultados**), su respectiva discusión (véase sección de **Discusión**), conclusiones (véase sección de **Conclusiones**) y recomendaciones (véase sección de **Recomendaciones**).

5.3.2 Recursos técnicos

5.3.2.1 Metodología XP

Esta metodología se utilizó para poder cumplir con el primer objetivo, esto, de acuerdo al desarrollo de cada una de sus fases (véase sección de **Metodología, apartado 5.2.1**).

5.3.2.2 Metodología TAM

Esta metodología se utilizó para poder cumplir con el segundo objetivo, esto, de acuerdo al desarrollo de cada una de sus fases (véase sección de **Metodología, apartado 5.2.2**).

5.3.2.3 Entrevista

Esta técnica se utilizó para poder abstraer los requisitos del sistema (véase sección de **Resultados, apartado 6.1.1.5**) y las historias de usuario (véase sección de **Resultados, apartado 6.1.1.6**). Esto, mediante la entrevista realizada al Ing. Pablo Ordoñez, director de la carrera de computación (véase sección de **Resultados, Análisis de entrevista, Tabla 3**).

5.3.2.4 Observación Participante

Esta técnica se utilizó para poder abstraer los requisitos del sistema (véase sección de **Resultados, apartado 6.1.1.5**) y las historias de usuario (véase sección de **Resultados, apartado 6.1.1.6**). Esto, mediante la observación participante realizada al Ing. Luis Sinche, encargado de los laboratorios de software de la carrera de computación (véase sección de **Resultados, Análisis de observación participante, Tabla 2**).

5.3.2.5 Muestreo por conveniencia

Esta técnica se utilizó para poder determinar una población accesible a evaluar de toda la carrera de Computación. Esto, ya que la carrera cuenta con más de 200 estudiantes actualmente, aparte de personal administrativo, personal docente, entre otras. Por tanto, evaluar a toda la población es totalmente inviable por la disponibilidad de tiempo y de recursos que se posee. Ante dicha problemática, se seleccionó por conveniencia una población de tres cursos (ciclos) que evaluarán el sistema, los cuales fueron: Segundo “A”, Séptimo “A” y Octavo “A”,

dando un total de 42 estudiantes. Cabe recalcar que, se seleccionó dichos cursos debido a: la viabilidad práctica, que, al ser una cantidad considerable, pero manejable de estudiantes, se pudo manejar de manera efectiva los resultados obtenidos; y a la facilidad de acceso a los estudiantes y docentes, pues, para poder evaluar el sistema, se tuvo que hacer uso de horas curriculares y de las instalaciones de los laboratorios, para realizar tanto la demostración del sistema como la evaluación del mismo, lo que conllevó una gran limitante de tiempo y espacio pertinente para dichos fines. No obstante, se consiguió demostrar, y, sobre todo, determinar el nivel de aceptación tecnológica del sistema al momento de efectuar la encuesta basada en el modelo de Aceptación Tecnológica (véase sección de **Metodología, apartado 5.3.2.7**).

5.3.2.6 Reunión

Esta técnica se utilizó, para que los clientes directos del sistema (director de la carrera de Computación y encargado de los laboratorios de Computación) puedan hacer uso del sistema y, una vez satisfechos con el resultado obtenido, puedan aprobar dicho sistema, mediante la firma de las pruebas de aceptación del software, donde se detalla y valida que el sistema se encuentra finalizado de acuerdo con el alcance definido (véase sección de **Resultados, apartado 6.1.4.3**).

5.3.2.7 Encuesta

Esta técnica se utilizó para poder obtener respuestas cuantificadas por parte de los usuarios que hicieron uso del sistema, esto mediante la encuesta realizada y aplicada en Google Forms a una población específica de usuarios. Dicha encuesta fue elaborada en base al modelo de aceptación Tecnológica (TAM) y, para las posibles respuestas, en base a la escala de Likert (véase sección de **Resultados, apartado 6.2.1**).

5.3.2.8 Revisión de documentos

Esta técnica se utilizó de manera extensiva, es decir, no solo se revisó la memoria (documentación que detalla todo el proceso realizado en la elaboración del sistema), sino también, los artefactos de software de los Trabajos de Integración Curricular (TIC) a integrar (véase sección de **Resultados, apartado 6.1.1.3**), específicamente: el código fuente y la observación del sistema en ejecución. Esto, en base al enfoque propuesto por [43], donde se argumenta que el código fuente de un sistema representa una forma primaria de documentación.

En primera instancia, se realizó un análisis al TIC denominado “Desarrollo de un prototipo de control de acceso a los laboratorios de la carrera de Computación de la UNL: SmartLab”, el cual está orientado al control de acceso a los laboratorios de software de la carrera de Computación de la UNL (véase sección de **Resultados, Revisión de documentos, Tabla 4**).

Posteriormente, al TIC denominado “Prototipo de Reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja”, donde dicho trabajo, está orientado al registro local de asistencias mediante reconocimiento facial a los laboratorios de software (véase sección de **Resultados, Revisión de documentos, Tabla 5**).

5.4 Participantes

Los participantes involucrados en el desarrollo del presente TIC son los siguientes:

- Luis Alfredo Jumbo Herrera, estudiante autor del presente TIC, el cual intervino desde el planteamiento del Proyecto de Integración Curricular (PIC), hasta el cumplimiento de los objetivos planteados en el TIC, los cuales abarcaron el desarrollo del sistema, y, posteriormente, la evaluación de la aceptación tecnológica del software siguiendo la metodología planteada en el proyecto (véase sección de **Metodología**).
- Ing. Edison Leonardo Coronel Romero, Mg. Sc., como director del TIC, el cual guió y supervisó el desarrollo del proyecto en su totalidad.
- Ing. Pablo Fernando Ordoñez Ordóñez, Mg. Sc. como director de la carrera de Computación en la UNL, el cual intervino: en la abstracción de los requisitos del sistema mediante una entrevista (véase sección de **Resultados, apartado 6.1.1.2**; Error! No se encuentra el origen de la referencia.); y en la realización de las pruebas de aceptación del sistema (véase sección de **Resultados, apartado 6.1.4.3**).
- Ing. Luis Sinche, Mg. Sc. como encargado de los laboratorios de software de la carrera de Computación en la UNL, el cual intervino: en la abstracción de los requisitos del sistema mediante una observación participante (véase sección de **Resultados, apartado 6.1.1.1**), y en la realización de las pruebas de aceptación del sistema (véase sección de **Resultados, apartado 6.1.4.3**).
- Estudiantes pertenecientes a los cursos: Segundo “A”, Séptimo “A” y Octavo “A”, los cuales participaron en la evaluación del sistema, para poder determinar el nivel de aceptación tecnológica (TAM) del mismo (véase sección de **Metodología, apartado 5.3.2.5**).

6 Resultados

En la presente sección se detalla los resultados obtenidos por cada uno de los dos objetivos específicos planteados en el trabajo de titulación. Para ello, en la sección **6.1** se indica el procedimiento que se llevó a cabo para cumplir con el primer objetivo específico planteado siguiendo la metodología XP. Del mismo modo, la sección **6.2** se indica el procedimiento que se llevó a cabo para cumplir con el segundo objetivo específico planteado siguiendo la metodología TAM.

6.1 Objetivo Específico 1. Desarrollar un software que permita integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso para el registro de asistencias en los laboratorios de software en Python, mediante la metodología XP

Para cumplir con el primer objetivo específico, se aplicó la metodología XP, la misma que posee cinco fases, las cuales son: planeación, diseño, codificación, pruebas y lanzamiento [17]. A continuación, se presentan los resultados obtenidos para la metodología, los cuales van desde la sección **Fase de planificación** hasta la sección **Fase de lanzamiento**.

6.1.1 Fase de planificación

En esta fase, se presenta el análisis, tanto de la observación participante (véase sección de **Resultados, apartado 6.1.1.1**), como de la entrevista (véase sección de **Resultados, apartado 6.1.1.2**) efectuadas a los clientes directos del sistema (Encargado de los laboratorios de software y Director de la carrera de Computación); la revisión de documentos de los dos Trabajos de Integración Curricular a integrar con su respectivo código fuente y observación del funcionamiento actual de ambos sistemas (véase sección de **Resultados, apartado 6.1.1.3**); los actores que intervienen en el sistema, así como sus respectivas funciones (véase sección de **Resultados, apartado 6.1.1.4**); los requisitos (véase sección de **Resultados, apartado 6.1.1.5**) e historias de usuario del sistema (véase sección de **Resultados, apartado 6.1.1.6**); además, se presenta el plan de iteración (véase sección de **Resultados, apartado 6.1.1.7**). Además, es necesario enfatizar en que esta sección presenta sólo de manera general la fase de planificación del desarrollo bajo la metodología XP (para ver el desarrollo completo de la fase de planificación bajo la metodología XP, véase sección de **Anexos, apartado ANEXO 4: Implementación de metodología XP, fase de planificación de cada iteración**).

6.1.1.1 Análisis de observación participante

En esta sección, se presenta el análisis realizado a la observación participante efectuada al cliente directo del sistema que cumple con el cargo de “encargado de los laboratorios de software”. La **Tabla 2** muestra la información más relevante obtenida de la observación

participante dirigida al Ing. Luis Sinche, encargado de los laboratorios de software de la carrera de computación de la UNL (para observar la observación participante completa, véase la sección de **Anexos, apartado ANEXO 1: Entrevista realizada al encargado de los laboratorios de software de la carrera de computación, Ing. Luis Sinche, para la elicitación de requisitos**). Cabe mencionar que, de toda la observación participante realizada, se consideró como “información más relevante”, a las observaciones encontradas, y a partir de ellas, las conclusiones resultantes. De esta manera, se pudo comprender de manera más práctica y detallada el funcionamiento actual del sistema de Control de Acceso, así como los nuevos requisitos a integrar al software.

Tabla 2. Informe de análisis a la observación participante efectuada al encargado de los laboratorios de software de la carrera de Computación

Análisis de observación participante	
Identificador: OP01	Fecha de la entrevista: 27 de octubre del 2023
Entrevistado: Ing. Luis Sinche, encargado de los laboratorios de software de la carrera de Computación de la Universidad Nacional de Loja	
Entrevistador: Sr. Luis Jumbo, autor del Trabajo de Integración Curricular (tesista)	
Introducción:	
<p>Para poder comprender el contexto actual de los laboratorios de software de la Universidad Nacional de Loja, se llevó a cabo una observación participante realizada al encargado de dichas instalaciones, el Ing. Luis Sinche. Esto, con la finalidad de poder comprender la mejor manera de integrar la nueva funcionalidad de “Reconocimiento Facial” al sistema actual de “Control de Acceso”. Ya que se debe integrar el reconocimiento facial como método biométrico para el registro de asistencias dentro de los laboratorios, el objetivo de esta observación participante es el de poder tener una visión clara de dicha integración, desde los requisitos a desarrollar hasta la forma en el que se va a desplegar.</p>	
Observaciones:	
<ul style="list-style-type: none"> • El sistema actual se centra en el control de acceso mediante tarjetas RFID, registra los ingresos a los laboratorios y verifica la temperatura corporal. Sin embargo, no está diseñado para gestionar las asistencias de los usuarios que hagan uso de estas instalaciones en horario curricular. • El módulo de Asistencias genera reportes, muestra intentos fallidos, y permite filtrar los registros de manera muy limitada, sin poder filtrar por rango de fechas, por meses, usuarios y cosas por el estilo. • Cuando se actualiza y se crea un nuevo periodo académico, se realiza toda la asignación del horario con sus respectivos cursos y estudiantes de manera manual. • Se necesita implementar una interfaz gráfica simple y sencilla de utilizar, con filtros de fechas, que permitan segmentar por meses o semanas según un usuario en concreto. • Se necesita restringir el registro de asistencia por un rango de IP específico que solo se encuentre dentro de los laboratorios de software de la UNL. 	
Conclusiones:	
<ul style="list-style-type: none"> • El sistema actual se centra en el control de acceso a los laboratorios de software, y no se encuentra diseño para soportar el registro y gestión de asistencias dentro de dichas instalaciones en horario curricular. • El sistema actual posee muchas deficiencias en torno a la interfaz gráfica, ya que no es muy intuitiva y los filtros que se posee no son de fácil acceso y de uso sencillo. Es un aspecto a considerar en las nuevas vistas para la gestión de asistencias mediante reconocimiento facial. 	

- Se debe verificar la dirección IP y el UUID de los dispositivos que intenten realizar un registro de asistencia antes que empiece el proceso de reconocimiento facial. De esta manera se asegura que el usuario se encuentra dentro de los laboratorios haciendo uso de una de las máquinas previamente registradas en el sistema.
- Se debe agregar la nueva funcionalidad del registro de asistencia mediante reconocimiento facial al sistema actual, sin modificar en gran medida las secciones que ya funcionan correctamente. Se puede considerar como una actualización de funcionalidades del sistema actual, por lo que es el sistema actual es la base en la que se debe construir el nuevo sistema propuesto.

6.1.1.2 Análisis de entrevista

En esta sección, se presenta el análisis realizado a la entrevista efectuada al cliente directo del sistema que cumple con el cargo de “director de la carrera de Computación”. La **Tabla 3** muestra la información más relevante obtenida de la entrevista dirigida al Ing. Pablo Ordoñez, director de la carrera de Computación de la UNL (para observar la entrevista completa, véase la sección de **Anexos, apartado ANEXO 2: Entrevista realizada al director de la carrera de computación, Ing. Pablo Ordoñez, para la elicitación de requisitos**). Cabe mencionar que, de toda la observación participante realizada, se consideró como “información más relevante”, a las observaciones encontradas, y a partir de ellas, las conclusiones resultantes. De esta manera, se pudo comprender de manera más precisa y detallada el nuevo funcionamiento deseado, tanto a nivel técnico (como reutilización del código fuente actual, reutilización de la base de datos existentes, refactorización del código fuente, etc.), como a nivel general (que se registre la asistencia teniendo en cuenta los horarios respectivos; que sólo el Administrador no posee horarios, el resto sí; etc.).

Tabla 3. Informe de análisis a la entrevista efectuada al director de la carrera de Computación

Análisis de entrevista	
Identificador: E01	Fecha de la entrevista: 01 de noviembre del 2023
Nombre del documento revisado: Ing. Pablo Ordoñez, director de la carrera de Computación de la Universidad Nacional de Loja.	
Entrevistado: Ing. Pablo Ordoñez, director de la carrera de Computación de la Universidad Nacional de Loja.	
Entrevistador: Sr. Luis Jumbo, autor del Trabajo de Integración Curricular (tesista)	
Introducción: Para poder comprender el contexto actual de los laboratorios de software de la Universidad Nacional de Loja (UNL), y de los sistemas a integrar, se llevó a cabo una entrevista realizada director de la carrera de Computación de la UNL, el Ing. Pablo Ordoñez. Esto, con la finalidad de poder comprender la mejor manera de integrar la nueva funcionalidad de “Reconocimiento Facial” al sistema actual de “Control de Acceso”. Ya que se debe integrar el reconocimiento facial como método biométrico para el registro de asistencias dentro de los laboratorios, el objetivo de esta entrevista es participante es el de poder tener una visión clara de dicha integración, desde los requisitos a desarrollar hasta la forma en el que se va a desplegar.	
Observaciones:	
<ul style="list-style-type: none"> • Se necesita que el reconocimiento facial se lo realice mediante la cámara web y no por la cámara local del servidor. 	

-
- Se necesita integrar el reconocimiento facial con el sistema existente que usa tarjetas RFID y mide la temperatura denominada [Control de Acceso].
 - Se necesita que la interfaz sea amigable para el usuario, que no represente un proceso complicado para marcar asistencia.
 - Se necesita que el administrador pueda ver reportes de asistencia, verificar horarios, y delimitar el desarrollo de cada práctica. Y que los estudiantes o docentes, solamente puedan desplegar la cámara web, realicen el reconocimiento facial y registren asistencia.
 - Se considera como un nuevo sistema, por lo que no se debe hacer uso de los datos existentes en el sistema actual.
 - El administrador es el rol que gestiona: periodos académicos, ciclos, asignaturas, carrera, estudiantes y docentes.
 - Se necesita mejorar el código existente, aplicando correctamente el patrón MVC.
 - Todos deben registrar asistencia, pero solamente el administrador no depende de un horario, el estudiante y docente sí.
-

Conclusiones:

- Debido a que existe un sistema actual que se encuentra en correcto funcionamiento, este se deberá tomar como base para integrar la nueva funcionalidad del reconocimiento facial.
 - Se debe implementar una solución basada en la cámara web y no en la cámara local del servidor. Esto implica combinar el desarrollo de Odoos con tecnologías web, como lo puede ser JavaScript.
 - La interfaz gráfica deberá ser muy amigable para el usuario, donde el proceso de registro de asistencia sea muy sencillo y eficaz.
 - Se deberá utilizar la misma segregación de roles que existe actualmente: administrador, estudiante y docente, agregando la dependencia del horario a dichos roles menos al administrador.
 - Se considera un sistema nuevo, por lo que se debe hacer uso del código fuente del sistema actual y manipularlo como se considere necesario, así mismo con la base de datos y el software desplegado en el ambiente de producción.
-

6.1.1.3 Revisión de documentos

En esta sección, se presenta la revisión extensiva realizada, tanto a la memoria (documentación que detalla todo el proceso realizado en la elaboración del sistema), como a los artefactos de software de los Trabajos de Integración Curricular (TIC) a integrar, específicamente: el código fuente y la observación del sistema en ejecución. Cabe mencionar que, esta revisión extensiva, se la realizó con la intención de comprender de mejor manera las necesidades de los clientes directos obtenidas en las secciones de **Análisis de observación participante** y **Análisis de entrevista**. Esto, ya que al analizar el funcionamiento actual de los sistemas y su respectivo código fuente, se puede determinar si las necesidades identificadas de los clientes son o no viables, entorno al tiempo estimado y los recursos disponibles. Es necesario hacer hincapié en que, las observaciones y conclusiones, abordan: el análisis de la memoria, código fuente y funcionamiento actual del sistema. Es decir, que no se encuentran separados por cada artefacto analizado, sino, los tres conjuntamente. Esto, debido a que los hallazgos encontrados se asocian unos con otros, y, para evitar redundancias, se colocó los hallazgos únicos encontrados tras el análisis a estos tres artefactos mencionados. Se considera necesario

explicar que, de toda la observación participante realizada, se consideró como “información más relevante”, a las observaciones encontradas, y a partir de ellas, las conclusiones resultantes. De esta manera, se pudo comprender de manera mucho más precisa y detallada el funcionamiento actual de ambos sistemas en un ambiente local (ambiente controlado), lo que permitió evaluar la memoria y el código fuente con dicho funcionamiento, dando lugar a hallazgos importantes, tanto a nivel técnico (como versiones específicas de librerías, tecnologías antiguas, desorganización del código fuente, falta de documentación en el código fuente, etc.), como a nivel general (carencia de documentación en la memoria de algunas partes que se encuentra en el código fuente pero no en la memoria, en la memoria se especifica ciertos experimentos o pruebas que avalan el uso de las tecnologías utilizadas, por lo que no se debe de volver a hacer; información relevante para sustentar la integración de ambos sistemas, dificultades que se presentaron y se deben de tomar en consideración para la integración; etc.).

La **Tabla 4** presenta la información más relevante obtenida del análisis del TIC que está orientado al control de acceso a los laboratorios de software de la carrera de Computación de la UNL.

Tabla 4. Informe de revisión de documentos, para el TIC, orientado al control de acceso a los laboratorios de software de la carrera de Computación de la UNL

Informe de revisión de documentos	
Identificador: RD01	Fecha de la revisión: 05 de noviembre del 2023
Nombre del documento revisado: Desarrollo de un prototipo de control de acceso a los laboratorios de la carrera de Computación de la UNL: SmartLab [1].	
Introducción: Para poder comprender el contexto actual del sistema, se llevó a cabo una revisión del Trabajo de Integración curricular, así como de su código fuente. Esto permitió obtener una visión general de la situación actual del sistema, tanto a nivel de código fuente como de documentación.	
Observaciones:	
<ul style="list-style-type: none"> • El sistema posee tres roles de usuario actuales: administrador, docente y estudiante. • El sistema no hace uso del patrón Modelo-Vista-Controlador (MVC) que recomienda Odoos para el desarrollo de nuevos módulos personalizados. • El código fuente no posee ninguna documentación, además, el nombre de las variables no es para nada intuitivas ni claras, por lo que no aportan un valor descriptivo. • Actualmente, Odoos se encuentra corriendo como un servicio en el servidor CentOS 7 de la carrera de Computación. Por tanto, implica que cualquier actualización o mantenimiento, conllevará a la interrupción momentánea pero completa de dicho servidor. • El sistema de [Control de Acceso] se encuentra ejecutando en un entorno virtual en el servidor, en donde sus librerías son relativamente antiguas. Se debe tomar en cuenta al momento de desarrollar el sistema propuesto. • Para actualizar el código fuente en producción, se debe tener muy en cuenta las versiones y librerías utilizadas antes de su despliegue. • Se debe agregar los campos de [UUID] y [rango de dirección IP] al momento de gestionar un [Laboratorios], con sus respectivas validaciones. 	

- Se debe agregar la gestión de [Equipos informáticos], la cual permita administrar los datos de los equipos que se van a hacer uso para el registro de asistencia. Esto lo realiza el administrador.
- Se debe agregar la gestión de [Asistencias Face ID], la cual permita administrar los registros de asistencia mediante reconocimiento facial. Esto lo pueden realizar todos los usuarios, pero no se puede eliminar ningún registro.

Conclusiones:

- Debido a la mala implementación del patrón MVC, el sistema actual no es para nada mantenible ni sostenible en el tiempo. Por tanto, se debe considerar primero implementar el patrón MVC para que resulte mucho más sencillo la integración del reconocimiento facial al software actual.
 - Existe una gran limitación al momento de comprender el código fuente actual, principalmente ya que no se documentó el código fuente, ni las variables son muy descriptivas. Esto conllevará a una demora considerable para comprender el código y posteriormente actualizarlo.
 - Debido a que Odoos es un servicio activo en el servidor de la carrera de Computación, se debe tener mucho cuidado al momento del despliegue, teniendo en cuenta el manejo del servicio, las versiones de las librerías del entorno virtual y el resto de módulos personalizados instalados en dicho servidor.
 - El sistema actual proporciona una base sólida para poder implementar la nueva funcionalidad del reconocimiento facial, aunque se debe reestructurar tanto su arquitectura interna para poder implementar mejoras significativas al software.
-

La **Tabla 5** presenta la información más relevante obtenida del análisis del TIC que está orientado al registro local de asistencias mediante reconocimiento facial a los laboratorios de software.

Tabla 5. Informe de revisión de documentos, para el TIC, orientado al registro local de asistencias mediante reconocimiento facial, a los laboratorios de software

Informe de revisión de documentos extendida	
Identificador: RD02	Fecha de la revisión: 10 de noviembre del 2023
Nombre del documento revisado: Prototipo de Reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja [2].	
Introducción: Para poder comprender el contexto actual del sistema, se llevó a cabo una revisión del Trabajo de Integración curricular, así como de su código fuente. Esto permitió obtener una visión general de la situación actual del sistema, tanto a nivel de código fuente como de documentación. Además, también se ejecutó el sistema en un entorno local (computador propio, lo que garantiza un ambiente controlado), consiguiendo así, evaluar el software en su totalidad, dando como resultado, una comprensión elevada de todas sus implicaciones.	
Observaciones:	
<ul style="list-style-type: none"> • Al realizar la detección de vida luego de pasar el reconocimiento facial, se cae el servidor de Odoos. Esto, ya que se satura la memoria RAM que se encuentra designado a Odoos. • El sistema sólo funciona de manera local, es decir que no se desplegó en ningún ambiente externo, como lo puede ser la nube o los servidores de la carrera. • El sistema está desarrollado en Odoos V13 y en Windows 10, lo que significa que la tecnología es una versión más antigua que la del sistema de Control de Acceso (Odoos V15) y el SO no es una distribución de Linux. • El archivo que tiene los pesos para el modelo de detección de vida, no posee la extensión adecuada. Es decir, la extensión es “.py” en vez de “.h5”. • Para poder conseguir que el modelo de detección funcione correctamente, se necesitan 24 frames. Además, procesar dichos frames representan una carga considerable de recursos computacionales. 	

-
- Debido a que el sistema funciona en un ambiente local, se abre la cámara del servidor (la de la computadora) para el reconocimiento facial, y más no una cámara web.
 - El sistema recorre todas las imágenes del servidor en vez de comparar sólo con el usuario que está intentado registrar asistencia. Esto consume muchos más recursos de lo que debería, y dicho consumo es exponencialmente proporcional a la cantidad de usuarios registrados en el software.
 - El sistema se encuentra desarrollado con versiones de librerías muy antiguas. Por tanto, se debe tener en consideración si al actualizar el sistema no se genera problemas de compatibilidad.
 - Existe suficiente justificación de manera cuantificada acerca de la elección de la librería "face_recognition", incluyendo datos comparativos de precisión frente a otros modelos y pruebas exhaustivas que validan el correcto funcionamiento de la librería. Esto significa que no se debe comparar ni evaluar dicho funcionamiento.
-

Conclusiones:

- Se debe hacer uso de la cámara web del cliente, no la cámara local del servidor. Esto permitirá que se abra la cámara web en cada dispositivo y puedan acceder varios usuarios a la vez.
 - Se necesita actualizar el formato del archivo con los pesos que hace uso el modelo de detección de vida a una extensión válida (.h5) para resolver problemas de compatibilidad y funcionalidad.
 - Se debe migrar de Odoov V13 a Odoov V15 para poder conseguir un único software que integre ambos sistemas. Esto también implica actualizar y definir las versiones de las librerías para dicho único software sin que presente problemas de compatibilidad para ninguno de los dos sistemas.
 - Se debe considerar que su implementación es un servidor basado en Linux y no en uno basado en Windows.
 - Se debe optimizar el proceso de reconocimiento facial. Por tanto, se necesita:
 - Implementar una validación directa de la imagen asociada al usuario que intenta registrar asistencia y no recorrer todas las imágenes de todos los usuarios del sistema.
 - Capturar los 24 frames y luego enviarlos al servidor para su validación y verificación.
 - No es necesario realizar pruebas exhaustivas del reconocimiento facial con detección de vida, ya que una de las tesis a integrar, ya realizó dichas pruebas y obtuvo datos precisos que avalan el uso de la tecnología mencionada.
-

6.1.1.4 Actores del sistema

En esta sección, se presenta las funcionalidades que puede realizar cada tipo de usuario (actor) que interactúa directamente con el sistema, los cuales son: estudiante, docente y administrador (encargado de los laboratorios de software/director de la carrera de Computación). Es necesario mencionar que, dichos actores y sus respectivas funcionalidades, se lograron obtener mediante el **Análisis de observación participante**, **Análisis de entrevista** y el análisis de la **Revisión de documentos**. Así mismo, se debe hacer hincapié en que estos actores pueden realizar varias funciones en el sistema, sin embargo, en este trabajo, solo se abordará aquellas funcionalidades que competan a la integración del sistema base (Sistema de Control de Acceso) con el Sistema de Reconocimiento Facial, es decir, aquellas funcionalidades orientadas a la gestión y control de las asistencias Face ID (asistencias mediante reconocimiento facial y detección de vida), así como también a las nuevas funcionalidades que permiten gestionar la imagen de usuario de cada individuo. La **Tabla 6** presenta la descripción de manera general de las funciones que pueden realizar dichos tipos de usuario en el sistema.

Tabla 6. Descripción de las funcionalidades que pueden realizar los actores del sistema

Actor	Descripción
Administrador	<ul style="list-style-type: none"> • Registrar su asistencia. • Ver reporte de sus asistencias y/o de todos los usuarios. • Generar reporte de sus asistencias y/o de todos los usuarios. • Imprimir/descargar reporte de sus asistencias y/o de todos los usuarios. • Registrar usuario con imagen. • Editar usuario con imagen. • Eliminar usuario con imagen. • Ver reporte de usuarios.
Docente	<ul style="list-style-type: none"> • Registrar su asistencia. • Ver reporte de sus asistencias y/o de sus estudiantes. • Generar reporte de sus asistencias y/o de sus estudiantes. • Imprimir/descargar reporte de sus asistencias y/o de sus estudiantes.
Estudiante	<ul style="list-style-type: none"> • Registrar su asistencia. • Ver reporte de sus asistencias. • Generar reporte de sus asistencias. • Imprimir/descargar reporte de sus asistencias.

6.1.1.5 Requisitos del sistema

En esta sección, se presenta los requisitos funcionales y no funcionales del sistema. Es necesario mencionar que, dichos requisitos, son una extensión mucho más detallada de las funcionalidades descritas en la sección de **Actores del sistema**. Además, dichos requisitos, son resultado de la recapitulación de la información recabada mediante el **Análisis de observación participante**, **Análisis de entrevista** y el análisis de la **Revisión de documentos**.

La **Tabla 7** presenta los requisitos funcionales del sistema, abordando su nombre, descripción y prioridad (la prioridad está regida a la importancia de su implementación para el negocio).

Tabla 7. Requisitos funcionales del sistema

Código	Requisito	Descripción	Prioridad
RF001	Refactorizar el sistema de control de acceso	El sistema de control de acceso debe mantener su funcionalidad actual luego de la implementación del patrón Modelo-Vista-Controlador (MVC) y de la reorganización del código. Esto implica la separación de la lógica de negocio en modelos, vistas y controladores distintos, con el objetivo de mejorar la estructura interna y la mantenibilidad del sistema.	Alta
RF002	Actualizar gestión de laboratorio	El sistema permitirá al administrador agregar tres campos adicionales al módulo de gestión de los laboratorios. Estos campos incluirán un campo de texto para capturar la dirección IP y la máscara de red asociadas a cada laboratorio, checkboxes	Alta

		para habilitar/deshabilitar tanto la dirección IP como los equipos informáticos.	
RF003	Obtener UUID	El sistema permitirá al administrador, docente y estudiante, generar un UUID único y aleatorio. Este UUID se almacenará en las cookies del dispositivo que lo está generando, para luego ser utilizado en el registro de un equipo informático.	
RF004	Registrar equipo informático	El sistema permitirá al administrador crear equipos informáticos vinculados a un único laboratorio. Cada equipo informático deberá contar con atributos válidos, que incluyen un UUID único, un identificador único, un estado para determinar su habilitación y una referencia al único laboratorio al que está asociado.	Alta
RF005	Editar equipo informático	El sistema permitirá al administrador editar la información de equipos informáticos existentes. Se permitirá la modificación de los atributos de cada equipo, asegurando que los cambios sean válidos. Estos atributos comprenden un UUID único, un identificador único, un estado para determinar su habilitación y una referencia al único laboratorio al que está asociado.	Alta
RF006	Dar de baja equipo informático	El sistema permitirá al administrador eliminar un equipo informático del sistema.	Media
RF007	Listar equipos informáticos	El sistema permitirá al administrador visualizar una lista de todos los equipos informáticos registrados. Esto incluye la capacidad de buscar equipos informáticos específicos y aplicar varios filtros según las preferencias del usuario, tales como: identificador, UUID, laboratorio al que pertenece y estado del equipo informático.	Alta
RF008	Generar reporte de equipos informáticos	El sistema permitirá al administrador generar un reporte de los equipos informáticos registrados en el sistema, con la capacidad de agrupar los equipos según sea necesario para facilitar su análisis.	Media
RF009	Imprimir reporte de equipos informáticos	El sistema permitirá al administrador imprimir o descargar en formato PDF el reporte de equipos informáticos generado, lo que proporciona una versión física o digital del mismo para su referencia futura.	Media
RF010	Registrar usuario e imagen de usuario	El sistema permitirá al administrador crear un nuevo usuario, junto con su imagen asociada que se almacenará en el servidor. Esta imagen será utilizada posteriormente para el reconocimiento facial.	Alta
RF011	Editar usuario e imagen de usuario	El sistema permitirá al administrador editar la información de un usuario existente, incluyendo la posibilidad de modificar su imagen, lo que actualizaría la imagen almacenada en el servidor.	Alta
RF012	Dar de baja usuario e imagen de usuario	El sistema permitirá al administrador eliminar un usuario del sistema, lo que incluye la eliminación de su imagen asociada almacenada en el servidor.	Media
RF013	Listar usuarios	El sistema permitirá al administrador visualizar una lista de todas las cuentas de usuarios registradas, junto con sus respectivas imágenes almacenadas en el servidor.	Alta
RF014	Registrar asistencia Face ID	El sistema permitirá a los usuarios (docente, estudiante y administrador) registrar su asistencia utilizando la cámara web para el reconocimiento facial. Este proceso se iniciará mediante un botón específico en la interfaz.	Alta

RF015	Listar asistencias Face ID	El sistema permitirá a los usuarios (docente, estudiante y administrador) visualizar una lista que detalle todas las asistencias registradas mediante reconocimiento facial. Esto incluye la capacidad de buscar registros específicos y aplicar varios filtros según las preferencias del usuario, tales como: materia, responsable, usuario, fecha y hora, período académico, dirección ip, UUID, rol y laboratorio.	Alta
RF016	Generar reporte de asistencias Face ID	El sistema permitirá a los usuarios (docente, estudiante y administrador) generar un reporte de asistencias registradas mediante reconocimiento facial, con la capacidad de agrupar los registros según sea necesario para facilitar su análisis.	Media
RF017	Imprimir reporte de asistencias Face ID	El sistema permitirá a los usuarios (docente, estudiante y administrador) imprimir o descargar en formato PDF el reporte de asistencias generado, lo que proporciona una versión física o digital del mismo para su referencia futura.	Media

La **Tabla 8**, presenta los requisitos no funcionales del sistema, abordando la categoría al que pertenece y su descripción.

Tabla 8. Requisitos no funcionales del sistema

Código	Categoría	Descripción
RNF01	Usabilidad	El sistema deberá garantizar la presencia de interfaces intuitivas y amigables, las cuales serán evaluadas a través de la puntuación de satisfacción del usuario obtenida mediante el Modelo de Aceptación Tecnológica (TAM), específicamente con la variable de Facilidad de Uso Percibida (FUP).
RNF02	Seguridad	El sistema deberá garantizar que las nuevas vistas desarrolladas solo sean visibles para los usuarios que tengan los permisos respectivos, los cuales se basan en los roles de administrador, docente y estudiante, de lo contrario, no deberán tener acceso a las mismas. Además, al tener acceso a dichas vistas, sólo podrán realizar las funciones definidas en los permisos del sistema.
RNF03	Fiabilidad	El sistema deberá garantizar que todas las funciones desarrolladas operen correctamente, abarcando la gestión de equipos informáticos, usuarios, laboratorios y asistencias Face ID. En caso de que ocurra algún error de cualquier tipo, el sistema deberá emitir un mensaje que indique el error ocurrido.
RNF04	Desarrollo	El sistema deberá ser desarrollado utilizando las siguientes tecnologías: Odo ERP v17.0, sistema operativo Linux, Python v3.0.0 o superior, y PostgreSQL v11.0 o superior.
RNF05	Rendimiento	El sistema deberá garantizar un tiempo de respuesta para todas las operaciones sea igual o inferior a 3 segundos. En el caso de registrar la asistencia mediante reconocimiento facial, el tiempo de respuesta deberá ser igual o inferior a 10 segundos.

6.1.1.6 Historias de usuario

En esta sección, se presentan las historias de usuarios determinadas en base a los requisitos del sistema abstraídos en la sección **Requisitos del sistema**. Para ello, la sección **Formato de historias de usuario**, presenta el formato utilizado para las historias de usuario,

así como su respectiva leyenda para cada atributo. Además, la sección **Historia de usuario más relevante**, presenta la historia de usuario más relevante de todas las historias de usuario desarrolladas para el sistema. Esto, con la finalidad de que se pueda observar la manera en la que se desarrollaron el resto de historias de usuario (para observar todas las historias de usuario elaboradas, véase sección de **Anexos, apartado ANEXO 3: Historias de Usuario**).

6.1.1.6.1 Formato de historias de usuario

La **Tabla 9** presenta el formato utilizado para la elaboración de las historias de usuario, así como su respectiva leyenda para cada uno de los campos empleados.

Tabla 9. Formato de historia de usuario

Historia de usuario	
Identificador:	Nombre:
Usuario:	
Iteración asignada:	Referencia de requisito:
Fecha de creación:	Fecha de entrega:
Prioridad en el negocio (Alta/Media/Baja):	Riesgo en el desarrollo (Alta/Media/Baja):
Descripción:	
Observaciones:	
Criterios de aceptación:	

Donde:

- **Identificador:** corresponde a un código único para cada historia de usuario.
- **Nombre:** corresponde a una descripción concisa e identificativa de la HU.
- **Usuario:** corresponde al tipo de usuario (actor) que interactúa con el sistema.
- **Interacción Asignada:** corresponde al número de iteración al que se le asignó dicha HU.
- **Referencia de requisito:** corresponde al identificador del requisito al que la HU está asociada.
- **Fecha de creación:** corresponde a la fecha de creación de la HU.
- **Fecha de entrega:** corresponde a la fecha planificada para la entrega (finalización) de la HU.
- **Prioridad en el negocio:** hace referencia a la relevancia o importancia que implica el desarrollo de la HU para el negocio.
- **Riesgo en el desarrollo:** corresponde al riesgo que se puede presentar al momento de desarrollar la HU.
- **Descripción:** detalla todas las tareas específicas a realizar en la HU.

- **Observaciones:** presenta aquellas consideraciones especiales que se deben tomar en cuenta al momento de realizar la HU.
- **Criterios de aceptación:** abarca las condiciones que se deben cumplir y satisfacer para que se pueda considera a la HU como culminada y aceptada.

6.1.1.6.2 Historia de usuario más relevante

La **Tabla 10** presenta la historia de usuario más relevante del sistema, que corresponde al registro de asistencia de los usuarios mediante reconocimiento facial. Cabe recalcar que, esta sección tiene como finalidad mostrar cómo se elaboraron todas las historias de usuario del sistema. Esta es solo una de las muchas historias elaboradas (para ver tanto esta como el resto de las historias de usuario, véase sección de **Anexos, apartado ANEXO 3: Historias de Usuario**).

Tabla 10. Historia de usuario más relevante, denominada “Registrar asistencia Face ID”

Historia de usuario	
Identificador: 014	Nombre: Registrar asistencia Face ID
Usuario: Docente, Estudiante, Administrador	
Iteración asignada: 004	Referencia de requisito: RF014
Fecha de creación: 13/12/2023	Fecha de entrega: 10/01/2024
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Alta
Descripción:	
COMO usuario (docente, estudiante o administrador) QUIERO poder registrar mi asistencia utilizando la cámara web para el reconocimiento facial PARA agilizar el proceso de registro de asistencia y garantizar su precisión.	
Observaciones:	
<ul style="list-style-type: none"> • La capacidad de registrar la asistencia mediante el reconocimiento facial permitirá a los usuarios realizar este proceso de manera rápida y precisa. • Se debe proporcionar un botón claro y accesible en la interfaz para iniciar el proceso de registro de asistencia utilizando la cámara web. • Es esencial que el sistema pueda reconocer y validar la identidad del usuario de manera confiable mediante el reconocimiento facial. 	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1. El sistema debe proporcionar un botón claro y accesible en la interfaz para que los usuarios puedan iniciar el proceso de registro de asistencia mediante el reconocimiento facial. 2. Se debe validar que sólo el administrador puede registrar asistencia a cualquier hora, mientras que, tanto el docente como el estudiante, sólo pueden registrar asistencia siempre y cuando se encuentren dentro del horario establecido. 3. Se debe de validar el registro de asistencia según la configuración del laboratorio, es decir: si está habilitado el rango ip, se valida la dirección ip del equipo; si está habilitado los equipos informáticos, se valida el UUID del equipo; si están habilitados ambos, se validan ambas direcciones; y si están deshabilitadas ambas, no se valida ninguna de las dos direcciones y se registra las direcciones sin importar cuáles sean. 4. Se debe validar el registro de asistencia según la configuración del equipo informático, es decir: si está habilitado el equipo, se podrá registrar asistencia; caso contrario, no se registra asistencia y se presenta que el equipo está deshabilitado. 	

5. Al hacer clic en el botón de registro de asistencia, el sistema debe activar la cámara web del dispositivo del usuario y comenzar el proceso de reconocimiento facial.
6. El sistema debe poder reconocer y validar la identidad del usuario utilizando el reconocimiento facial de manera confiable.
7. Después de completar el proceso de reconocimiento facial, el sistema debe registrar la asistencia del usuario de manera automática y precisa.
8. Se debe proporcionar retroalimentación clara al usuario durante todo el proceso de registro de asistencia mediante el reconocimiento facial.
9. Se realizarán pruebas para verificar que el proceso de registro de asistencia mediante el reconocimiento facial funcione correctamente y de manera confiable.
10. Se documentará el proceso de registro de asistencia mediante el reconocimiento facial en el sistema para futuras referencias y mantenimiento del sistema.

6.1.1.7 Plan de iteración

La **Tabla 11** presenta el plan de iteración seguido para el desarrollo del software bajo la metodología XP, detallando la fecha de creación y entrega de cada iteración. Este plan tiene como objetivo demostrar la distribución de las historias de usuario elaboradas en la sección de **Historias de usuario**. Las historias de usuario se agruparon en cuatro iteraciones, debido a la naturaleza de los requisitos identificados. Los grupos fueron: refactorización del sistema actual (primera iteración), gestión de equipo informático (segunda iteración), gestión de imagen de usuario (tercera iteración) y gestión de asistencias mediante Face ID (cuarta iteración). Cabe destacar que este plan muestra sólo de manera general la organización que se va a llevar a cabo en el desarrollo bajo la metodología XP (para ver el desarrollo completo bajo la metodología XP, véase sección de **Anexos, apartado ANEXO 4: Implementación de metodología XP**).

Tabla 11. Plan de iteración.

#	Historia de usuario	Fecha de creación	Fecha de entrega	Iteración
001	Refactorizar el sistema de control de acceso	13/11/2023	20/11/2023	Primera
	Actualizar gestión de laboratorio			
002	Obtener UUID	21/11/2023	27/11/2023	Segunda
	Registrar equipo informático			
	Editar equipo informático			
	Dar de baja equipo informático			
	Listar equipos informáticos			
	Generar reporte de equipos informáticos			
	Imprimir reporte de equipos informáticos			
003	Registrar usuario e imagen de usuario	28/11/2023	12/12/2023	Tercera
	Editar usuario e imagen de usuario			

	Dar de baja usuario e imagen de usuario			
	Listar usuarios			
004	Registrar asistencia Face ID	13/12/2023	13/01/2024	Cuarta
	Listar asistencias Face ID			
	Generar reporte de asistencias Face ID			
	Imprimir reporte de asistencias Face ID			

6.1.2 Fase de diseño

En esta sección, se presenta la **Arquitectura de software**, los **Diagramas UML**, y los **Prototipos de interfaces gráficas**. Cabe mencionar que, estos diagramas fueron elaborados a partir de las historias de usuario creadas en la sección de **Historias de usuario**. Además, es necesario enfatizar en que esta sección presenta sólo de manera general la fase de diseño del desarrollo bajo la metodología XP (para ver el desarrollo completo de la fase de diseño bajo la metodología XP, véase sección de **Anexos, apartado ANEXO 4: Implementación de metodología XP, fase de diseño de cada iteración.**)

6.1.2.1 Arquitectura de software

La **Figura 4** presenta el diagrama de despliegue y la **Figura 5** presenta el diagrama de componentes.

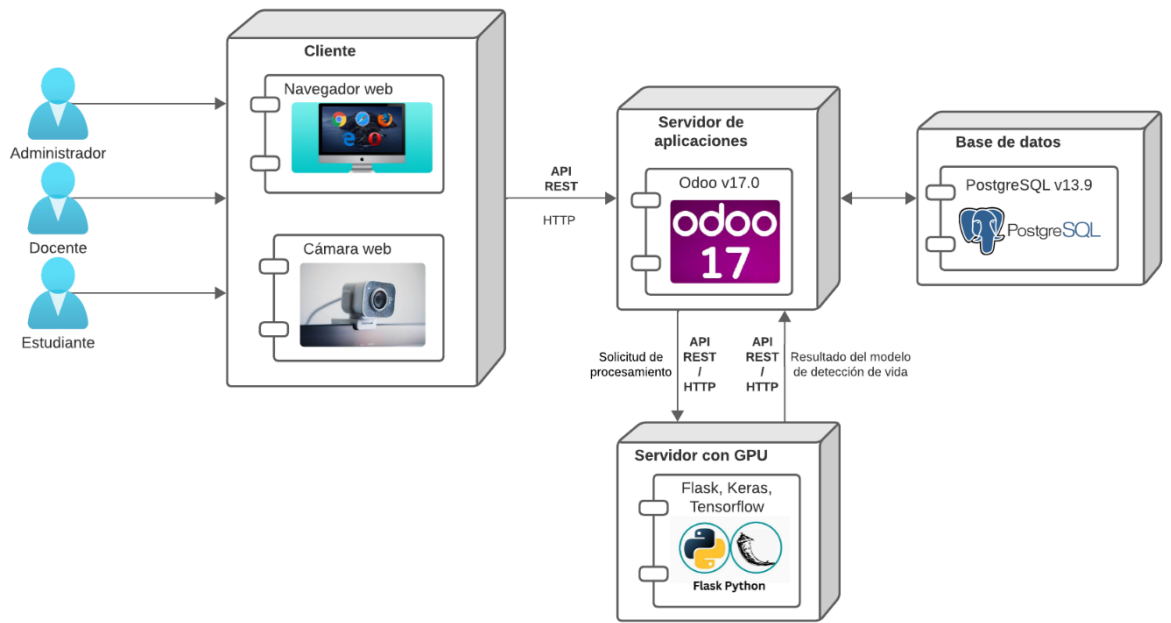


Figura 4. Diagrama de despliegue

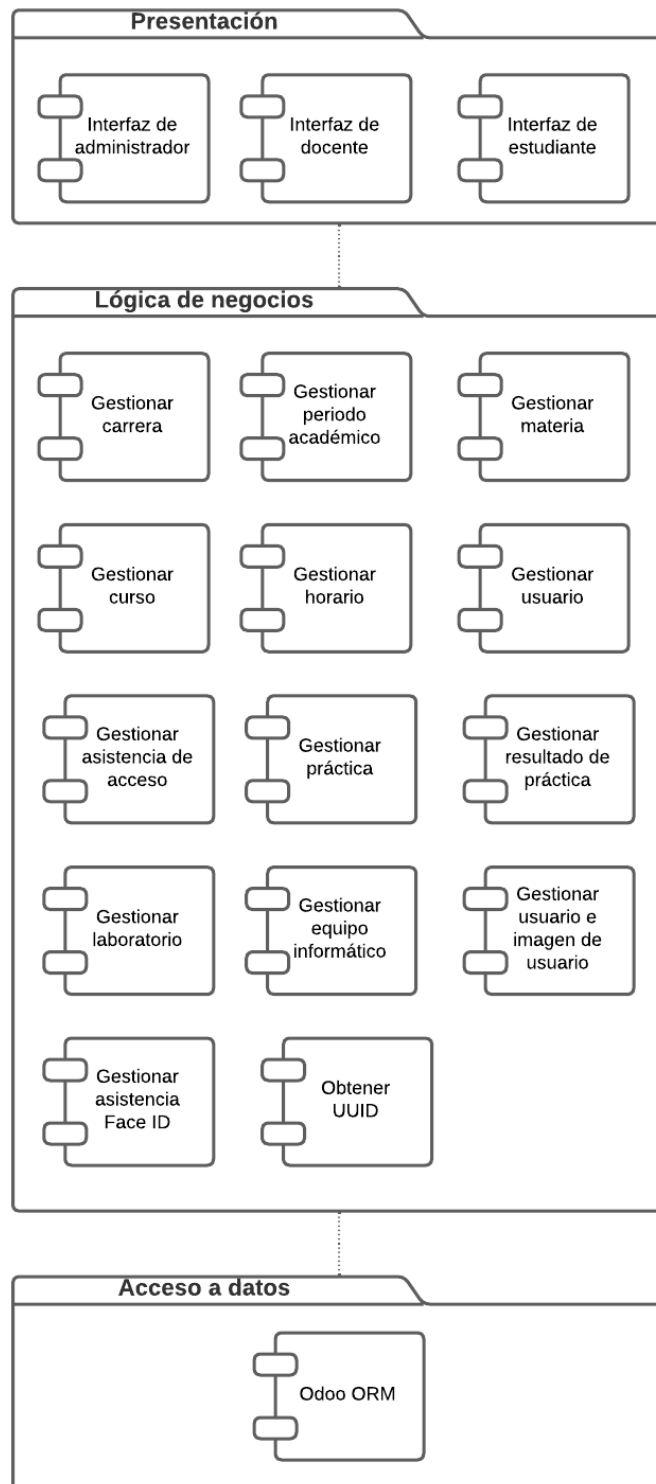


Figura 5. Diagrama de componentes

6.1.2.2 Diagramas UML

La **Figura 6** presenta el diagrama de clases, la **Figura 7** presenta el diagrama de modelo relacional y la **Figura 8**, presenta el diagrama de casos de uso.

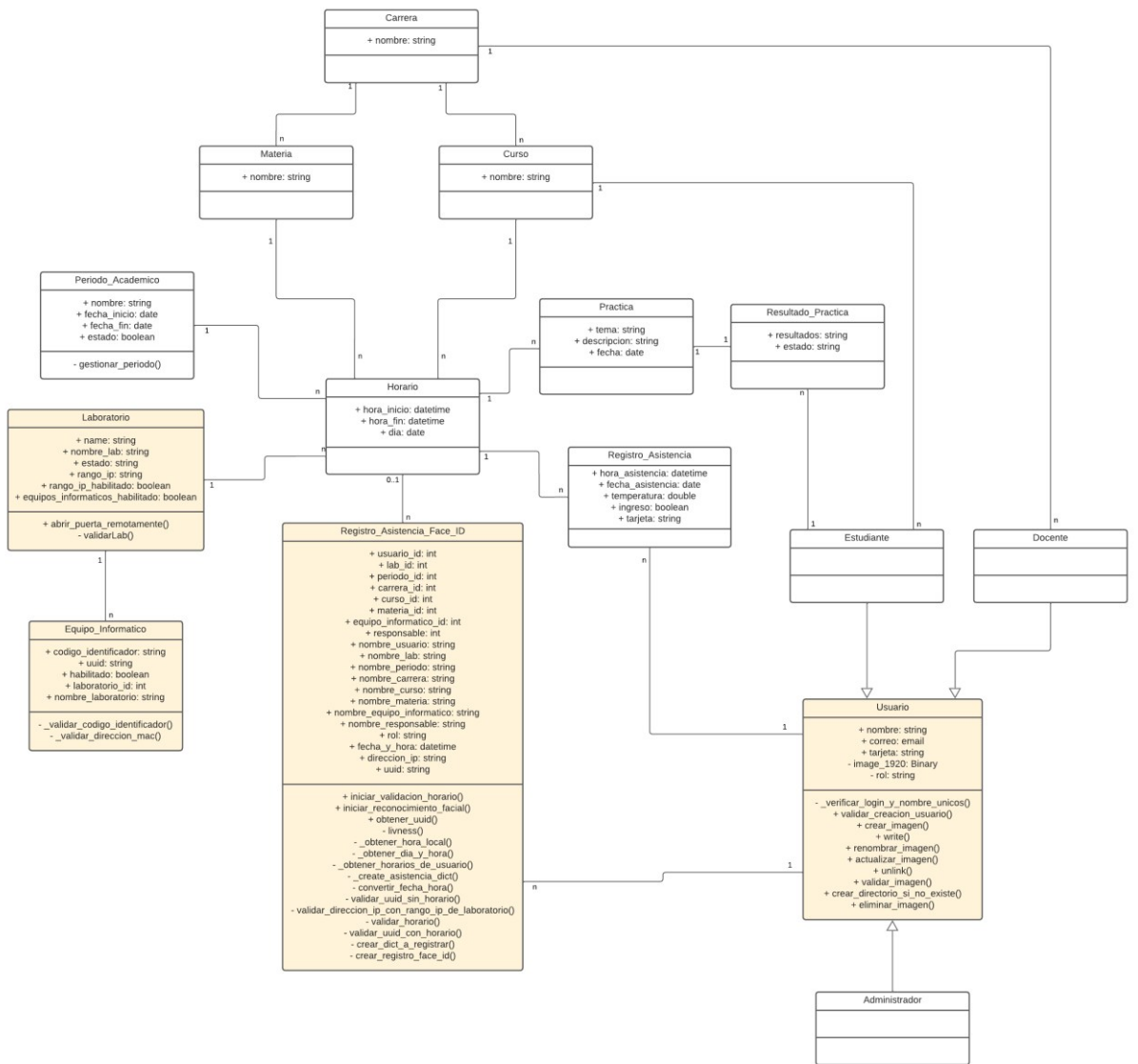


Figura 6. Diagrama de clases

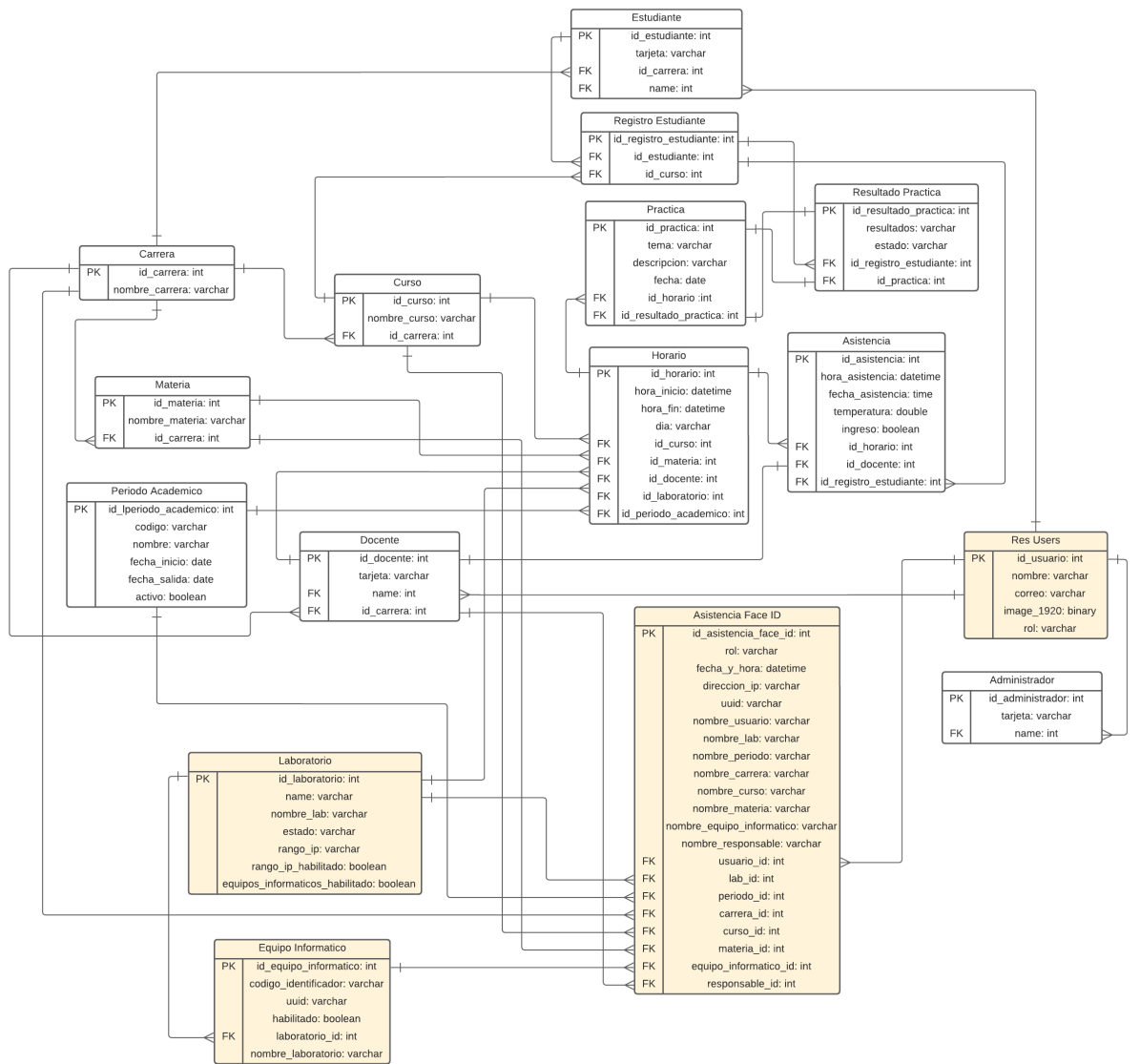


Figura 7. Diagrama Modelo Relacional

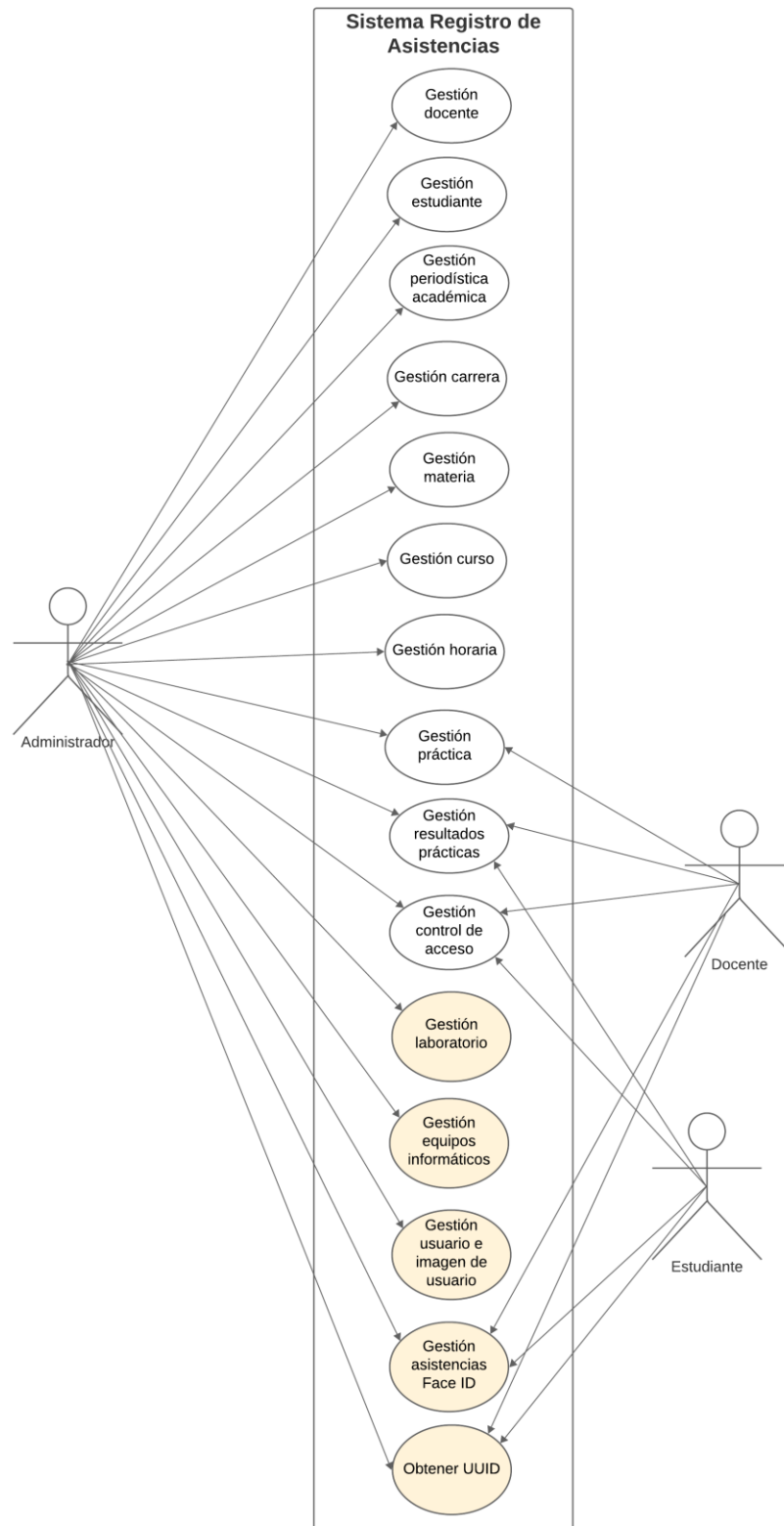


Figura 8. Diagrama de casos de uso

La **Figura 9** presenta el diagrama de actividades más relevante en el sistema, el cual corresponde al registro de asistencia de los usuarios en el sistema mediante reconocimiento facial (para ver tanto el diagrama de actividades mencionado como el resto de diagramas de

actividades, véase sección de **Anexos, apartado ANEXO 4: Implementación de metodología XP, fase de diseño de cada iteración).**

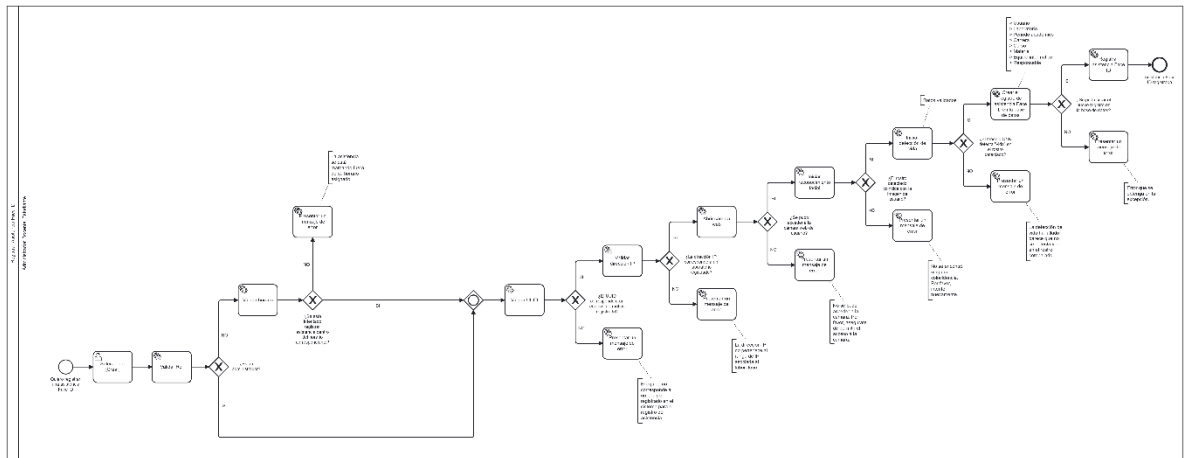


Figura 9. Diagrama de actividades

6.1.2.3 Prototipos de interfaces gráficas

La **Figura 10** presenta el prototipo más relevante en el sistema, el cual corresponde al registro de asistencia de los usuarios en el sistema mediante reconocimiento facial (para ver tanto el prototipo mencionado como el resto de prototipos, véase sección de **Anexos, apartado ANEXO 4: Implementación de metodología XP, fase de diseño de cada iteración).**

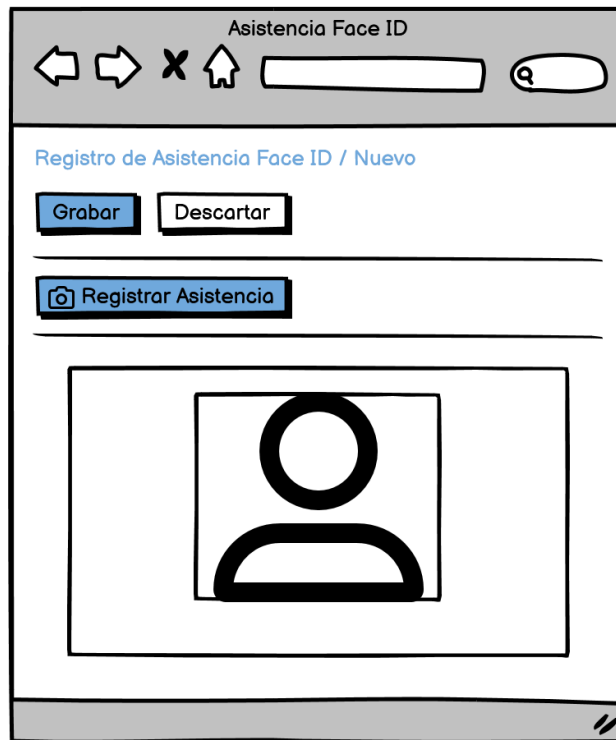


Figura 10. Prototipo de Interfaz Gráfica más relevante del sistema

6.1.3 Fase de codificación

En esta fase, se presenta un resumen general de la codificación realizada en el sistema, destacando principalmente la implementación del patrón Modelo-Vista-Controlador (MVC) que emplea Odoe para el desarrollo de sus módulos (para ver el desarrollo completo de la codificación realizada, véase sección de **Anexos, apartado ANEXO 4: Implementación de metodología XP, fase de codificación de cada iteración**). Por tanto, se aborda: **Estructura del sistema, Modelo, Vista y Controlador**.

Además, se anexa el link del repositorio del sistema alojado en GitHub, el cual es el siguiente: https://github.com/Luis0613Jh/Integracion_Control_Acceso_Face_Recognition.git

6.1.3.1 Estructura del sistema

La **Figura 11** presenta el prototipo a seguir para el desarrollo del sistema en base al patrón MVC.

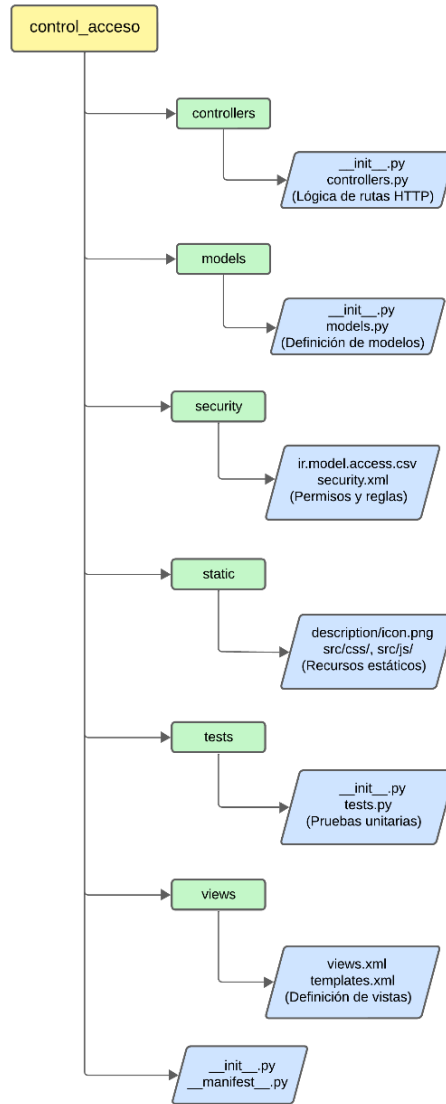


Figura 11. Prototipo de la estructura general del sistema.

La **Figura 12** presenta la estructura general del sistema siguiendo los principios propuestos por el patrón MVC.

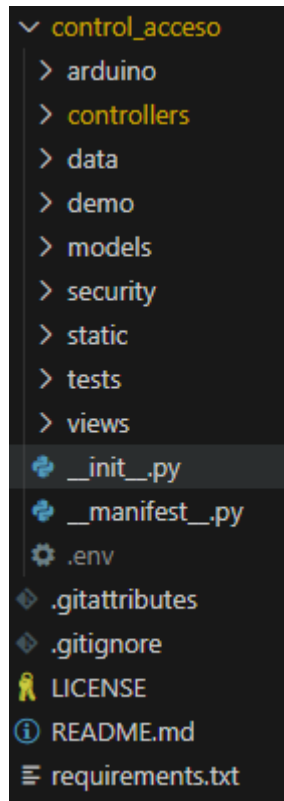


Figura 12. Estructura general del sistema.

6.1.3.2 Modelo

La **Figura 13** presenta el esquema general de los modelos desarrollados.

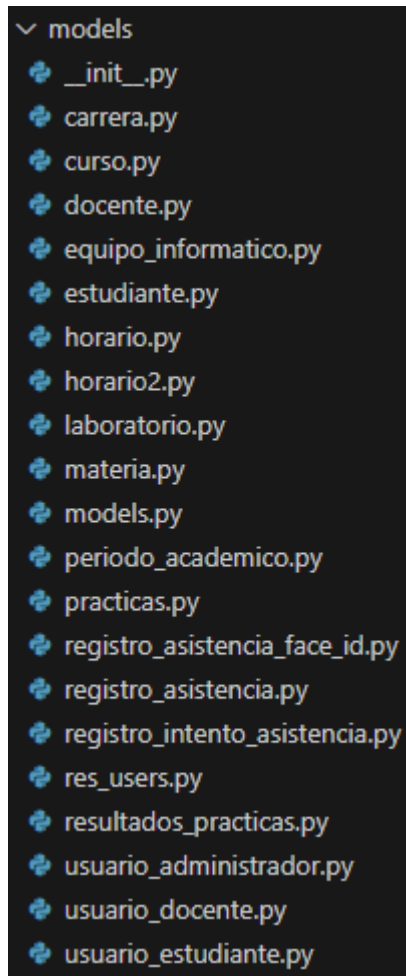


Figura 13 Esquema general de los modelos desarrollados.

La **Figura 14** presenta una muestra de un modelo desarrollado en el sistema.

```

1 from odoo import models, fields, api
2 from odoo.exceptions import ValidationError
3 import re
4 import uuid
5
6 class EquipoInformatico(models.Model):
7     _name = 'controlacceso.equipoinformatico'
8     _description = 'Equipos Informáticos'
9
10    codigo_identificador = fields.Char(string='Código Identificador', required=True, index=True, copy=False, readonly=True, default='Nuevo')
11    uuid = fields.Char(string='UUID', required=True)
12    habilitado = fields.Boolean(string='Habilitado', default=True)
13    laboratorio_id = fields.Many2one('controlacceso.lab', string='Laboratorio', ondelete='cascade', required=True)
14
15    nombre_laboratorio = fields.Char(related='laboratorio_id.nombre_lab', string='Nombre del Laboratorio', store=True)
16
17    # Campo adicional para notas o descripción
18    descripcion = fields.Text(string='Descripción', help='Notas o comentarios adicionales sobre el equipo informático.')
19
20    @api.constrains('codigo_identificador')
21    def _validar_codigo_identificador(self):
22        for equipo in self:
23            # Verificar el formato del código identificador (ejemplo: E001)
24            if not re.match(r'^[A-Za-z]\d{3}$', equipo.codigo_identificador):
25                raise ValidationError('El código identificador debe tener el formato "E001".')
26
27            # Verificar la unicidad del código identificador
28            if self.search([('codigo_identificador', '=', equipo.codigo_identificador), ('id', '!=', equipo.id)]):
29                raise ValidationError('El código identificador debe ser único.')
30
31    @api.model
32    def create(self, vals):
33        # Generar el código identificador incremental
34        if vals.get('codigo_identificador', 'Nuevo') == 'Nuevo':
35            # Usar una secuencia para generar el código incremental
36            seq = self.env['ir.sequence'].next_by_code('controlacceso.equipoinformatico') or '/'
37            vals['codigo_identificador'] = f'E{seq.zfill(3)}' # Formato E001, E002, etc.
38
39        return super(EquipoInformatico, self).create(vals)
40
41    @api.constrains('uuid')
42    def _validar_uuid(self):
43        for equipo in self:
44            try:
45                # Verificar si el formato del UUID es válido
46                uuid_obj = uuid.UUID(equipo.uuid, version=4) # Solo acepta UUID versión 4
47            except ValueError:
48                raise ValidationError('El UUID no tiene un formato válido.')
49
50            # Verificar si el UUID ya existe en otros equipos
51            if self.search([('uuid', '=', equipo.uuid), ('id', '!=', equipo.id)]):
52                raise ValidationError('El UUID debe ser único. Por favor, vuelva a generar un nuevo UUID.')

```

Figura 14. Muestra del modelo desarrollado denominado “EquipoInformatico”.

6.1.3.3 Vista

La Figura 15 presenta el esquema general de las vistas desarrolladas.

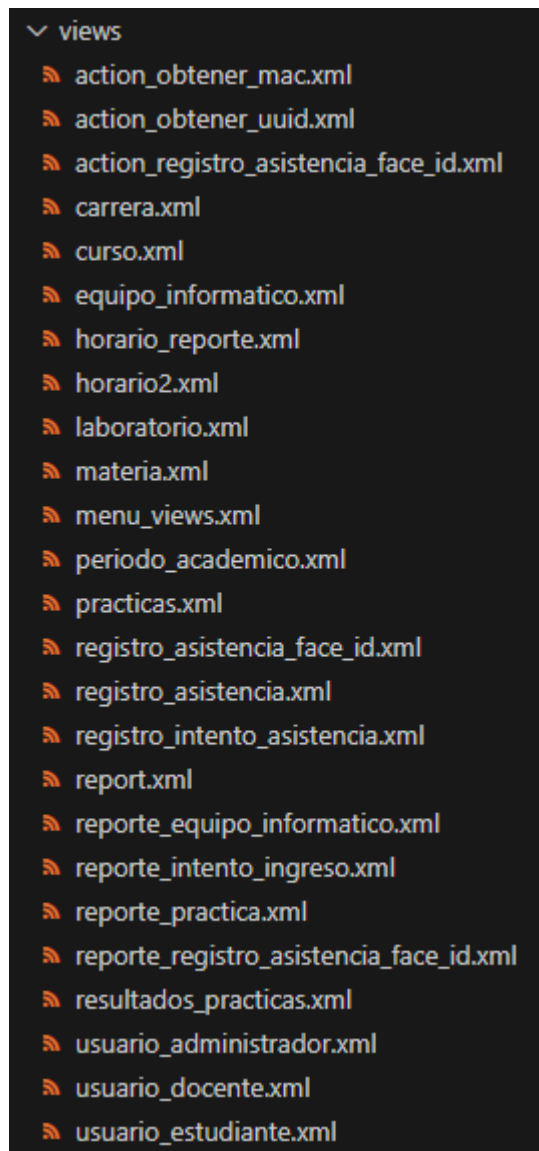


Figura 15 Esquema general de las vistas desarrolladas.

La **Figura 16** presenta una muestra de una vista desarrollada en el sistema.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <odoo>
3   <data>
4     <!-- Vista del formulario de equipos informáticos -->
5     <record id="view_form_equipoinformatico" model="ir.ui.view">
6       <field name="name">equipoinformatico.form</field>
7       <field name="model">controlacceso.equipoinformatico</field>
8       <field name="arch" type="xml">
9         <form>
10          <sheet>
11            <group>
12              <field name="codigo_identificador" widget="char" style="width:50%;" placeholder="Ejemplo: E001" invisible="1"/>
13              <field name="uuid" widget="char" style="width:50%;" placeholder="Ejemplo: e068cd67-6ac3-4a48-8665-cca2f9ae5b39"/>
14              <field name="descripcion" widget="char" placeholder="Notas o comentarios adicionales sobre el equipo informático."/>
15              <field name="laboratorio_id"/>
16              <field name="habilitado"/>
17            </group>
18          </sheet>
19        </form>
20      </field>
21    </record>
22
23    <!-- Vista del árbol de equipos informáticos -->
24    <record id="view_tree_equipoinformatico" model="ir.ui.view">
25      <field name="name">equipoinformatico.tree</field>
26      <field name="model">controlacceso.equipoinformatico</field>
27      <field name="arch" type="xml">
28        <tree>
29          <field name="codigo_identificador"/>
30          <field name="uuid"/>
31          <field name="descripcion"/>
32          <field name="nombre_laboratorio"/>
33          <field name="habilitado"/>
34        </tree>
35      </field>
36    </record>
37
38    <!-- Vista de búsqueda de equipos informáticos -->
39    <record id="view_search_equipoinformatico" model="ir.ui.view">
40      <field name="name">equipoinformatico.search</field>
41      <field name="model">controlacceso.equipoinformatico</field>
42      <field name="arch" type="xml">
43        <search>
44          <field name="codigo_identificador" string="Código Identificador"/>
45          <field name="uuid" string="UUID"/>
46          <field name="descripcion" string="Descripción"/>
47          <field name="nombre_laboratorio" string="Laboratorio"/>
48          <field name="habilitado" string="Habilitado"/>
49        </search>
50      </field>
51    </record>
52
53    <!-- Acción para abrir la vista de equipos informáticos -->
54    <record id="action_view_equipoinformatico" model="ir.actions.act_window">
55      <field name="name">Equipos Informáticos</field>
56      <field name="res_model">controlacceso.equipoinformatico</field>
57      <field name="view_mode">tree,form</field>
58      <field name="view_id" ref="view_tree_equipoinformatico"/>
59      <field name="search_view_id" ref="view_search_equipoinformatico"/>
60      <field name="help" type="html">
61        <p class="oe_view_nocontent_create">
62          Click para crear un nuevo equipo informático.
63        </p>
64      </field>
65    </record>
66  </data>
67 </odoo>
68

```

Figura 16. Muestra de la vista desarrollada denominada “EquipoInformatico”.

6.1.3.4 Controlador

La Figura 17 presenta el esquema general de los controladores desarrollados.

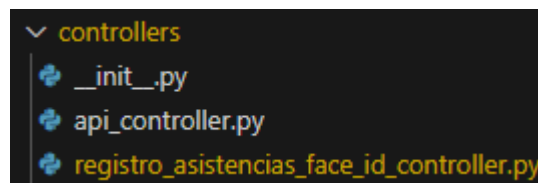


Figura 17 Esquema general de las vistas desarrolladas.

La Figura 18 presenta una muestra de un controlador desarrollado en el sistema.


```

1 @http.route('/api/iniciar_validacion_horario', type='json', methods=['POST'], auth='user', csrf=False)
2 def iniciar_validacion_horario(self, **kwargs):
3     try:
4
5         usuario_actual = request.env.user
6         # Capturar el contenido completo de kwargs
7         kwargs_data = kwargs.get('kwargs')
8
9         # Obtener el UUID
10        uuid_recibido = kwargs_data.get('uuid_recibido')
11
12        if not uuid_recibido:
13            return self.error_response(-1, 'El UUID no fue proporcionado.')
14
15        _logger.info(f'UUID recibido: {uuid_recibido}')
16
17        # Validar si el usuario SI es administrador
18        if usuario_actual.has_group('control_acceso.res_groups_administrador'):
19            # Si es administrador, puede continuar sin validar horarios
20            asistencia_dict = self.validar_uuid_sin_horario(usuario_actual, uuid_recibido)
21            if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
22                # Caso de error
23                return asistencia_dict
24            else:
25                # Caso exitoso
26                asistencia_dict = self.validar_direccion_ip_con_rango_ip_de_laboratorio(asistencia_dict)
27
28                if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
29                    # Caso de error
30                    return asistencia_dict
31                else:
32                    return self.success_response(1, "Datos validados para el administrador", asistencia_dict)
33        # Validar si el usuario SI es docent
34        elif usuario_actual.has_group('control_acceso.res_groups_docentes'):
35            # Lógica para docentes
36            asistencia_dict = self.validar_horario("Docente")
37
38            if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
39                # Caso de error
40                return asistencia_dict
41            else:
42                # Caso exitoso
43                asistencia_dict = self.validar_uuid_con_horario(asistencia_dict, uuid_recibido)
44
45                if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
46                    # Caso de error
47                    _logger.error(f"Error detectado MAC CON HORARIO: {asistencia_dict['mensaje']}")
48                    return asistencia_dict
49                else:
50                    asistencia_dict = self.validar_direccion_ip_con_rango_ip_de_laboratorio(asistencia_dict)
51                    if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
52                        # Caso de error
53                        _logger.error(f"Error detectado DIRECCION IP CON HORARIO: {asistencia_dict['mensaje']}")
54                        return asistencia_dict
55                    else:
56                        return self.success_response(1, "Datos validados para el administrador", asistencia_dict)
57        elif usuario_actual.has_group('control_acceso.res_groups_alumnos'):
58            # Lógica para estudiantes
59            asistencia_dict = self.validar_horario("Estudiante")
60
61            if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
62                # Caso de error
63                _logger.error(f"Error detectado HORARIO: {asistencia_dict['mensaje']}")
64                return asistencia_dict
65            else:
66                # Caso exitoso
67                asistencia_dict = self.validar_uuid_con_horario(asistencia_dict, uuid_recibido)
68
69                if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
70                    # Caso de error
71                    _logger.error(f"Error detectado MAC CON HORARIO: {asistencia_dict['mensaje']}")
72                    return asistencia_dict
73                else:
74                    asistencia_dict = self.validar_direccion_ip_con_rango_ip_de_laboratorio(asistencia_dict)
75                    if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
76                        # Caso de error
77                        _logger.error(f"Error detectado DIRECCION IP CON HORARIO: {asistencia_dict['mensaje']}")
78                        return asistencia_dict
79                    else:
80                        return self.success_response(1, "Datos validados para el administrador", asistencia_dict)
81        else:
82            return self.error_response(-1, "Horario NO Válido, el usuario no pertenece a ningún grupo en el sistema")
83    except Exception as e:
84        return self.error_response(-1, str(e))

```

Figura 18. Muestra del controlador desarrollado denominado “Registro Asistencias Face Id” para “Validar Horario”.

6.1.4 Fase de pruebas

En esta fase, se presenta un resumen general de la **Prueba de validación operativa del sistema en un ambiente controlado**, con la finalidad de simular un escenario real en donde se ejecutará el software. Además, se presenta un resumen de las **Pruebas unitarias** realizadas para comprobar el correcto funcionamiento del sistema. Del mismo modo, se presenta un resumen de las **Pruebas de aceptación** que permiten validar el que el sistema cumple con el alcance definido.

6.1.4.1 Prueba de validación operativa del sistema en un ambiente controlado

La **Figura 19** presenta el reporte de asistencias obtenido en la demostración del sistema en vivo, el cual contó con la presencia del docente a cargo de la materia "Programación Orientada a Objetos", Ing. Edison Coronel; el director de la carrera, Ing. Pablo Ordoñez; el encargado de los laboratorios, Ing. Luis Sinche; y el autor del Trabajo de Integración Curricular (tesista), Luis Jumbo (para ver el desarrollo completo de la prueba de validación operativa, véase sección de **Anexos, apartado ANEXO 9: Pruebas de validación operativas del sistema en un entorno controlado**).

Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Jessica Catalina Rivas Jimenez	Estudiante	Laboratorio de software	14/01/2025 11:27:31	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.200.40	76f944d-7f54-4b36-a049-090f13bb9d8b
Luis Alejandro Blacio Torres	Estudiante	Laboratorio de software	14/01/2025 11:33:03	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.204.155	6618f16c-9348-4518-8b76-5070034fc026
Cristhian Danghelo Valverde Villamagua	Estudiante	Laboratorio de software	14/01/2025 11:31:36	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.200.42	36ec2590-6e18-42c4-b83d-e5848f650dc6
Viviana Elizabeth Córdova Celi	Estudiante	Laboratorio de software	14/01/2025 11:33:08	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.204.159	5fca322d-c52d-4285-9a21-07a17867310a
Mathías Fabricio Medina Aponte	Estudiante	Laboratorio de software	14/01/2025 11:33:16	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.204.158	47b539ec-7955-40a4-a489-1d000292913f
Miguel Angel Veintimilla Esparza	Estudiante	Laboratorio de software	14/01/2025 11:33:27	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.200.46	9cb584a6-0838-4718-9a1d-e0b72c663aab
Yober Edison Gaona Gaona	Estudiante	Laboratorio de software	14/01/2025 11:33:32	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.204.154	2abfa909-eb1e-4fa2-9144-e21dc0d3101c
Fabricio Josue Ruiz Aguilar	Estudiante	Laboratorio de software	14/01/2025 11:33:36	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.200.70	4db0c669-06d3-4032-9589-b0e90930f5ee
Ana Cristina Panamirto Flores	Estudiante	Laboratorio de software	14/01/2025 11:33:36	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.204.157	f1f32ec8-6217-4b86-946a-c2ec96839bde
Juan Alberto Calpino Martinez	Estudiante	Laboratorio de software	14/01/2025 11:34:45	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.200.55	449a8859-2750-4697-8ffc-b1671f542db
Steeven Daniel Pardo Lima	Estudiante	Laboratorio de software	14/01/2025 11:34:07	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.200.75	a10297ac-3c1c-4864-9be7-66d38c524015
Gerardo David Naula Saraguro	Estudiante	Laboratorio de software	14/01/2025 11:34:48	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.204.156	c8a1aa37-41c8-4b17-aca5-6a8b699ed3f5
Daniel Alejandro Saavedra Jaramillo	Estudiante	Laboratorio de software	14/01/2025 11:37:17	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.200.53	5163753f-d605-4a20-aadd-b0a730d7ad44
Izrael Sebastián Rodríguez Torres	Estudiante	Laboratorio de software	14/01/2025 11:38:37	sep-feb-2025	Computación	Segundo	Programación Orientada a Objetos	Edison Leonardo Coronel Romero	10.20.204.145	dca4176-b2ad-4503-ae68-ee2c77172b2e

Figura 19. Reporte de asistencias en la demostración en vivo.

6.1.4.2 Pruebas unitarias

La **Tabla 12** presenta el resumen general de las pruebas unitarias realizadas (para ver el desarrollo completo de las pruebas unitarias, véase sección de **Anexos, apartado ANEXO 4: Implementación de metodología XP, fase de pruebas de cada iteración**).

Tabla 12. Resumen general de las pruebas unitarias realizadas

ID	Descripción	Estado
PU001	Validar que un laboratorio se registra correctamente	Aprobado
PU002	Validar que no se puede registrar un laboratorio con IP no válida	Aprobado
PU003	Validar que un laboratorio se edita correctamente	Aprobado
PU004	Validar que no se puede editar un laboratorio con IP no válida	Aprobado
PU005	Validar que un laboratorio se puede dar de baja (eliminar) correctamente	Aprobado
PU006	Validar que se listan correctamente todos los laboratorios	Aprobado
PU007	Validar que un equipo informático se registre correctamente	Aprobado
PU008	Validar que un equipo informático no se registre con código inválido	Aprobado
PU009	Validar que un equipo informático se edite correctamente	Aprobado
PU010	Validar que un equipo informático no se edite con UUID inválido	Aprobado
PU011	Validar que un equipo informático se pueda dar de baja	Aprobado
PU012	Validar que se puedan listar los equipos informáticos	Aprobado
PU013	Validar que un usuario se registre correctamente	Aprobado
PU014	Validar que un usuario no se registre sin imagen válida	Aprobado
PU015	Validar que un usuario se edite correctamente	Aprobado
PU016	Validar que un usuario no se edite con nombre duplicado	Aprobado
PU017	Validar que un usuario se puede dar de baja	Aprobado
PU018	Validar que se puedan listar usuarios	Aprobado
PU019	Validar que el equipo no está habilitado para el registro de asistencia	Aprobado
PU020	Validar que el equipo no está registrado para el registro de asistencia	Aprobado
PU021	Validar que no se encontraron interfaces de red	Aprobado
PU022	Validar registro exitoso de un equipo habilitado para el registro de asistencia	Aprobado
PU023	Validar que la dirección IP del equipo no pertenece al laboratorio asociado para el registro de asistencia.	Aprobado
PU024	Validar que no se puede obtener la dirección IP del equipo para el registro de asistencia.	Aprobado
PU025	Validar el registro de asistencia cuando el rango IP del laboratorio está deshabilitado.	Aprobado
PU026	Validar el registro de asistencia cuando la dirección IP del equipo se encuentra dentro del rango IP del laboratorio.	Aprobado

PU027	Validar rechazo de registro por falta de horario de estudiante	Aprobado
PU028	Validar rechazo de asistencia ya registrada por estudiante	Aprobado
PU029	Validar registro exitoso de estudiante con horario correcto	Aprobado
PU030	Validar rechazo de registro por falta de horario de docente	Aprobado
PU031	Validar rechazo de asistencia ya registrada por docente	Aprobado
PU032	Validar registro exitoso de docente con horario correcto	Aprobado

6.1.4.3 Pruebas de aceptación

La **Tabla 13** presenta el resumen general de las pruebas de aceptación realizadas a los clientes directos del proyecto (para ver el desarrollo completo de las pruebas de aceptación, véase sección de **Anexos, apartado ANEXO 10: Pruebas de aceptación realizadas al director de la carrera de computación, Ing. Pablo Ordoñez, y ANEXO 11: Pruebas de aceptación realizadas al encargado de los laboratorios de software de la carrera de computación, Ing. Luis Sinche**).

Tabla 13. Resumen general de las pruebas de aceptación realizadas

HU #	Historia de usuario	Estado
HU01	Refactorizar el sistema de control de acceso	Aprobado
HU02	Actualizar gestión de laboratorio	Aprobado
HU03	Obtener UUID	Aprobado
HU04	Registrar equipo informático	Aprobado
HU05	Editar equipo informático	Aprobado
HU06	Dar de baja equipo informático	Aprobado
HU07	Listar equipos informáticos	Aprobado
HU08	Generar reporte de equipos informáticos	Aprobado
HU09	Imprimir reporte de equipos informáticos	Aprobado
HU10	Registrar usuario e imagen de usuario	Aprobado
HU11	Editar usuario e imagen de usuario	Aprobado
HU12	Dar de baja usuario e imagen de usuario	Aprobado
HU13	Listar usuarios	Aprobado
HU14	Registrar asistencia Face ID	Aprobado
HU15	Listar asistencias Face ID	Aprobado
HU16	Generar reporte de asistencias Face ID	Aprobado
HU17	Imprimir reporte de asistencias Face ID	Aprobado

6.1.5 Fase de lanzamiento

En esta fase, se presenta el conjunto de manuales de usuario necesarios para entregar el sistema, los cuales abarcan: **Manual de usuario para administrador**, **Manual de usuario para docente**, **Manual de usuario para estudiante** y **Manual de instalación**. para la instalación en los servidores de los laboratorios de software de la carrera de computación.

6.1.5.1 Manual de usuario para administrador

El manual de usuario perteneciente al rol de administrador en el sistema, se puede encontrar en la sección de **Anexos, apartado ANEXO 6: Manual de usuario para administrador**.

6.1.5.2 Manual de usuario para docente

El manual de usuario perteneciente al rol de docente en el sistema, se puede encontrar en la sección de **Anexos, apartado ANEXO 7: Manual de usuario para docente**.

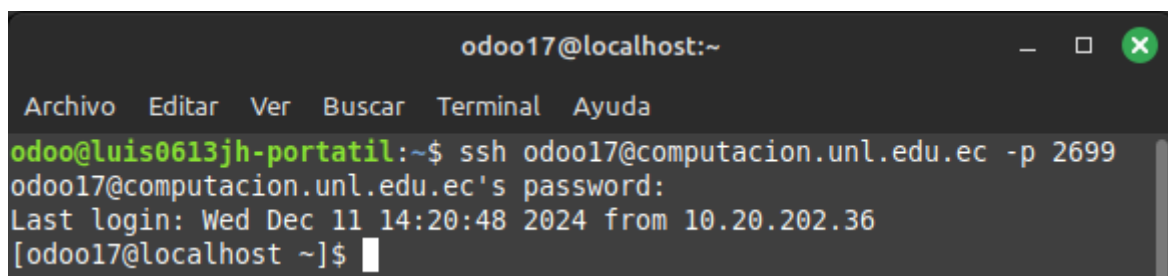
6.1.5.3 Manual de usuario para estudiante

El manual de usuario perteneciente al rol de estudiante en el sistema, se puede encontrar en la sección de **Anexos, apartado ANEXO 8: Manual de usuario para estudiante**.

6.1.5.4 Manual de instalación

Debido a que el sistema desarrollado es una actualización del sistema de “Control de Acceso”, el cual se encuentra ya en funcionamiento en un ambiente de producción, no se detallará toda la instalación desde cero en un servidor de Odoo, sino, de cómo se puede actualizar un módulo existente. A continuación, se presenta la documentación respectiva para dicha actualización en los servidores de la carrera de Computación de la Universidad Nacional de Loja.

La **Figura 20** muestra la conexión mediante SSH a los servidores de la carrera de Computación.



```
odoo17@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
odoo@luis0613jh-portatil:~$ ssh odoo17@computacion.unl.edu.ec -p 2699  
odoo17@computacion.unl.edu.ec's password:  
Last login: Wed Dec 11 14:20:48 2024 from 10.20.202.36  
[odoo17@localhost ~]$
```

Figura 20. Conexión a los servidores de la carrera de Computación mediante SSH.

La **Figura 21** y **Figura 22** muestran la actualización del código fuente del módulo “control_acceso” en el servidor.

```

odoo@luis0613jh-portatil:~$ scp -P 2699 -r /opt/odoo/Integracion_Control_Acceso_Face_Recognition/control_acceso odoo17@computacion.unl.edu.ec:/opt/odoo17/odoo17/custom-addons/control_acceso despliegue
odoo17@computacion.unl.edu.ec's password:
  _init_.py 100% 71 7.7KB/s 00:00
  lr_model_access.xml 100% 41KB 2.3MB/s 00:00
  res_groups.xml 100% 809 64.9KB/s 00:00
  _init_.py 100% 163 9.7KB/s 00:00
  test_laboratorio.py 100% 3305 148.2KB/s 00:00
  _init_.cpython-310.pyc 100% 376 26.6KB/s 00:00
  test_registro_asistencia_face_id.cpython-310.pyc 100% 9891 469.8KB/s 00:00
  test_usuario_e_imagen_de_usuario.cpython-310.pyc 100% 3378 411.4KB/s 00:00
  test_equipo_informatico.cpython-310.pyc 100% 3354 327.8KB/s 00:00
  test_laboratorio.cpython-310.pyc 100% 3110 728.5KB/s 00:00
  test_usuario_e_imagen_de_usuario.py 100% 2719 659.9KB/s 00:00
  test_equipo_informatico.py 100% 3602 784.6KB/s 00:00
  test_registro_asistencia_face_id.py 100% 21KB 891.9KB/s 00:00
  _init_.cpython-310.pyc 100% 209 48.8KB/s 00:00
  cron.xml 100% 590 122.8KB/s 00:00
  model.py 100% 4315KB 458.8KB/s 00:09
  model.h5 100% 4315KB 653.8KB/s 00:06
  Luis_Sinche.png 100% 101KB 545.2KB/s 00:00
  Luis_Jumbo.png 100% 101KB 291.7KB/s 00:00
  control_acceso_lab_2.ino 100% 13KB 243.6KB/s 00:00
  control_acceso_lab.ino 100% 13KB 311.1KB/s 00:00
  _init_.py 100% 107 25.0KB/s 00:00
  api_controller.py 100% 27KB 308.7KB/s 00:00
  _init_.cpython-310.pyc 100% 56KB 1.6MB/s 00:00
  controllers.cpython-310.pyc 100% 286 33.5KB/s 00:00
  registro_asistencias_face_id_controller.cpython-310.pyc 100% 14KB 1.7MB/s 00:00
  api_controller.cpython-310.pyc 100% 13KB 1.9MB/s 00:00
  .env 100% 14KB 531.0KB/s 00:00
  materia.xml 100% 332 81.9KB/s 00:00
  usuario_administrador.xml 100% 1199 273.6KB/s 00:00
  carrera.xml 100% 1342 362.0KB/s 00:00
  reporte_equipo_informatico.xml 100% 647 147.1KB/s 00:00
  menu_views.xml 100% 3412 274.2KB/s 00:00
  reporte_practica.xml 100% 5200 964.5KB/s 00:00
  equipo_informatico.xml 100% 3826 222.0KB/s 00:00
  equipo_informatico.xml 100% 2991 363.0KB/s 00:00

```

Figura 21. Actualización del código fuente del módulo “control_acceso” en el servidor.

```

registro_asistencia.py 100% 1168 225.1KB/s 00:00
registro_intento_asistencia.py 100% 582 93.4KB/s 00:00
registro_asistencia_face_id.py 100% 2893 153.0KB/s 00:00
equipo_informatico.cpython-310.pyc 100% 1967 99.1KB/s 00:00
registro_asistencia.cpython-310.pyc 100% 1185 240.4KB/s 00:00
usuario_estudiante.cpython-310.pyc 100% 2098 282.9KB/s 00:00
resultados_practicas.cpython-310.pyc 100% 3919 204.5KB/s 00:00
carrera.cpython-310.pyc 100% 939 43.7KB/s 00:00
test_open_camera.cpython-310.pyc 100% 638 70.2KB/s 00:00
horario.cpython-310.pyc 100% 1185 147.8KB/s 00:00
usuario_administrador.cpython-310.pyc 100% 681 61.8KB/s 00:00
periodo_academico.cpython-310.pyc 100% 1499 73.2KB/s 00:00
_init_.cpython-310.pyc 100% 801 194.7KB/s 00:00
estudiante.cpython-310.pyc 100% 1050 241.3KB/s 00:00
usuario_docente.cpython-310.pyc 100% 1539 89.5KB/s 00:00
curso.cpython-310.pyc 100% 1389 330.8KB/s 00:00
registro_asistencia_face_id.cpython-310.pyc 100% 2022 298.0KB/s 00:00
horario2.cpython-310.pyc 100% 2587 162.8KB/s 00:00
practicas.cpython-310.pyc 100% 2802 481.7KB/s 00:00
res_users.cpython-310.pyc 100% 5755 694.6KB/s 00:00
open_camera_js.cpython-310.pyc 100% 1069 62.6KB/s 00:00
registro_intento_asistencia.cpython-310.pyc 100% 861 90.6KB/s 00:00
laboratorio.cpython-310.pyc 100% 2716 161.5KB/s 00:00
materia.cpython-310.pyc 100% 809 133.0KB/s 00:00
docente.cpython-310.pyc 100% 1282 253.0KB/s 00:00
estudiante.py 100% 729 155.5KB/s 00:00
horario.py 100% 1395 265.7KB/s 00:00
periodo_academico.py 100% 1121 99.9KB/s 00:00
resultados_practicas.py 100% 4788 364.8KB/s 00:00
models.py 100% 26KB 274.2KB/s 00:00
usuario_estudiante.py 100% 1891 390.3KB/s 00:00
docente.py 100% 920 234.3KB/s 00:00
usuario_docente.py 100% 1456 77.2KB/s 00:00
practicas.py 100% 2707 541.2KB/s 00:00
carrera.py 100% 544 161.0KB/s 00:00
materia.py 100% 418 87.6KB/s 00:00
horario2.py 100% 3894 765.9KB/s 00:00
demo.xml 100% 954 260.2KB/s 00:00
odoo@luis0613jh-portatil:~$

```

Figura 22. Actualización del código fuente del módulo “control_acceso” en el servidor.

La Figura 23 muestra la instalación de los requerimientos para que el código funcione correctamente mediante el archivo de requerimientos (requirements.txt).

```

(odoo17-venv-python3.12) [odoo17@localhost odoo17]$ ls
addons          COPYRIGHT      debian        LICENSE      odoo         README.md    SECURITY.md   setup.cfg
CONTRIBUTING.md  custom-addons  doc          MANIFEST.in  odoo-bin     requirements.txt  setup        setup.py
(odoo17-venv-python3.12) [odoo17@localhost odoo17]$ pip install -r requirements.txt

```

Figura 23. Instalación de los requerimientos que necesita el sistema.

La Figura 24 muestra la página de administración en Odoo de la carrera de Computación (solamente es visible para aquellos usuarios con rol de administrador en dicha página).

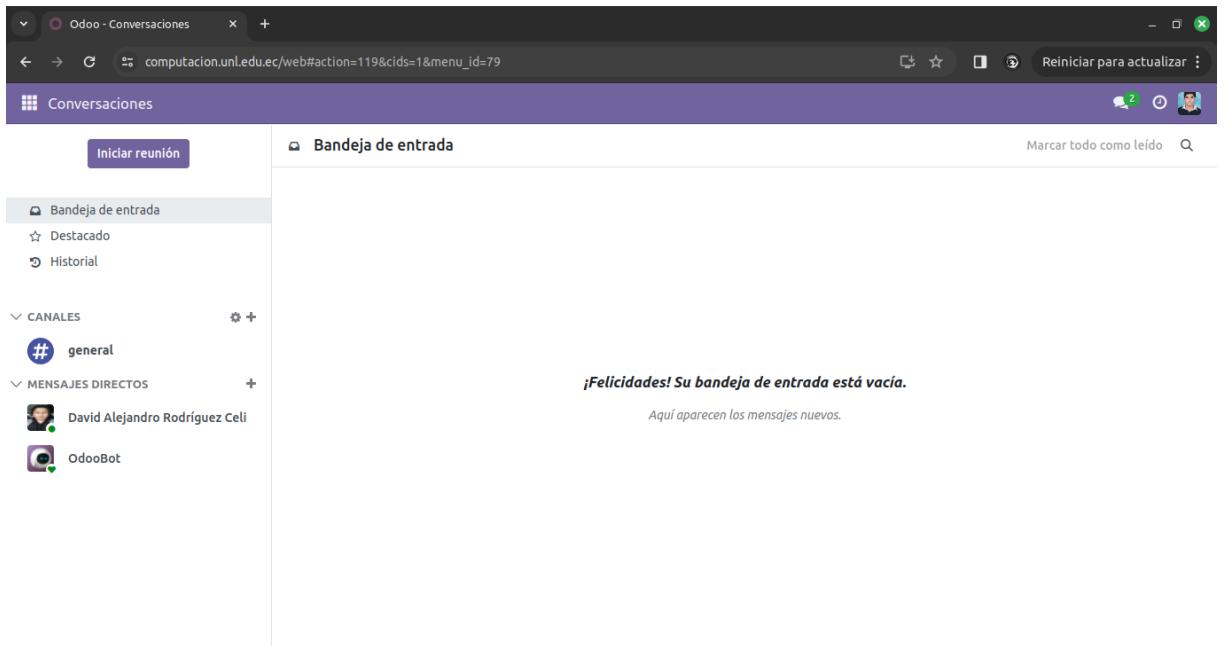


Figura 24. Página de administración en Odoo de la carrera de Computación.

La **Figura 25** muestra la actualización de las aplicaciones en el servidor.

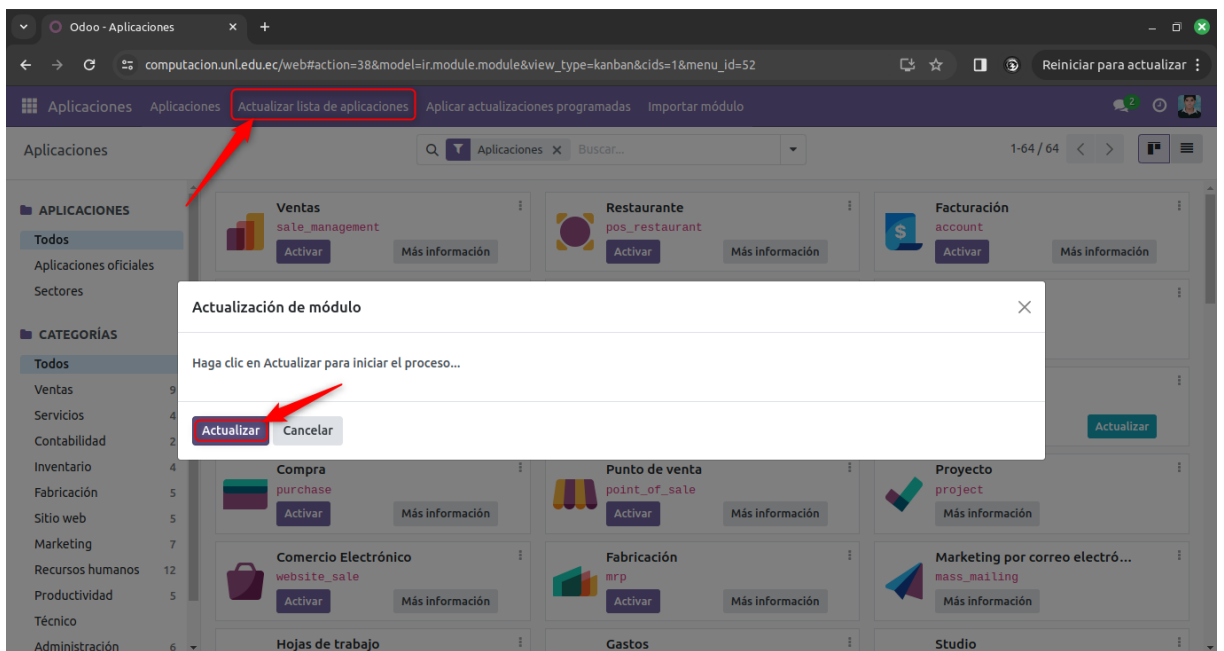


Figura 25. Actualización de las aplicaciones en el servidor.

La **Figura 26** muestra la actualización del módulo “control_acceso”.

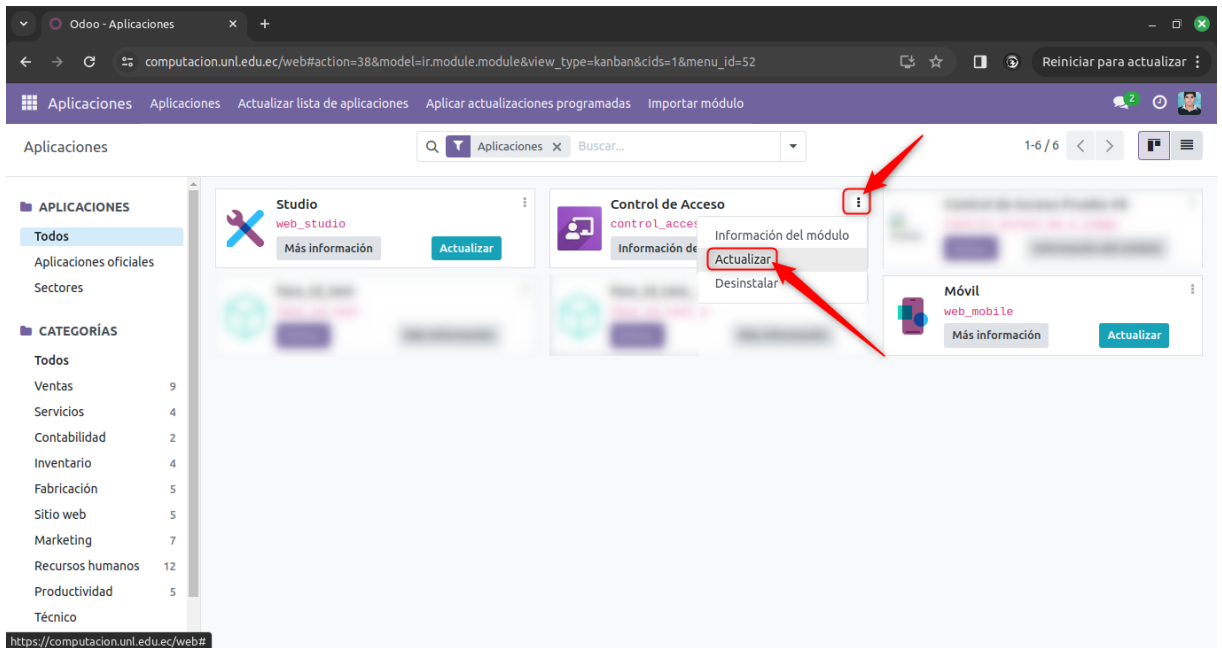


Figura 26. Actualización del módulo “control_acceso”.

La **Figura 27** muestra el módulo “control_acceso” funcionando correctamente luego de la actualización.

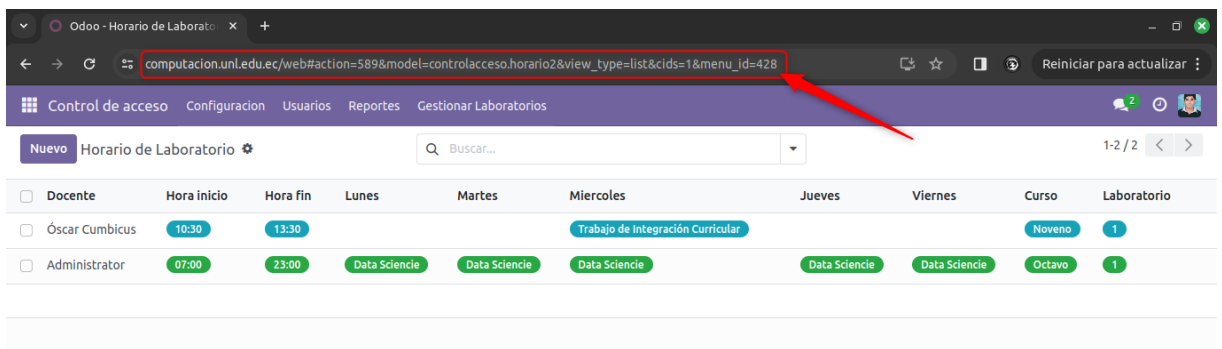


Figura 27. Módulo “control_acceso” funcionando correctamente luego de la actualización.

La **Figura 28** muestra la funcionalidad del registro de asistencia mediante reconocimiento facial con detección de vida funcionando correctamente en el servidor de la carrera de Computación de la Universidad Nacional de Loja.

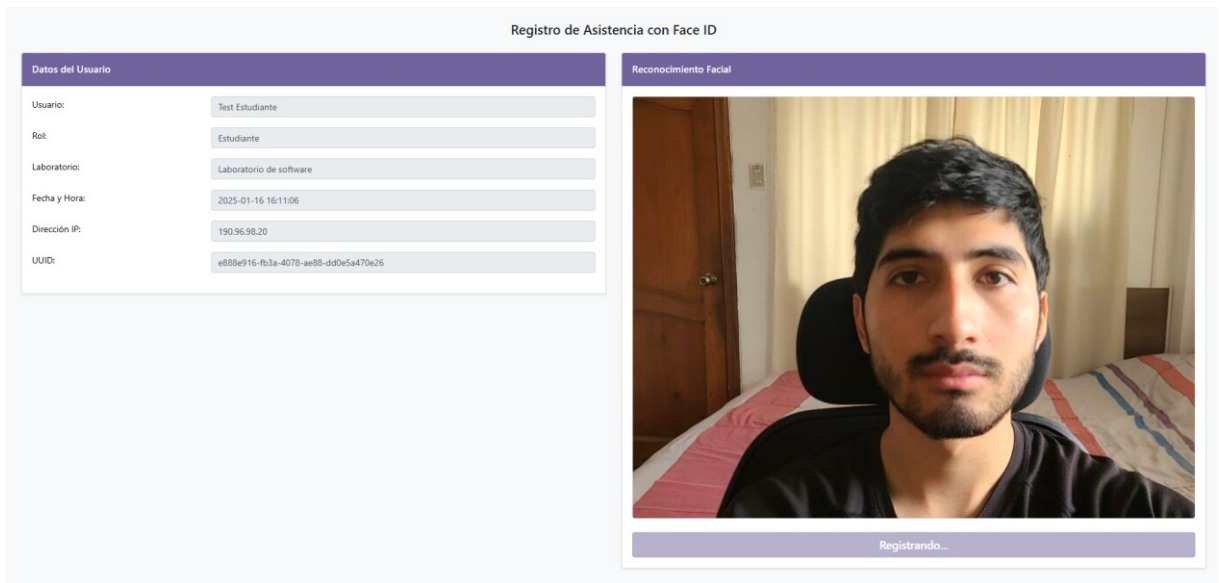


Figura 28. Funcionamiento correcto para registrar asistencia mediante reconocimiento facial en el servidor de producción.

6.2 Objetivo Específico 2. Determinar el nivel de aceptación tecnológica del software para el registro de asistencia en los laboratorios de software mediante el modelo TAM

Para cumplir con el segundo objetivo específico, se aplicó el Modelo de Aceptación Tecnológica (TAM), el cual, se dividió en tres fases, las cuales son: elaboración, medición y análisis [24]. A continuación, se presentan los resultados obtenidos por cada fase mencionada, los cuales van desde la **Fase de elaboración**, hasta la **Fase de análisis**.

6.2.1 Fase de elaboración

En esta fase, se presenta el cuestionario elaborado para poder determinar la percepción del usuario al hacer uso del sistema. La **Tabla 14** muestra cada pregunta lindada a un identificador, y el conjunto de preguntas pertenecientes a cada constructo (variable de TAM). La **Tabla 15** muestra las ponderaciones cuantificadas para cada respuesta del cuestionario según la escala de Likert.

Tabla 14. Cuestionario para evaluar el nivel de aceptación tecnológica (TAM)

Constructo	Identificador	Preguntas
Utilidad percibida (UP)	UP1	¿Está de acuerdo que el uso del registro de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software mejora la gestión y control de los usuarios que hacen uso de dicha instalación?
	UP2	¿Está de acuerdo que el proceso de registro de asistencia en el laboratorio mediante reconocimiento facial y detección de vida es

		más sencillo y práctico de realizar que los métodos manuales que se llevan a cabo actualmente?
	UP3	¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio es realmente útil para controlar las asistencias de los usuarios que hacen uso del laboratorio de software?
	FUP1	¿Está de acuerdo que es fácil de utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software?
Facilidad de uso percibida (FUP)	FUP2	¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio reduce el tiempo necesario para el registro de asistencias?
	FUP3	¿Está de acuerdo que el sistema web que gestiona las asistencias y reportes en el laboratorio de software es fácil de navegar y encontrar la información necesaria?
	AU1	¿Se siente cómodo/a al utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software?
	AU2	¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es beneficioso para gestionar y controlar los usuarios que hacen uso del laboratorio de software?
Actitud hacia el uso (AU)	AU3	¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software, es una buena idea en general?
	AU4	¿Considera agradable el uso del sistema web que gestiona las asistencias y reportes en el laboratorio de software?
	AU5	¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es una opción viable y conveniente para el registro de asistencia en el laboratorio de software en comparación con el método que se lleva actualmente?
Intención de uso (IU)	IU1	¿Tiene usted la intención de utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida dentro del laboratorio de software?

IU2	¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es una opción beneficiosa en el futuro, considerándola como una alternativa viable en el registro de asistencias dentro del laboratorio de software?
IU3	¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida mejorará la experiencia de mantener el control de las asistencias en el laboratorio de software?

Tabla 15. Ponderaciones para cada respuesta del cuestionario

Respuesta	Valor asignado
Totalmente en desacuerdo	0
En desacuerdo	1
Ni de acuerdo, ni en desacuerdo	2
De acuerdo	3
Totalmente de acuerdo	4

6.2.2 Fase de medición

En esta fase, se presenta los resultados obtenidos tras aplicar la encuesta mediante Google Forms que se basó en el cuestionario elaborado en la **Fase de elaboración** (para ver los resultados tal y como lo presenta Google Forms, véase sección de **Anexos, apartado ANEXO 5: Resultados de la aceptación tecnológica del sistema mediante TAM**). La **Tabla 16** muestra la matriz de resultados que recopila las respuestas seleccionadas por la muestra de 42 estudiantes de manera clara y organizada.

Tabla 16. Matriz de resultados en base a cada constructo

Constructo	Identificador	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo, ni en desacuerdo	De acuerdo	Totalmente de acuerdo
Utilidad percibida (UP)	UP1	0%	0%	9,50%	33,30%	57,10%
	UP2	0%	0%	7,10%	35,70%	57,10%
	UP3	0%	0%	4,80%	45,20%	50%
	Total (TUP)	0%	0%	7,13%	38,07%	54,73%
Facilidad de uso percibida (FUP)	FUP1	0%	0%	19%	40,50%	40,50%
	FUP2	0%	0%	14,30%	40,50%	45,20%
	FUP3	0%	0%	2,40%	50%	47,60%
	Total (TFUP)	0%	0%	12%	43,67%	44,43%

Actitud hacia el uso (AU)	AU1	0%	2,40%	19%	40,50%	38,10%
	AU2	0%	0%	7,10%	40,50%	52,40%
	AU3	0%	0%	4,80%	38,10%	57,10%
	AU4	0%	2,40%	4,80%	50%	42,90%
	AU5	0%	0%	4,80%	40,50%	54,80%
	Total (TAU)	0%	1%	8%	41,92%	49,06%
Intención de uso (IU)	IU1	0%	2,40%	16,70%	45,20%	35,70%
	IU2	0%	0%	4,80%	45,20%	50%
	IU3	0%	0%	4,80%	40,50%	54,80%
	Total (TIU)	0%	1%	8,77%	43,63%	46,83%

6.2.3 Fase de análisis

En esta fase, se presenta el análisis de la matriz de resultados obtenida en la **Fase de medición**. Para ello, se analizó cada pregunta de manera individual, y al final, un análisis general de la variable, esto se siguió para cada constructo.

6.2.3.1 Análisis de resultados para el constructo de Utilidad Percibida (UP)

La **Figura 29** presenta los resultados obtenidos para el constructo de Utilidad Percibida (UP).

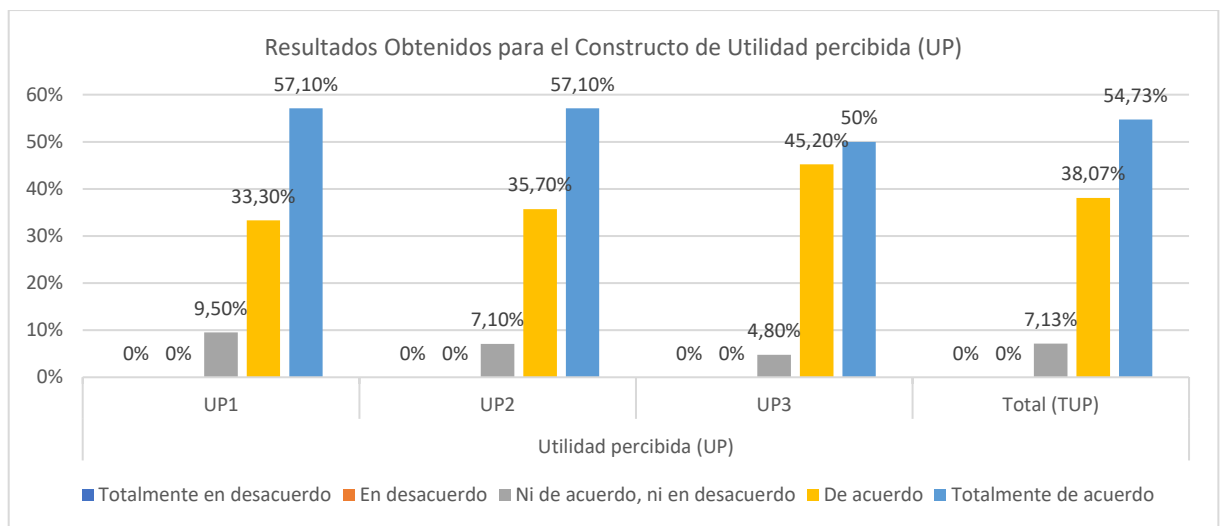


Figura 29. Resultados obtenidos por cada ítem del constructo de Utilidad Percibida (UP) de acuerdo a la escala de Likert

El ítem UP1, relacionado con la pregunta "¿Está de acuerdo que el uso del registro de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software mejora la gestión y control de los usuarios que hacen uso de dicha instalación?", reveló que el 57,10% de los usuarios encuestados están totalmente de acuerdo en que la implementación del

sistema mejora la gestión y control de los usuarios del laboratorio. Además, el 33,30% de los encuestados están de acuerdo con la afirmación, lo que sugiere una predisposición generalizada hacia la percepción positiva del sistema. Por otro lado, el 9,50% indicó no tener una posición clara al respecto. Estos resultados sugieren que la mayoría de los encuestados considera que la implementación del sistema es beneficiosa, mejorando así la gestión y control de los usuarios que hacen uso de la instalación.

El ítem UP2, relacionado con la pregunta "¿Está de acuerdo que el proceso de registro de asistencia en el laboratorio mediante reconocimiento facial y detección de vida es más sencillo y práctico de realizar que los métodos manuales que se llevan a cabo actualmente?", reveló que el 57,10% de los encuestados están totalmente de acuerdo en que la implementación del sistema vuelve el proceso de registro de asistencias más sencillo y práctico. Además, el 35,70% de los encuestados están de acuerdo con la afirmación, lo que sugiere una percepción generalizada de la simplicidad y practicidad del nuevo proceso. Por otro lado, el 7,10% indicó no tener una posición clara al respecto. Estos resultados refuerzan la idea de una fuerte percepción positiva del proceso automatizado y su impacto en la simplificación y practicidad del registro de asistencias en el laboratorio.

El ítem UP3, relacionado con la pregunta "¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio es realmente útil para controlar las asistencias de los usuarios que hacen uso del laboratorio de software?", reveló que el 50,0% de los encuestados están totalmente de acuerdo en que la implementación del sistema es realmente útil para controlar las asistencias. Además, el 45,20% de los encuestados están de acuerdo con la afirmación, lo que sugiere una alta aceptación y percepción positiva de la utilidad del sistema para el control de asistencias. Por otro lado, el 4,80% indicó no tener una posición clara al respecto. Estos resultados respaldan la percepción positiva de la utilidad del sistema para el control de asistencias en el laboratorio.

El ítem TUP, correspondiente al total de la variable de Utilidad Percibida, muestra que el 54,73% de los participantes están totalmente de acuerdo con la utilidad percibida del sistema propuesto en todas sus facetas, mientras que el 38,07% están de acuerdo. Esta alta proporción sugiere una predisposición generalizada y positiva hacia la aceptación del sistema. Por otro lado, el 7,13% no expresó una posición clara al respecto. En conjunto, estos resultados indican una aceptación generalizada y positiva del sistema propuesto en términos de utilidad percibida.

6.2.3.2 Análisis de resultados para el constructo de Facilidad de Uso Percibida (FUP)

La **Figura 30** presenta los resultados obtenidos para el constructo de Facilidad de Uso Percibida (FUP).

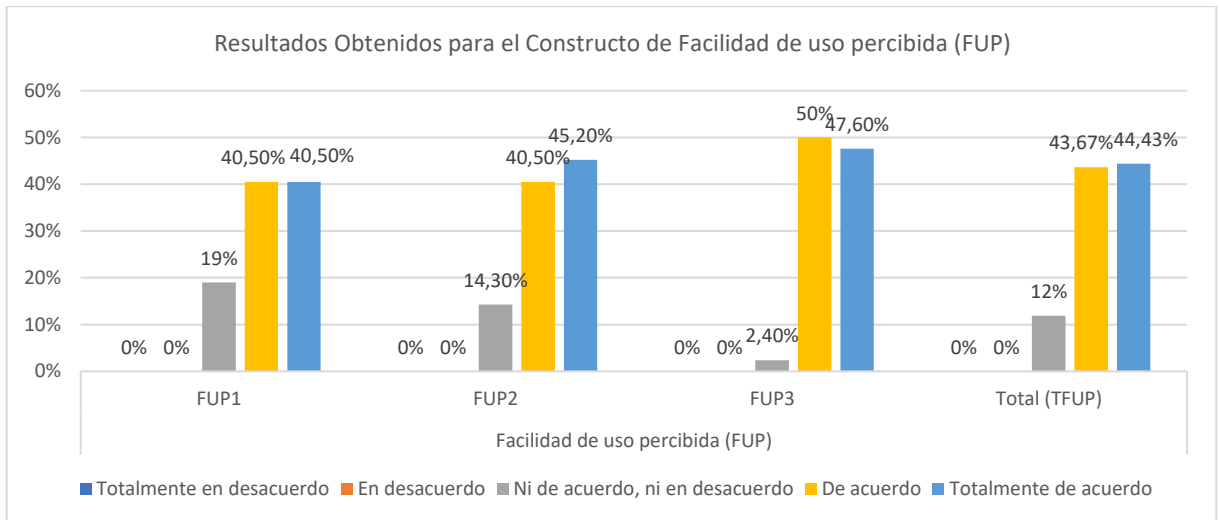


Figura 30. Resultados obtenidos por cada ítem del constructo de Facilidad de Uso Percibida (FUP) de acuerdo a la escala de Likert

El ítem FUP1, relacionado con la pregunta "¿Está de acuerdo que es fácil de utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software?", reveló que el 40,50% de los encuestados están totalmente de acuerdo y, el 40,50%, están de acuerdo con la facilidad de uso del sistema en general, lo que sugiere una percepción buena acerca de la facilidad de uso del sistema. Por otro lado, el 19% restante indicó no tener una posición concreta, es decir, que no están 'ni de acuerdo, ni en desacuerdo'. Al analizar los resultados obtenidos, se determina que la mayoría de los encuestados expresó estar 'totalmente de acuerdo' o 'de acuerdo'. Esto representa un buen indicador de la facilidad de uso del sistema, aunque también se debe considerar que existe un porcentaje considerable que no posee una opinión objetiva clara y evidente, lo que permite asumir que existen aún áreas potenciales de mejora con respecto a la facilidad de uso percibida.

El ítem FUP2, relacionado con la pregunta "¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida reduce el tiempo necesario para el registro de asistencias?", reveló que el 45,20% de los encuestados están totalmente de acuerdo y, el 40,50%, están de acuerdo con la facilidad de uso del sistema en general, lo que sugiere una buena percepción por parte de los encuestados en relación a la reducción del tiempo necesario para el registro de asistencia en el sistema. No obstante, el 14,30% restante indicó no tener una posición concreta, es decir, que no están 'ni de acuerdo, ni en desacuerdo'. Al analizar los resultados obtenidos, se determina que la mayoría de los encuestados expresó estar 'totalmente de acuerdo' o 'de acuerdo'. Esto representa un buen indicador de la facilidad de uso del sistema, aunque también se debe considerar que existe un porcentaje considerable que no posee una opinión objetiva clara y evidente, lo que permite

asumir que existen aún áreas potenciales de mejora con respecto a la reducción del tiempo necesario para el registro de asistencia en el sistema.

El ítem FUP3, relacionado con la pregunta "¿Está de acuerdo que el sistema web que gestiona las asistencias y reportes en el laboratorio de software es fácil de navegar y encontrar la información necesaria?", reveló que el 47,60% de los encuestados están totalmente de acuerdo y, el 50%, están de acuerdo con la facilidad de uso del sistema en general, lo que sugiere una buena percepción por parte de los encuestados en relación a la facilidad de navegar y encontrar la información necesaria en el sistema. Sin embargo, el 2,40% restante indicó no tener una posición concreta, es decir, que no están 'ni de acuerdo, ni en desacuerdo'. Al analizar los resultados obtenidos, se determina que la mayoría de los encuestados expresó estar 'totalmente de acuerdo' o 'de acuerdo'. Esto representa un excelente indicador de la facilidad de uso del sistema, ya que la parte que no posee una opinión clara, es muy baja (2,40%), lo que permite asumir que existen aún áreas potenciales de mejora con respecto a la facilidad de navegar y encontrar la información necesaria en el sistema.

El ítem TFUP, correspondiente al total de la variable de Facilidad de Uso Percibida, denota que el 44,43% de los encuestados están totalmente de acuerdo y, el 43,67%, están de acuerdo con la facilidad de uso del sistema en general, lo que sugiere una buena percepción por parte de los encuestados en relación a la facilidad de uso del sistema. Por otro lado, el 12% restante indicó no tener una posición concreta, es decir, que no están 'ni de acuerdo, ni en desacuerdo'. Al analizar los resultados obtenidos para este constructo, se determina que la mayoría de los encuestados expresó estar 'totalmente de acuerdo' o 'de acuerdo'. Esto representa un excelente indicador de la facilidad de uso percibida del sistema propuesto, aunque aún existen varios aspectos que se pueden mejorar para poder optimizar aún más la experiencia de usuario.

6.2.3.3 Análisis de resultados para el constructo de Actitud hacia el Uso (AU)

La **Figura 31** presenta los resultados obtenidos para el constructo de Actitud hacia el uso (AU).

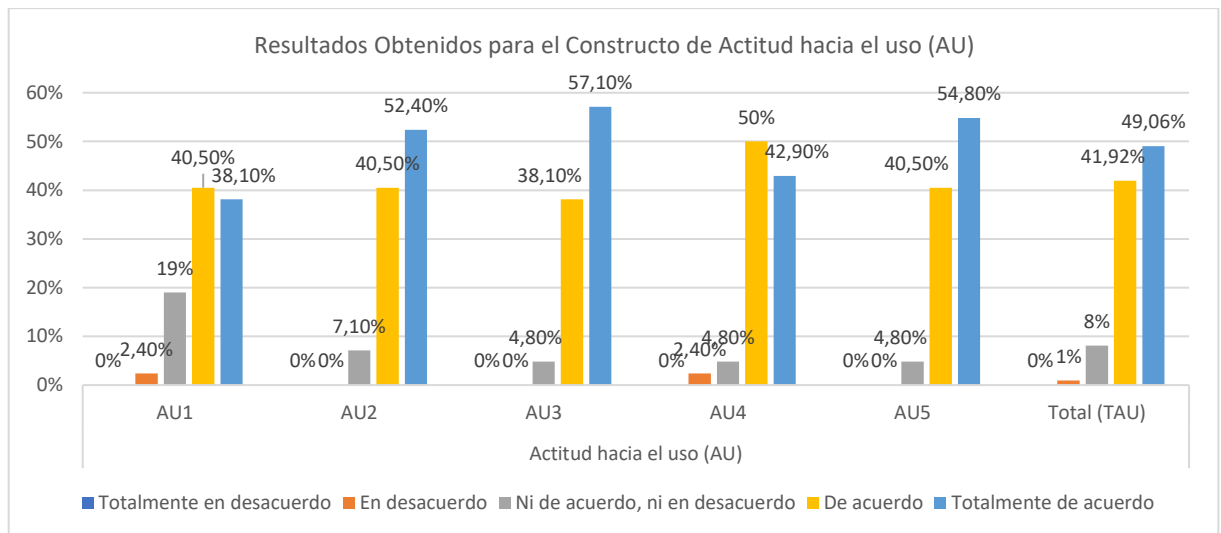


Figura 31. Resultados obtenidos por cada ítem del constructo de Actitud hacia el uso (AU) de acuerdo a la escala de Likert

El ítem AU1, relacionado con la pregunta "¿Se siente cómodo/a al utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software?", reveló que el 38,10% de los encuestados están totalmente de acuerdo y, el 40,50%, están de acuerdo con sentirse cómodos al utilizar el sistema, lo que sugiere una percepción buena acerca de la actitud hacia el uso del sistema. Por otro lado, el 2,40% restante indicó estar en desacuerdo. Al analizar los resultados obtenidos, se determina que la mayoría de los encuestados expresó estar 'totalmente de acuerdo' o 'de acuerdo'. Esto representa un buen indicador de la actitud hacia el uso, es decir, que se sienten cómodos al hacer uso del sistema, pero existe una porción significativa que no se encuentra convencida de hacer uso del mismo. Esto indica que se debe mejorar la comodidad de los usuarios al hacer uso del sistema, implementando correcciones que mejoren la satisfacción del mismo.

El ítem AU2, relacionado con la pregunta "¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es beneficioso para gestionar y controlar los usuarios que hacen uso del laboratorio de software?", reveló que el 52,40% de los encuestados están totalmente de acuerdo y, el 40,50%, están de acuerdo que implementar el sistema representaría un beneficio al momento de gestionar y controlar los usuarios que hacen uso del laboratorio de software. No obstante, el 7,10% restante indicó no tener una posición concreta, es decir, que no están 'ni de acuerdo, ni en desacuerdo'. Al analizar los resultados obtenidos, se determina que la mayoría de los encuestados expresó estar 'totalmente de acuerdo' o 'de acuerdo'. Esto representa un buen indicador de que el sistema representa un beneficio para la gestión y control de los usuarios que hacen uso del laboratorio, aunque se debe considerar que aún existen dudas sobre su implementación.

El ítem AU3, relacionado con la pregunta "¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software, es una buena idea en general?", reveló que el 57,10% de los encuestados están totalmente de acuerdo y, el 38,10%, están de acuerdo que el sistema es una buena idea en general. Sin embargo, el 4,80% restante indicó no tener una posición concreta, es decir, que no están 'ni de acuerdo, ni en desacuerdo'. Al analizar los resultados obtenidos, se determina que la mayoría de los encuestados expresó estar 'totalmente de acuerdo' o 'de acuerdo'. Esto representa un buen indicador de que el sistema es una buena idea en general, es decir, que es del agrado de los usuarios, aunque se debe considerar que aún existen dudas sobre la idea del sistema propuesto.

El ítem AU4, relacionado con la pregunta "¿Considera agradable el uso del sistema web que gestiona las asistencias y reportes en el laboratorio de software?", reveló que el 42,90% de los encuestados están totalmente de acuerdo y, el 50%, están de acuerdo que es agradable el uso del sistema. Sin embargo, el 4,80% restante indicó no tener una posición concreta, es decir, que no están 'ni de acuerdo, ni en desacuerdo'. Al analizar los resultados obtenidos, se determina que la mayoría de los encuestados expresó estar 'totalmente de acuerdo' o 'de acuerdo'. Esto representa un buen indicador de que es agradable hacer uso del sistema en general, aunque se debe tener en cuenta que aún existen áreas potenciales de mejora para aumentar la satisfacción del usuario.

El ítem AU5, relacionado con la pregunta "¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es una opción viable y conveniente para el registro de asistencia en el laboratorio de software en comparación con el método que se lleva actualmente?", reveló que el 54,80% de los encuestados están totalmente de acuerdo y, el 40,50%, están de acuerdo en que el sistema es una opción alternativa muy viable y conveniente en relación al método manual que se encuentra implementado actualmente. Sin embargo, el 4,80% restante indicó no tener una posición concreta, es decir, que no están 'ni de acuerdo, ni en desacuerdo'. Al analizar los resultados obtenidos, se determina que la mayoría de los encuestados expresó estar 'totalmente de acuerdo' o 'de acuerdo'. Esto representa un buen indicador de que el sistema puede solventar las deficiencias del método manual actual, haciéndolo una alternativa viable, aunque se debe tener en consideración que aún existen áreas potenciales de mejora para aumentar el porcentaje de aceptación de este método automatizado por parte de los usuarios.

El ítem TAU, correspondiente al total de la variable de Actitud Hacia el Uso, denota que el 49,06% de los encuestados están totalmente de acuerdo y, el 41,92%, están de acuerdo con

la facilidad de uso del sistema en general, lo que sugiere una buena percepción por parte de los encuestados en relación a la facilidad de uso del sistema. Por otro lado, el 8% restante indicó no tener una posición concreta, es decir, que no están ‘ni de acuerdo, ni en desacuerdo’. Y solamente el 1% señaló que está en desacuerdo. Al analizar los resultados obtenidos para este constructo, se determina que la mayoría de los encuestados expresó estar ‘totalmente de acuerdo’ o ‘de acuerdo’. Esto representa un excelente indicador de la actitud hacia el uso percibida, dando a entender una clara tendencia para adoptar y utilizar el sistema propuesto. Sin embargo, aún existen varios aspectos que se pueden mejorar para poder conseguir una mayor predisposición de los usuarios para que hagan uso del sistema.

6.2.3.4 Análisis de resultados para el constructo de Intención de Uso (IU)

La **Figura 32** presenta los resultados obtenidos para el constructo de Intención de uso (IU).

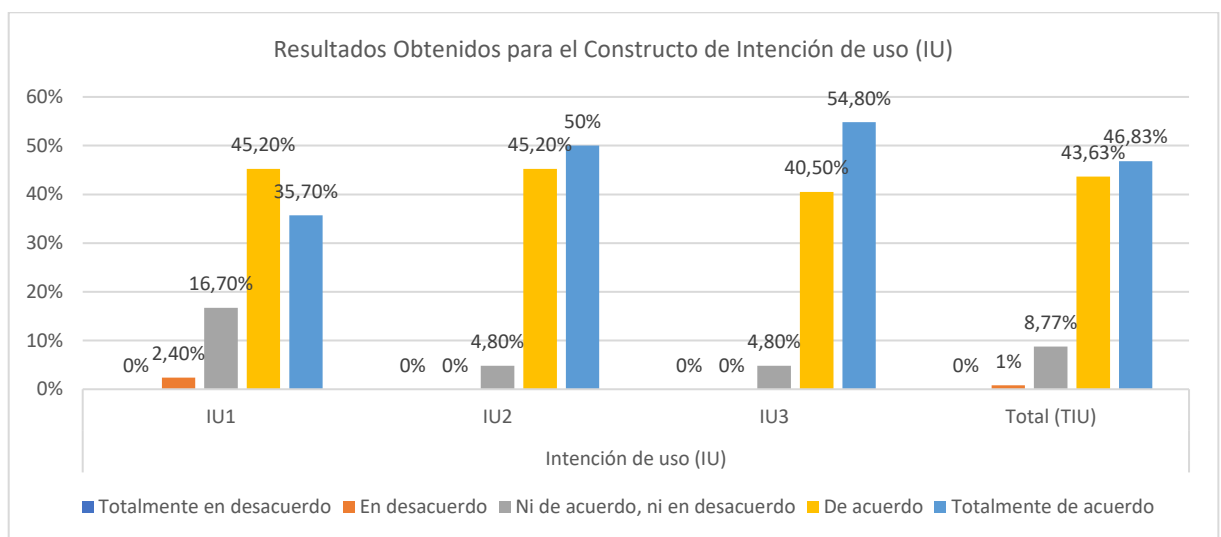


Figura 32. Resultados obtenidos por cada ítem del constructo de Intención de uso (IU) de acuerdo a la escala de Likert

El ítem IU1, relacionado con la pregunta ‘¿Tiene usted la intención de utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida dentro del laboratorio de software?’, reveló que el 35,70% de los participantes están totalmente de acuerdo con la intención de utilizar el sistema. Además, un 45,20% están de acuerdo con esta afirmación. Un 16,70% no expresó una posición clara al respecto, mientras que un 2,40% se mostró en desacuerdo. Estos resultados sugieren una disposición significativa por parte de los usuarios para adoptar el sistema propuesto, aunque también indican la presencia de ciertas dudas o reservas que podrían requerir atención para promover una mayor aceptación y uso del sistema.

El ítem IU2, correspondiente a la pregunta "¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es una opción beneficiosa en el futuro, considerándola como una alternativa viable en el registro de asistencias dentro del laboratorio de software?", mostró que el 50% de los encuestados están totalmente de acuerdo en que el sistema representa una opción beneficiosa en el futuro. Además, el 45,20% están de acuerdo con la afirmación. Por otro lado, el 4,80% indicó no tener una posición clara al respecto. Estos resultados sugieren una percepción mayoritariamente positiva de que el sistema es una opción beneficiosa en el futuro.

El ítem IU3, relacionado con la pregunta "¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida mejorará la experiencia de mantener el control de las asistencias en el laboratorio de software?", mostró que el 54,80% de los participantes están totalmente de acuerdo en que el sistema mejorará la experiencia de mantener el control de las asistencias en el laboratorio. Además, el 40,50% están de acuerdo con la afirmación. Por otro lado, el 4,80% indicó no tener una posición clara al respecto. Estos resultados sugieren una expectativa positiva hacia los beneficios que proporcionará el sistema en términos de experiencia de usuario.

El ítem TIU, que representa el total de la variable de Intención de Uso, revela que el 46,83% de los participantes están totalmente de acuerdo con su intención de uso del sistema propuesto en todas sus facetas. Además, un 43,63% están de acuerdo con esta afirmación. Un 8,77% no expresó una posición clara al respecto, mientras que un 1% se mostró en desacuerdo. Estos resultados indican una disposición generalizada por parte de los usuarios para utilizar el sistema en el futuro, lo que sugiere una aceptación positiva y una posible adopción del sistema en el contexto del laboratorio de software.

7 Discusión

7.1 Objetivo Específico 1. Desarrollar un software que permita integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso para el registro de asistencias en los laboratorios de software en Python, mediante la metodología XP

Este objetivo se basó en el desarrollo de un sistema que integre el reconocimiento facial con el sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la carrera de Computación de la Universidad Nacional de Loja, mediante la metodología de desarrollo XP. A continuación, se presentan las experiencias vividas y la comprensión de los resultados obtenidos.

Es necesario mencionar que, durante todo el desarrollo del presente objetivo, se tuvo varios aciertos, tanto con las tecnologías seleccionadas, como con el sistema desarrollado. No obstante, también se presentaron varias limitaciones en cada una de las fases que compete a la metodología XP que retrasaron el desarrollo del software.

Dentro de los aciertos, se destaca que la creación de un sistema que integre el reconocimiento facial como método biométrico para el registro de asistencias en los laboratorios de software fue una decisión idónea. Esto, ya que el uso del reconocimiento facial, representa una mejora considerable en la seguridad para el registro de asistencias, omitiendo varios fallos que se pudiesen presentar en otros métodos más tradicionales, como lo pueden ser el manejo manual de asistencias dentro de del laboratorio. Por tanto, el conocimiento técnico que se necesita para manejar el sistema, es demasiado bajo, es decir, la facilidad de uso del sistema es uno de los mayores aciertos que se pudo conseguir.

El uso de la metodología de desarrollo de software XP, también fue un gran acierto. Esto, debido a que, al ser una metodología ágil, se pudo darle un enfoque iterativo e incremental para el desarrollo del software, permitiendo construir el sistema mediante entregables pequeños, los cuales fueron siendo validados e integrados poco a poco, hasta llegar a la implementación del sistema en su totalidad.

Dentro de las limitaciones, la fase de planificación, donde se tuvo la dificultad en la obtención y elicitación de los requisitos del sistema, no obstante, con el uso de técnicas de recopilación de información, como entrevistas, observación participante y revisión de documentos, se logró abstraer los requisitos del sistema.

La fase de diseño no presentó grandes dificultades, de hecho, la mayor dificultad fue conseguir integrar las nuevas funcionalidades al sistema existente en el diagrama de clases, así como el correcto diseño del diagrama de despliegue y de componentes, con la correcta interpretación de los requisitos, se consiguió diagramar de tal manera que se pudo implementar

dichos diagramas, tanto el de clases y el de componentes en el código, como el de despliegue en el lanzamiento.

La fase de codificación tuvo muchas dificultades debido a la evidente complejidad del desarrollo del sistema, desde la refactorización de los dos sistemas actuales (Control de Acceso y Reconocimiento Facial), hasta la integración del reconocimiento facial al sistema y que funcione correctamente. Con respecto al primer punto, se tuvo que refactorizar mucho más de lo previsto en un inicio, esto, ya que Odoos recomienda emplear el patrón Modelo-Vista-Controlador (MVC), y el sistema “Control de Acceso” no lo implementaba, por lo que refactorizarlo, de tal manera que lo pueda implementar tomó su respectivo tiempo y esfuerzo. Sin embargo, el verdadero reto se presentó con el segundo punto, puesto que el sistema de “Reconocimiento Facial” presentaba muchas falencias en su código, además de que dicho reconocimiento facial lo hacía de manera local (abría la cámara del servidor), y el cliente solicitaba que se abriera la cámara web de cada cliente (en vez de que se abra la cámara del servidor, se abra la cámara de cada usuario que hace uso del sistema en la página web de la carrera de Computación). Esto conllevó a una inversión de tiempo y esfuerzo muy grande, mucho más de lo previsto (no se pudo identificar este inconveniente en la fase de planificación), puesto que migrar de una cámara local a una cámara web representa un conocimiento técnico considerable, ya que implica combinar JavaScript con Odoos, manejo de eventos, API, y demás. Sin embargo, gracias a los conocimientos técnicos, al uso de herramientas de IA y a la documentación oficial que se podía encontrar en internet, se pudo completar el sistema en su totalidad, consiguiendo así una versión estable en fase beta. Cabe mencionar que el sistema se construyó utilizando las prácticas de programación recomendadas por la documentación oficial de Odoos, como la implementación correcta del MVC, segmentación del código, manejo de API, uso de endpoints, retornos, etc.

La fase de pruebas presentó una dificultad evidente, y es la cobertura de las pruebas unitarias. Esto, principalmente se debió al tiempo muy limitado que se tuvo para el desarrollo del presente Trabajo de Integración Curricular que, al juntarlo con la complejidad de las pruebas unitarias del módulo, se escapaban del alcance de dicho trabajo. Además, dentro del proceso del reconocimiento facial, sólo se validó lo necesario antes de dicho reconocimiento (validar horario, validar UUID, validar dirección IP) y más no del proceso de reconocimiento facial propiamente dicho. Esto, ya que una prueba unitaria no brindaba la posibilidad de simular una interacción real del usuario realizando el reconocimiento facial, donde se analizaba más que una imagen o frame, un video que lo componen muchas imágenes o frames. Por tanto, es necesario hacer hincapié en que, con más tiempo, se hubiera podido ampliar la cobertura de las

pruebas unitarias, tal vez no al 100% (por la extensión del sistema y de sus respectivos componentes), pero sí extender en gran medida dichas pruebas.

La fase de lanzamiento tuvo muchos inconvenientes que retrasaron notablemente la entrega del presente Trabajo de Integración Curricular. Esto, debido a varios factores, en donde se destacan la falta de disponibilidad de los servidores de la carrera de Computación, actualización de tecnologías del entorno virtual sin previo aviso, deficiencias en las configuraciones del servidor de Odoo, actualización sin previo aviso de la versión de Odoo V15 a Odoo V17. Todos estos imprevistos, conllevaron a problemas de incompatibilidad entre el código desarrollado con la versión de Odoo (se desarrolló el sistema en Odoo V15 y se tuvo que desplegar en Odoo V17), retraso para poder hacer uso de los servidores de la carrera (muchas veces no estaban disponibles o en mantenimiento, lo que retrasaba el despliegue del mismo), cambio de versiones entre el código desarrollado y las versiones de despliegue. Esto último presenta un grave problema, ya que al actualizar tecnologías sin previo aviso involucra que las versiones anteriores queden obsoletas y se tenga que migrar, como lo puede ser un cambio de la versión de Python de 3.11.0 a 3.12.0, y, esto no compete sólo a la versión de dicha tecnología, sino al resto de tecnologías y sus versiones que dependen de ella (por ejemplo, la versión de TensorFlow depende de la versión de Python, es decir, que una versión específica de TensorFlow no puede ejecutarse en versiones anteriores de Python, etc.), que, al actualizarlas, presentaron muchos problemas de compatibilidad que se tuvieron que resolver, alargando notablemente el tiempo de entrega del presente trabajo. No obstante, con el tiempo y el conocimiento necesario, se pudo superar dichas dificultades y conseguir desplegar el sistema funcionando correctamente en los servidores de la carrera de Computación, dando por culminado el primer objetivo y todo lo que corresponde a su alcance.

7.2 Objetivo Específico 2. Determinar el nivel de aceptación tecnológica del software para el registro de asistencia en los laboratorios de software mediante el modelo TAM

Este objetivo se basó en evaluar la aceptación del sistema por parte de los usuarios, mediante el Modelo de Aceptación Tecnológica (TAM). A continuación, se presentan las experiencias vividas y la comprensión de los resultados obtenidos.

Es necesario mencionar que, durante todo el desarrollo del presente objetivo, se tuvieron muchos aciertos, como lo fue la elección de la tecnología para la evaluación de la aceptación del sistema, el modelo TAM. No obstante, también se presentaron varias limitaciones, sobre todo, en la fase de medición del modelo, donde se presentaron varias dificultades.

Dentro de los aciertos, se comprueba que la elección del Modelo de Aceptación de Tecnología (TAM) para evaluar la aceptación del sistema web de gestión de asistencias mediante reconocimiento facial en los laboratorios de software en la carrera de Computación fue acertada y proporcionó un marco cualitativo bastante sólido para dicha evaluación. El uso de TAM para la evaluación tecnológica, posibilitó tener una guía muy bien estructurada y definida que permite comprender la percepción de los usuarios con respecto al uso del sistema para el registro de asistencias mediante reconocimiento facial en los laboratorios de software. Además, se hizo uso de las cuatro variables que plantea TAM (Utilidad percibida, Facilidad de uso percibida, Actitud hacia el uso, Intención de uso) para una correcta evaluación tecnológica, evaluando diferentes aspectos, desde varios puntos de vista del sistema. Esto permitió identificar muchas áreas de mejora, y, sobre todo, permitió comprender la aceptación del sistema por parte de los usuarios de manera cuantitativa mediante la escala de Likert.

Dentro de las limitaciones, la fase de elaboración no presentó muchas dificultades, es más, la mayor dificultad presentada fue la complejidad para encontrar información oficial que sirva de guía al momento de realizar las preguntas de manera clara y concisa. Sin embargo, gracias a la investigación realizada, se pudo tomar como referencia a muchos otros trabajos de titulación y experimentos que hacían uso del modelo TAM, permitiendo elaborar preguntas acordes a cada variable según las necesidades a evaluar del sistema.

La fase de medición presentó varias dificultades y limitaciones. La más clara y evidente fue el tiempo necesario para poder realizar la demostración en tiempo real del sistema, para luego poder realizar las encuestas a los usuarios. Además, efectuar dichas demostraciones y luego las encuestas de aceptación tecnológica, también representaron una gran limitante. Esto, ya que conseguir permisos para poder realizar las demostraciones y encuestas en horarios curriculares, resultó bastante más complejo de lo previsto. No obstante, invirtiendo el tiempo y esfuerzo necesarios, se consiguió realizar las demostraciones a los tres cursos seleccionados y aplicarles las encuestas para poder cuantificar su percepción del sistema, permitiendo así tener resultados reales de los usuarios que en un futuro cercano van a hacer uso del software.

La fase de análisis representó un gran reto debido a la cantidad de información recadaba en la fase de medición, y, al no tener sólo que presentarla, sino, analizarla, conllevó bastante más tiempo del estimado conseguir un análisis profundo y convincente que tenga sustento para poder argumentar que realmente el usuario posee una percepción positiva al hacer uso del sistema. Pese a ello, dedicando aún más tiempo de lo planeado, y con la ayuda de la IA, se logró interpretar de la mejor manera los resultados obtenidos con las encuestas realizadas, generando

un análisis satisfactorio que pueda dar respuesta a la pregunta de investigación planteada en un inicio.

Todos los resultados obtenidos de las variables del Modelo de Aceptación Tecnológica (TAM), reflejan niveles medios y altos de aceptación del sistema. Lo que permite inferir que, la percepción de los usuarios respecto al software es realmente útil, fácil de usar, con una tendencia positiva para su uso y con una intención de seguir usando el sistema en un futuro. Por tanto, para dar respuesta a la pregunta de investigación planteada, la cual es: “**¿Cuál es el nivel de aceptación tecnológica que se podría obtener al integrar el sistema de Reconocimiento Facial con el sistema de Control de Acceso en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja, mediante un software que permita el registro de asistencia en los mismos?**”, basándose en los resultados obtenidos, se puede afirmar y argumentar que la aceptación tecnológica del sistema que integra el reconocimiento facial como mecanismo para el registro de asistencia en los laboratorios de software en la carrera de Computación en el sistema de Control de Acceso, se encuentra entre **media y alta** según lo planteado por la escala de Likert. Esto permite argumentar que el desarrollo del presente sistema es una opción muy viable que mejora considerablemente la seguridad de procesos automatizados y manuales al hacer uso de un dato biométrico como lo es el reconocimiento facial. Además, se puede evidenciar notablemente la gran aceptación del sistema por parte de los usuarios, lo que sugiere que su implementación traerá consigo avances considerables en la implementación de nuevas tecnologías para el manejo de procesos internos en la carrera de Computación.

8 Conclusiones

- El desarrollo del sistema que integra el reconocimiento facial como método biométrico para el registro de asistencias en los laboratorios de software de la carrera de Computación de la Universidad Nacional de Loja fue un total éxito. Este enfoque mejora significativamente la seguridad y confiabilidad en el registro de asistencias al eliminar los posibles errores asociados a los métodos manuales. Además, la facilidad de uso del sistema resultó ser un factor clave para su adopción, ya que el manejo técnico requerido es mínimo, lo que facilita su utilización por parte de los usuarios sin necesidad de capacitación especializada.
- La elección de la metodología XP para el desarrollo del sistema fue un éxito, ya que su enfoque ágil permitió la creación del sistema de manera iterativa e incremental. Este enfoque facilitó la implementación y validación de entregables pequeños y constantes, lo que aseguró la calidad y funcionalidad del sistema en cada etapa del proceso. La flexibilidad de XP permitió adaptaciones rápidas y efectivas, lo que contribuyó significativamente a la finalización exitosa del proyecto.
- El uso del Modelo de Aceptación Tecnológica (TAM) para evaluar la aceptación del sistema de control de asistencia mediante reconocimiento facial en los laboratorios de software fue un éxito. TAM proporcionó una estructura clara y eficaz para medir la percepción de los usuarios respecto a cuatro dimensiones críticas: utilidad percibida, facilidad de uso percibida, actitud hacia el uso e intención de uso. Este marco no solo ayudó a identificar áreas potenciales de mejora, sino que también permitió una comprensión profunda y cuantitativa de la aceptación del sistema, facilitando la recolección de datos objetivos mediante la escala de Likert. La elección de TAM, por tanto, ofreció una visión integral que fundamenta el éxito del sistema desde la perspectiva de sus usuarios.
- Con base en los resultados obtenidos a través del Modelo de Aceptación Tecnológica (TAM), se puede concluir que el nivel de aceptación tecnológica del sistema de reconocimiento facial integrado al control de acceso en los laboratorios de software se sitúa entre medio y alto. La percepción de los usuarios respecto a la utilidad y facilidad de uso del sistema, junto con su actitud positiva e intención de seguir utilizando la tecnología, confirman que la solución es una opción viable y eficiente. Este sistema no solo mejora significativamente

la seguridad de los procesos automatizados de registro de asistencia, sino que también demuestra una notable aceptación por parte de los usuarios, lo que refuerza su valor como una herramienta innovadora para los procesos internos de la carrera de Computación.

- El sistema desarrollado no solo constituye una solución tecnológica innovadora y eficiente para el control de asistencias, sino que además cuenta con un sólido respaldo por parte de los usuarios, lo cual afianza su potencial de integración y su valor como una herramienta fundamental en la carrera de Computación.

9 Recomendaciones

- Se recomienda aumentar los mecanismos para el registro de asistencias dentro de los laboratorios, ampliando el uso de datos biométricos, como lo pueden ser: reconocimiento de huella dactilar, de iris, de voz, entre otras. Aplicar más mecanismos de autenticación mediante datos biométricos, que más allá de mejorar la seguridad de dichos registros, también ofrece al usuario más posibilidades, y no se limita solamente al uso del reconocimiento facial.
- Se recomienda darle un enfoque iterativo e incremental a la metodología de desarrollo XP, abarcando todas sus fases. De esta manera, se aprovecha al máximo lo que puede llegar a ofrecer una metodología ágil, garantizando así un desarrollo exitoso del software. Además, seguir los lineamientos que establece esta metodología, permite realizar una documentación amplia y detallada, donde en cada fase se documenta lo realizado, generando así una guía meticulosa que presenta todo lo realizado para conseguir el software final.
- El despliegue del sistema en los servidores de la carrera de Computación presentó demasiadas dificultades y limitaciones. Por tanto, para evitar estos retrasos notables en los tiempos de entrega, se recomienda dockerizar el sistema o emplear alguna solución que evite tener problemas de disponibilidad y de versiones dentro de los servidores de la carrera. Además, si se pudiese sembrar las bases para futuras actualizaciones e instalaciones mediante una guía bien estructurada y detallada incrementaría notablemente el número de Trabajos de Integración Curricular (TIC) enfocados en ser desplegados en los servidores de la carrera de Computación, permitiendo así grandes avances tecnológicos para la misma.
- Se recomienda ampliar la población utilizada en el Modelo de Aceptación Tecnológica (TAM). Esto brindaría datos mucho más apegados a la realidad, ya que, al incrementar el número de usuarios encuestados, se expande el número de puntos de vista disponibles. De hecho, esta diversidad otorgaría una validez y fiabilidad en los datos obtenidos mucho mayor, permitiendo llegar a conclusiones con un argumento mayor.

9.1 Trabajos futuros

- Se recomienda crear un modelo de detección de vida que reemplace el modelo actual, que, al no haberlo entrenado, posee muchas limitaciones y restricciones para poderlo manejar de manera idónea y a total libertad del desarrollador. Esto

ayudaría significativamente a la mantenibilidad del sistema, asegurando una base sólida y configurable para nuevas mejoras o funcionalidades que se desee agregar.

- Se recomienda migrar a una solución basada en la nube, es decir, que todo el proceso de reconocimiento facial no lo realice el servidor, sino, la nube. Esto garantizará emplear solamente los recursos computacionales necesarios para el reconocimiento facial y la detección de vida, despreocupándose por el uso de una tarjeta gráfica (GPU) dentro del servidor, reduciendo notablemente los costos computacionales del mismo para mantener funcional el software.

10 Bibliografía

- [1]. I. A. Campoverde y L. Castillo, “Desarrollo del Prototipo de Control de acceso a los laboratorios de la Carrera de Computación de la UNL: SmartLab.”, *Unl.edu.ec*, 2023, doi: <https://dspace.unl.edu.ec/jspui/handle/123456789/27713>
- [2]. D. Alvarado, “Prototipo de Reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja.”, *Unl.edu.ec*, 2023, doi: <https://dspace.unl.edu.ec/jspui/handle/123456789/27009>
- [3]. O. G. Huamaní and D. A. Esquivel, “El proceso enseñanza–aprendizaje–evaluación (PEAE) una didáctica universitaria,” *Horizonte de la Ciencia*, vol.11, no. 20, pp. 243-254, 2021.
- [4]. G. S. Coutinho and L. F. Morán-Mirabal, “Los laboratorios virtuales y su contribución al futuro de la educación,” *Tecnológico de Monterrey*, 2022.
- [5]. R. L. Glass, *Software Conflict 2.0: The Art and Science of Software Engineering*. New York, NY, USA: Developer Press, 2006. [En línea]. Disponible en: <https://www.amazon.com/Software-Conflict-2-0-Engineering-Management/dp/0201634182>
- [6]. M. Fowler, "Patterns of Enterprise Application Architecture," Addison-Wesley, 2003. [En línea]. Disponible en: <https://martinfowler.com/books/ea.html>
- [7]. J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Boston, MA, USA: Addison-Wesley, 2010. [En línea]. Disponible en: <https://www.amazon.com/Continuous-Delivery-Deployment-Automation-Addison-Wesley/dp/0321601912>
- [8]. L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2013. [En línea]. Disponible en: <https://www.informit.com/store/software-architecture-in-practice-9780321815736>
- [9]. P. Kruchten, "The Rational Unified Process: An Introduction," 3rd ed., Addison-Wesley, 2003. [En línea]. Disponible en: <https://www.amazon.com/Rational-Unified-Process-Introduction-Edition/dp/0321197704>
- [10]. M. López, J. Gómez, and A. Molina, “Integración de sistemas de información: una revisión de la literatura,” *Revista de investigación en sistemas e informática*, vol. 3, no. 2, pp. 7–18, 2006.

- [11]. Sommerville, Ingeniería de software, 9.^a ed. Madrid: Pearson Educación, 2011.
- [12]. K. Beck, Extreme Programming Explained: Embrace Change, 1st ed. Boston, MA, USA: Addison-Wesley Professional, 1999. [En línea]. Disponible en: <https://archive.org/details/extremeprogrammi0000beck>
- [13]. K. Beck and C. Andres, Extreme Programming Explained: Embrace Change, 2nd ed. Boston, MA, USA: Addison-Wesley Professional, 2004. [En línea]. Disponible en: https://archive.org/details/extremeprogrammi0000beck_u5f0
- [14]. M. Fowler, Refactoring: Improving the Design of Existing Code. Boston, MA, USA: Addison-Wesley Professional, 1999. [En línea]. Disponible en: <https://refactoring.com/>
- [15]. J. Shore and S. Warden, The Art of Agile Development. Sebastopol, CA, USA: O'Reilly Media, 2007. [En línea]. Disponible en: <https://www.oreilly.com/library/view/the-art-of/9780596527679/>
- [16]. J. Newkirk and R. Martin, Extreme Programming in Practice. Boston, MA, USA: Addison-Wesley, 2001. [En línea]. Disponible en: <https://archive.org/details/extremeprogrammi0000new>
- [17]. R. Pressman, Ingeniería de Software: Un enfoque práctico. McGraw Hill New York, 2010.
- [18]. F. D. Davis, "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," MIS Quarterly, vol. 13, no. 3, pp. 319–340, Sep. 1989. [En línea]. Disponible en: <https://www.jstor.org/stable/249008>
- [19]. V. Venkatesh and F. D. Davis, "A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies," Management Science, vol. 46, no. 2, pp. 186–204, 2000. [En línea]. Disponible en: <https://pubsonline.informs.org/doi/10.1287/mnsc.46.2.186.11926>
- [20]. I. Ajzen and M. Fishbein, Understanding Attitudes and Predicting Social Behavior. Englewood Cliffs, NJ, USA: Prentice-Hall, 1980. [En línea]. Disponible en: <https://psycnet.apa.org/record/1981-05747-000>
- [21]. V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User Acceptance of Information Technology: Toward a Unified View," MIS Quarterly, vol. 27, no. 3, pp. 425–478, 2003. [En línea]. Disponible en: <https://www.jstor.org/stable/30036540>
- [22]. P. Legris, J. Ingham, and P. Colletette, "Why Do People Use Information Technology? A Critical Review of the Technology Acceptance Model,"

- Information & Management, vol. 40, no. 3, pp. 191–204, 2003. [En línea]. Disponible en: [https://doi.org/10.1016/S0378-7206\(01\)00143-4](https://doi.org/10.1016/S0378-7206(01)00143-4)
- [23]. Y. Liu, L. Li, and X. Carl, "An Analysis of Technology Acceptance Model in E-Commerce," *Journal of Technology and Management*, vol. 15, no. 5, pp. 45–54, 2020. [En línea]. Disponible en: <https://doi.org/10.1016/j.techfore.2019.11.016>
- [24]. I. Rodríguez del Bosque, M. Garcia de los Salmones, and fi. Herrero, "La influencia de la propensión a innovar del gestor en la implantación de programas de fidelización de clientes en el pequeño comercio," *Cuadernos de Economía y Dirección de la Empresa*, vol. 12, 03 2009.
- [25]. G. van Rossum, "Python Tutorial," Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands, 1995. [En línea]. Disponible en: <https://docs.python.org/3/tutorial/>
- [26]. M. Lutz, *Learning Python*, 5th ed. Sebastopol, CA, USA: O'Reilly Media, 2013. [En línea]. Disponible en: <https://www.oreilly.com/library/view/learning-python-5th/9781449355722/>
- [27]. P. Barry, *Head First Python*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2016. [En línea]. Disponible en: <https://www.oreilly.com/library/view/head-first-python/9781491919521/>
- [28]. A. Grinberg, *Flask Web Development: Developing Web Applications with Python*. Sebastopol, CA, USA: O'Reilly Media, 2018. [En línea]. Disponible en: <https://www.oreilly.com/library/view/flask-web-development/9781491991725/>
- [29]. T. Oliphant, *Guide to NumPy*, 2nd ed. USA: Continuum Press, 2015. [En línea]. Disponible en: <https://www.numpy.org>
- [30]. J. VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data*. Sebastopol, CA, USA: O'Reilly Media, 2016. [En línea]. Disponible en: <https://jakevdp.github.io/PythonDataScienceHandbook/>
- [31]. F. Pinckaers, "Odoo: The Open Source Business Model," Odoo S.A., 2005. [En línea]. Disponible en: <https://www.odoo.com/page/about-us>
- [32]. Odoo Documentation Team, "Odoo Functional Documentation," Odoo S.A., 2024. [En línea]. Disponible en: <https://www.odoo.com/documentation/>
- [33]. A. Khaefi, *Odoo Development Essentials: A fast-paced guide for developers to leverage Odoo's powerful framework*. Birmingham, UK: Packt Publishing, 2015. [En línea]. Disponible en: <https://www.packtpub.com/product/odoo-development-essentials/9781785883645>

- [34]. G. Antunes, M. Mira da Silva, and J. Bernardino, "Open Source ERP Adoption: A Case Study of Odoo," in *Journal of Enterprise Information Management*, vol. 31, no. 2, pp. 229-245, 2018. [En línea]. Disponible en: <https://doi.org/10.1108/JEIM-10-2016-0163>
- [35]. M. Stonebraker and L. A. Rowe, "The Design of POSTGRES," *Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data*, pp. 340–355, 1986. [En línea]. Disponible en: <https://dl.acm.org/doi/10.1145/16894.16888>
- [36]. PostgreSQL Global Development Group, "PostgreSQL Documentation," 2024. [En línea]. Disponible en: <https://www.postgresql.org/docs/>
- [37]. D. Choi and A. Jung, *Mastering PostgreSQL 15*. Birmingham, UK: Packt Publishing, 2023. [En línea]. Disponible en: <https://www.packtpub.com/product/mastering-postgresql-15/9781804619030>
- [38]. R. Furtado and H. Lopes, "PostgreSQL: The Emerging Leader in Open Source Databases," in *Journal of Database Management*, vol. 34, no. 2, pp. 45–60, 2022. [En línea]. Disponible en: <https://doi.org/10.4018/JDM.2022.341192>
- [39]. T. Ortiz and V. Waldemar, "Sistema de reconocimiento facial automatizado para el control de asistencia de recursos humanos," Master's thesis, Universidad Nacional de Loja, Loja, Ecuador, 2014.
- [40]. D. Mendoza and J. Bastidas, "Registro de asistencia de alumnos por medio de reconocimiento facial utilizando visión artificial," Master's thesis, Universidad Técnica de Ambato, Ambato, Ecuador, 2019.
- [41]. C. Dorado and M. Ayala, "Sistema biometrico de reconocimiento facial para el control de asistencia del personal docente y administrativo de la uniandes tulcán," Master's thesis, Universidad Regional Autónoma de los Andes, Ambato, Ecuador, 2018.
- [42]. L. Albarracín and S. Lara, "Prototipo de reconocimiento facial para mejorar el control de asistencia de estudiantes en el laboratorio de cómputo, uniandes-quevedo," Master's thesis, Universidad Regional Autónoma de los Andes, Ambato, Ecuador, 2020.
- [43]. T. C. Lethbridge, J. Singer y A. Forward, "Cómo los ingenieros de software utilizan la documentación: el estado de la práctica," *IEEE Software*, vol. 20, no. 6, pp. 35-39, nov.-dic. 2003, doi: 10.1109/MS.2003.1241364.

11 Anexos

ANEXO 1: Entrevista realizada al encargado de los laboratorios de software de la carrera de computación, Ing. Luis Sinche, para la elicitación de requisitos



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- **Observador Participante:**
 - Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).
- **Operador:**
 - Ing. Luis Darío Sinche Cueva, Mg.Sc., encargado del laboratorio de software de la carrera de Computación de la Universidad Nacional de Loja.
- **Fecha:**
 - 27 de octubre del 2023

Descripción: El presente anexo detalla la observación participante realizada al encargado de los laboratorios de Software de la Universidad Nacional de Loja, Ing. Luis Sinche, para analizar las funcionalidades actuales del Sistema de Control de Acceso y su integración con Reconocimiento Facial para el registro de asistencias en dichos laboratorios.

TRANSCRIPCIÓN DE LA OBSERVACIÓN PARTICIPANTE

Observador Participante: ¿Qué funcionalidades considera esenciales o mínimas para el software de integración de lo que es el acceso laboratorio con el reconocimiento facial para el registro de asistencias?

Operador: Yo lo que manejo es este, el ingreso de las tarjetas.

Observador Participante: Claro, sería más porque quiero ver cómo esta segmentado, porque yo no tengo acceso a esos sistemas, entonces no puedo testear cómo están las funcionalidades.

Operador: Entonces verás, aquí me sale para producir los reportes, y tengo lo de asistencias. Me salen todas las asistencias que han existido, desde los administrativos, estudiantes y docentes. Ya está todo, donde dice “no”, es que no pudo acceder.

Observador Participante: Es lo de los intentos fallidos.

Operador: Por ejemplo, la compañera, su número de tarjeta, el responsable, o sea, la materia donde no pudo ingresar, una hora de ingreso, la temperatura de 39.2, que es muy alta, por eso no se le dejó el ingreso. Ya, entonces tenemos eso, filtros, podemos añadir un filtro personalizado creado “por materia, contiene Análisis”, aplicar. Ya, entonces me da un filtro de, por ejemplo, de los muchachos que han entrado el 26 de abril, esto es del ciclo pasado, mira, me da desde la primera vez que ingresaron, voy hacia abajo, hasta el 10 de mayo. Ya, eso.

Observador Participante: Todo eso es dentro del módulo de asistencia que está en esa parte.

Operador: Toda la parte que es generar reportes y asistencias. Eso es lo del intento de ingreso, tengo los intentos de ingreso.

Observador Participante: Digamos, tema de asistencias, ¿no pueden gestionar verdad? O sea, ¿justificar o cosas así?

Operador: No, este sistema no está para esto. O sea, lo que hace el sistema básicamente es leer la tarjeta del estudiante y permitirle el acceso si tiene la temperatura adecuada. Y si tiene, lo registra aquí, lo va registrando, si es docente o estudiante, eso es lo que va registrando, nada más. Pero, por ejemplo, eso sería bueno que: dividir las asistencias por horas. Por ejemplo: de siete y media die y media. No sé de porque mira, aquí está en conjunto todo.

Observador Participante: Claro, porque yo esa duda tengo, de cómo hacer la interfaz gráfica. Por lo que el tema de los recursos de backend, la mayoría ya se están construidos. Entonces, queríamos consumir un recurso y modificarlo como un recurso propio mío. Para lo que yo necesito en la parte visual, eso quería comprobar cómo estaba el sistema como tal.

Operador: Y mira que no está bien, porque estamos desde abril y mayo. No me da todo. No me da lo que se ha trabajado ahora. Así le falta a esto de aquí, no me está dando todo lo que necesito. Usuarios, estudiantes, estos son todos los estudiantes que tenemos.

Observador Participante: Los cursos siguen siendo los mismos, ¿verdad?

Operador: Sí, pero igual hay que modificarlos. Tenemos que actualizarlos. De administradores, está: Pablo, Eduardo Lemberg, María Armijos, ella es la conserje, estoy yo, y Pablo que siempre entramos aquí. La tarjeta de la conserje, es la única que no le pide la temperatura, a mí sí me pide la temperatura. Eso... pasamos con el conjunto de administradores, un conjunto de docentes, un conjunto de estudiantes, y sus reportes de las asistencias que tenemos así. Sí quisiera, por ejemplo, poner filtros, “añadir filtro personalizado” ... bueno, esta parte de aquí, sí, mejorarla, no sé cómo podríamos mejorarla, aquí tal vez, tenemos la materia que contiene, en este caso, solo tenemos unas pocas.

Pongamos “human perception”, y ya me sale esto de aquí, tengo, mira, no sé, tengo de abril, tengo del 02 mayo, junio, julio. No sé si podríamos segmentarlas por mes, para que no esté todo combinado.

Observador Participante: Ese tipo de calendario puede ser algo así.

Operador: Algo así, o separarlas por mes. No sé cómo. Tener algo que se vea mejor, no, poder dividir. Fecha de ingreso... ya, haciendo doble clic, recién lo arregla de otra forma. Estamos desde julio. Y eso... mira, son de julio, ¿cierto?

Observador Participante: Y, digamos, entonces, cuando, por ejemplo, hay algunos que pasan de ciclo y empiezan de nuevo el ciclo, usted tiene que gestionar todo eso manualmente, ¿verdad?

Operador: Sí, hay que gestionarlo todo así, a mano, hay que hacer de nuevo todo, totalmente.

Observador Participante: Eso solo lo hace el administrador, ¿cierto? El docente solamente registra asistencia y está lo de las prácticas.

Operador: Mira, otra cosa, por ejemplo, en la fecha de ingreso, me sale solo de abril, hago que me salga desde el otro, me sale, a ver, ahí estamos... de octubre. Sí, esta es la última, me salió el 23 de octubre, pero, mira, al hacerle varios clics, se hizo, al principio no estaba así. Reporte, asistencias, el 4. Mira, ahí le hice un click y sigue en el 4. Ahí en el tercer click, me vota el último, el 10.

Observador Participante: Claro, es más tema de, experiencia de usuario ya.

Operador: Sí, eso. Pero esto es el único acceso que tengo de reporte de asistencia. Claro. Entonces mira, por ejemplo, no sé... reportes, asistencias, y me sale todito. Quiero filtrarlos, por, qué se yo, por los estudiantes: nombre de usuario, temperatura, tarjeta,

responsable, materia, laboratorio, ingreso de usuario, id. Mira, no hay para estudiantes, quiero ingreso por estudiantes, no me sale; quiero docentes, no me sale. O por materia, sea el profesor con los estudiantes. No sé si tal vez por ahí tal vez puedas modificarlo. Sí, para tener, por ejemplo: quiero ver de tal profesor, entonces, muy bien, tal profesor, si ha asistido con sus estudiantes en tales fechas. No sé si me explico. Sí, sí, comprendo. Eso, aquí es como que muy general, todo está metido en un solo cajón todo.

Observador Participante: Claro, eso lo trae de la base de datos y viene tuplas digamos.

Operador: Eso, eso es lo que tenemos, no sé si quieres ver algo más.

Observador Participante: No, solo eso sería, porque, como le mencioné, con el ingeniero Pablo quedamos en que... o sea, hacerle como un segundo registro acá adentro. Aunque decía que algunos llevaban manual, algunos que se olvidan la tarjeta y todo eso. Y acá dentro tienen que tomar la asistencia y todo eso. Entonces, yo decía centrarme solamente en este apartado. Y, digamos, esto quedaría como registro de ingreso digamos solamente de la puerta, pero ya propiamente lo que es manejo de asistencia de los estudiantes que están aquí recibiendo clases. Tenía planeado pues crear otro módulo o simplemente crear, tal vez, otra sección en el módulo de asistencias. Por eso quería ver cómo estaba construido. Porque en el manual de usuario, por ejemplo, se especifica las funcionalidades que tiene, pero, por ejemplo, no muestra toda la página principal, sino que va por secciones digamos en asistencias, y ya le pone un cuadrito de gestión, entonces no se ve todo cómo está construido.

Operador: Sí, sí, chévere que lo hagan así por materia. Por materia sería bueno. Entonces, por ejemplo, aquí está incluido también el docente, entonces, yo por ejemplo ver, tal docente asistió o no asistió en sus dieciséis semanas que, además, se hace una práctica, serían dieciséis prácticas. Como tal materia o el nombre del docente que me deban de decir los dieciséis días que él estuvo aquí. Entonces, digamos, no apareció aquí. Ah no, no ha estado, él no ha venido esa semana a cumplir sus horas prácticas.

Observador Participante: Claro. Porque, como igual estábamos viendo cómo se va a, como solución, restringir, pues, que solo se acceda y registre asistencia en estos rangos de IPs y todo eso. Entonces, con esa parte también se asegura de que la asistencia corra solo con los que esté acá adentro de la universidad.

Operador: Exacto, sería muy bueno.

Observador Participante: Entonces, eso también ya está planteada la solución. Lo que quería ver más o menos, la parte visual, cómo ponerlo, porque eso serían requisitos que tengo que evaluar para hacer el prototipado y todo eso, por ejemplo, con el ingeniero Pablo pues, como no maneja el tema de reportes de asistencias ni nada de eso, pues no, no me he ido mucho con esa parte, entonces quería ver ahí. Aquí con esto ya me queda más claro cómo construir la página.


Operador: No sé si te entendí. Por ejemplo, esto es lo que nosotros hacemos, nosotros para registrar la asistencia. Nosotros lo podemos registrar aquí internamente. Yo, digamos, estoy fuera de la Universidad, no puedo registrar mi entrada, ni mi salida. Entonces lo que tú harías es, por ejemplo, que ellos vengan aquí a las computadoras, porque estas computadoras tienen justamente una dirección, podrías consultarlo con DTI, que se active, por ejemplo, tu sistema solamente con esa dirección IP. Entonces que los chicos se sientan, prendan las computadoras y ahí colocan su asistencia, así como te muestra ahoritas. Muy bien, abro la computadora, registro mi existencia, esa sería otra opción ¿no?

Observador Participante: Sí, de ahí solo ahí sería porque, como le menciono, lo que quería yo era el apartado de asistencia mejorar esa parte, en la materia, con el tema de gestión porque ya está programado ya. Eso no más sería ingeniero.

Link del drive con el audio de la entrevista:

<https://drive.google.com/file/d/1irGCKmB2SFekaothIy7ZJmGJOtHLRWNS/view?usp=sharing>

Para constancia y validación de la información expuesta, firma a continuación el encargado del laboratorio de software de la carrera de Computación de la Universidad Nacional de Loja, en su calidad de cliente principal del proyecto:

Ing. Luis Darío Sinche Cueva, Mg.Sc.	 <p>El código QR pertenece a: LUIS DARIO SINCHE CUEVA</p>
--------------------------------------	---

ANEXO 2: Entrevista realizada al director de la carrera de computación, Ing. Pablo Ordoñez, para la elicitación de requisitos



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- **Entrevistador:**
 - Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).
- **Entrevistado:**
 - Ing. Pablo Fernando Ordoñez Ordoñez, Mg.Sc., director de la carrera de Computación y encargado principal del laboratorio de software de la Universidad Nacional de Loja.
- **Fecha:**
 - 01 de noviembre del 2023

Descripción: El presente anexo detalla la entrevista realizada al director de la carrera de Computación de la Universidad Nacional de Loja, Ing. Pablo Ordoñez, para discutir las funcionalidades esenciales del Software de Integración del Sistema de Control de Acceso y su incorporación con Reconocimiento Facial para el registro de asistencias en los laboratorios de Software de la carrera de Computación.

TRANSCRIPCIÓN DE LA ENTREVISTA

Entrevistador: ¿Qué funcionalidades considera esenciales o mínimas para el software de integración de lo que es el acceso laboratorio con el reconocimiento facial para el registro de asistencias?

Entrevistado: Bueno, hay muchas funcionalidades, pero creo que las principales son obviamente el reconocimiento facial. Sí, con gestos del usuario o, bueno, en este caso al estudiante, o el docente que podría ser uso de los laboratorios, mediante una cámara, puede ser la cámara que esté conectada al computador, y que la herramienta por la interfaz de reconocimiento sea web, sí, en como ecosistema de Odoo, eso, en la versión que más nos convenga. Entonces eso atado al o a integrado al reconocimiento de los usuarios también con tarjeta RFID que ya está en una aplicación de producción que toma la temperatura y utiliza tarjeta RFID, y pues lo que queremos hacer es unir esas dos cosas para que la principal funcionalidad sea el reconocimiento de la persona tenía el laboratorio.

Entrevistador: ¿Cuáles considera que serían las principales características que debe tener la interfaz gráfica? Por ejemplo, un calendario puede ser o que se presente en forma de listas o tablas las asistencias o ¿cómo es de su preferencia ver dicha interfaz?

Entrevistado: Bueno, ahí depende mucho de la realidad y la necesidad que se le dé a la interfaz web. Lo que tenemos que buscar siempre es la interfaz más amigable, entre más amigable, o sea, una interfaz es amigable cuando los usuarios se sienten cómodos haciendo las transacciones en la aplicación. Entonces, obviamente, si estamos en una aplicación que va a detectar el rostro de la persona y va a permitir ingresar mediante una tarjeta RFID y controlar su temperatura para que no excedan, no cierto, los grados y haya un riesgo de enfermedad de contagio como lo que pasó en la pandemia. Entonces hay que ver la mejor forma de no cansar al usuario. Entonces la interfaz, obviamente, tiene que ser amigable. Sí, tiene que ser lo más simple y lo más ágil. Sí, tampoco llena de botones o llena de colores. Sí. Que desvíen un poco la atención de los usuarios. Simplemente hay que desplegar la cámara,

reconocer, se registra. Sí, y es lo único que va a hacer el estudiante o el docente, pero el administrador, si tiene que tener muchas otras opciones, tiene que poder ver los reportes de las resistencias, esas asistencias si corresponde al horario de laboratorio que está basado en un calendario, ese calendario sea planificado previa utilización de los de los laboratorios. Sí, que cada hora designada en el laboratorio tiene un docente técnico responsable o un docente tutor que se encarga de hacer las prácticas experimentales y, que al final de cada práctica, se pueda delimitar su desarrollo de la práctica.

Entrevistador: ¿De qué manera desea que se manejen o se almacenen las imágenes de los involucrados al momento de marcar la asistencia? Es decir, dichas imágenes, ¿se guardarán en una misma base de datos o si tal vez es de su preferencia guardar en alguna base de datos independiente que se dedique solo a archivos multimedia?

Entrevistado: No, no vas a guardar imágenes. O sea, no vas a almacenar, no vas a tomar foto lo que la cámara lo que va a hacer es reconocer el rostro de la persona, cierto que ya mediante un modelo inteligente que ya está pre entrenado. Sí. Entonces, obviamente, ese modelo ya está obviamente entrenado y se basó en miles de imágenes para lograr ese reconocimiento. Sí. Bueno, me imagino que lo que hace la foto para luego para comparar, ya es que Odoos utiliza su base de datos propia, obviamente la foto va a quedar almacenada en la misma base de datos de Odoos.

Entrevistador: ¿Es necesario que los administradores, es decir, los encargados de los laboratorios, registren su asistencia cuando hagan uso de laboratorios o solamente estudiantes y docentes con materias curriculares?

Entrevistado: O sea, el registro de quién entra es obligatorio para todos. Así sea un administrador, sea un docente, sea un estudiante. La diferencia es que el administrador no está sujeto a horarios. Porque obviamente es el responsable del laboratorio, entonces él va a estar ahí todo el tiempo. Entonces, si yo a un usuario le asigno un rol de administrador, significa que él puede entrar en cualquier momento. ¿Cómo va a entrar? igual con el

reconocimiento. Sí, es lo que hacemos también con el RFID. Los administradores tienen una tarjeta RFID, pero no tienen horario, a diferencia de los otros que sí tienen el horario. Sí, designado. Por ejemplo, tengo prácticas de software, de ocho a 10, entonces solo de 8 a 10 te va a dejar entrar a ese usuario. Y los administradores, no tienen horarios, pero sí se registra el acceso también.

Entrevistador: ¿Se debe de segregar por roles o si todos pueden ver el tema de asistencia sin hacer una segregación por roles?

Entrevistado: No pues, cada rol tiene un rango de permisos. Sí, el administrador tiene de 1 a 20 permisos y el súper usuario, que es el que administra todo el sistema, cuando crea un administrador, te dice: vas a tener los 20 permisos, o solo vas a tener uno, dos, tres, cuatro, cinco, o para ti, solo puedes ver los registros, pero no puedes crear más horarios o no puedes registrar estudiantes. Sí, esas son propiedades que se heredan en cada objeto.

Entrevistador: ¿Cómo se pueden realizar las pruebas para el registro de asistencias? Porque, por ejemplo, cuando se registran las asistencias, eso también corresponde a una materia específica, a su vez, esa materia tiene más estudiantes. Entonces, como todos se relacionan, existe la duda de si ¿se maneja por separado, se tendría que tomar en cuenta el tema de horarios, materias o se puede dejar quemado solamente para realizar las pruebas?

Entrevistado: No. O sea, tú siempre tienes que utilizar datos de prueba. Nada, lo único que quemas es en la base de datos, pero sin ninguna variable histórica. Ahora déjame pensar, no, no hay nada estático, todo es dinámico. Son periodos, los periodos tienen ciclos, los ciclos tienen asignaturas, las asignaturas tienen estudiantes y docentes. Depende del modelo que has hecho.

Entrevistador: Pero, ¿todo eso se tiene que desarrollar, aunque ya esté desarrollado y en producción en el sistema de control de acceso?

Entrevistado: No, puedes tomar el mismo modelo. Puedes tomar el mismo modelo y aumentarle el reconocimiento facial. O sea, puedes manejar las mismas clases, el mismo modelo, la misma lógica que tienes en el módulo ya desarrollado. Por eso es la pertenencia, porque ya tienes una base, o sea, ¿por qué vas a volver a hacer la base?

Entrevistador: ¿Cómo se pueden consumir los datos existentes de la base de datos del sistema de control de acceso?

Entrevistado: Tú te refieres a lo que hay, no, no te preocupes. Haz cuenta que es borrón y cuenta nueva. O sea, vos estás creo preocupado por los datos que ya están ahí, y después cómo te los llevas para acá. No, haz cuenta que es borrón y cuenta nueva porque eso ya se hizo un informe y ya está archivado. Pues este ciclo estamos otra vez reiniciando.

Entrevistador: ¿Se puede reutilizar la lógica o el código fuente del sistema de control de acceso o se debe volver a crear desde cero la lógica de programación?

Entrevistado: No pues, tienes que manejar los ambientes del ciclo de vida de software. Tienes un ambiente de desarrollo que, obviamente, tú vas a tener toda la accesibilidad y ya cuando hayas pasado el ciclo de desarrollo vas producción. Sí, o sea, si quieres una réplica de lo que está ahora en producción, se puede replicar. Sí, para para ver los registros que se hizo el ciclo pasado. Por ejemplo, porque desde el ciclo pasado estamos manteniendo registros de acceder a los laboratorios. Pero, ya diría que poco y nada te va a servir, porque igual tienes que hacer pruebas tú. Entonces, la ventaja de odoo como es modular, lo único que haces es traer el módulo a instalarlo en producción.

Inclusive, lo puedes mejorar porque los chicos que hicieron eso, hicieron el modelo MVC (modelo-vista-controlador) en un solo “.py” (archivo.py). Esa no es una buena práctica de desarrollo. Tú tienes que hacer un “.py” por cada clase. O depende del modelado, que cada clase sea un objeto, tienes que aplicar las buenas prácticas. Sí, lo que ellos hicieron, les sirvió para titularse, pero no es una buena práctica de software. O sea, no puedes meter todo el MVC en un solo archivo.

Link del drive con el audio de la entrevista:

https://drive.google.com/file/d/18PNoaM8GXRVoZQ8ljVYyF32Ea9GVD_GI/view?usp=sharing

Para constancia y validación de la información expuesta, firma a continuación el director de la carrera de Computación y encargado principal del laboratorio de software de la Universidad Nacional de Loja, en su calidad de cliente principal del proyecto:

Ing. Pablo Fernando Ordoñez Ordoñez, Mg.Sc.



ANEXO 3: Historias de Usuario



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

- **Revisado y aprobado por:**

- Ing. Edison Leonardo Coronel Romero, Mg.Sc., tutor del Trabajo de Integración Curricular.

Descripción: El presente anexo detalla las historias de usuario recopiladas en la fase de planificación del presente Trabajo de Integración Curricular. Estas historias de usuario son una extensión mucho más detallada y específica de los requisitos abstraídos de los clientes directos del sistema.

HISTORIAS DE USUARIO

Historia de usuario	
Identificador: 001	Nombre: Refactorizar el sistema de control de acceso
Usuario: Administrador	
Iteración asignada: 001	Referencia de requisito: RF001
Fecha de creación: 13/11/2023	Fecha de entrega: 17/11/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Alta
Descripción:	
<p>COMO administrador</p> <p>QUIERO que la funcionalidad actual del sistema de control de acceso se mantenga inalterada después de la implementación del patrón Modelo-Vista-Controlador (MVC) y la reorganización del código (implica la separación de la lógica de negocio en modelos, vistas y controladores distintos)</p> <p>PARA mejorar la estructura interna y la mantenibilidad del sistema.</p>	
Observaciones:	
<ul style="list-style-type: none"> • La implementación del patrón MVC permitirá una separación clara de las responsabilidades dentro del código, lo que facilitará futuras actualizaciones y mejoras. • Es fundamental que la reorganización del código se realice de manera que se mantenga la coherencia y la eficiencia del sistema, sin introducir errores ni afectar negativamente a la funcionalidad existente. • La implementación del patrón MVC y la reorganización del código son cruciales para mejorar la estructura y la mantenibilidad del sistema de control de acceso. 	
Criterios de aceptación:	
<ol style="list-style-type: none"> 1. La funcionalidad actual del sistema de control de acceso debe permanecer completamente operativa después de la implementación del patrón MVC y la reorganización del código. 2. La lógica de negocio del sistema debe estar claramente separada en modelos, vistas y controladores distintos, según los principios del patrón MVC. 3. Los modelos deben ser responsables de representar los datos y la lógica de negocio de manera coherente y estructurada. 4. Las vistas deben encargarse de la presentación de la información al usuario de forma clara y organizada. 5. Los controladores deben manejar la interacción entre el usuario y el sistema, procesando las solicitudes y coordinando las acciones necesarias. 6. La reorganización del código no debe afectar negativamente el rendimiento ni la estabilidad del sistema. 7. Se realizarán pruebas para verificar que la funcionalidad existente no se vea comprometida y que los cambios internos sean transparentes para los usuarios finales. 8. Se documentarán los cambios realizados en la estructura interna del sistema para facilitar el mantenimiento y la comprensión del mismo en el futuro. 	

Historia de usuario

Identificador: 002	Nombre: Actualizar gestión de laboratorio
Usuario: Administrador	
Iteración asignada: 001	Referencia de requisito: RF002
Fecha de creación: 18/11/2023	Fecha de entrega: 20/11/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Alta

Descripción:

COMO administrador

QUIERO agregar tres campos adicionales al módulo de gestión de los laboratorios, los cuales son: campo de texto para capturar la dirección IP y la máscara de red asociadas a cada laboratorio, checkboxes para habilitar o deshabilitar tanto la dirección IP como los equipos informáticos

PARA mejorar la eficiencia en la gestión de la red y la configuración de los equipos informáticos asociados a cada laboratorio, sin afectar la experiencia de los usuarios finales.

Observaciones:

- La adición de los tres campos adicionales tiene como objetivo mejorar la gestión de la red y la configuración de los equipos informáticos asociados a cada laboratorio.
- Es fundamental garantizar que los nuevos campos no introduzcan errores ni afecten negativamente a la funcionalidad existente del sistema.

Criterios de aceptación:

1. Se agregarán tres campos adicionales en la interfaz de gestión de laboratorios para capturar la dirección IP, la máscara de red y los checkboxes de habilitación/deshabilitación de la dirección IP y los equipos informáticos.
2. Los nuevos campos deben integrarse de manera coherente en la interfaz de usuario, sin afectar la usabilidad ni la experiencia del usuario.
3. Los checkboxes de habilitación/deshabilitación deben funcionar correctamente, permitiendo al administrador controlar de manera efectiva la configuración de la dirección IP y los equipos informáticos asociados a cada laboratorio.
4. Se realizarán pruebas para verificar que la adición de los nuevos campos no introduzca errores en la gestión de laboratorios y que la funcionalidad existente permanezca intacta.
5. Se documentarán los cambios realizados en la gestión de laboratorios para facilitar el mantenimiento y la comprensión del sistema en el futuro.
6. Se documentarán los cambios realizados en la estructura interna del sistema para facilitar el mantenimiento y la comprensión del mismo en el futuro.

Historia de usuario

Identificador: 003	Nombre: Obtener UUID
Usuario: Administrador, Docente, Estudiante	
Iteración asignada: 002	Referencia de requisito: RF003
Fecha de creación: 21/11/2023	Fecha de entrega: 21/11/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Alta

Descripción:

COMO administrador, docente o estudiante del sistema de control de acceso
QUIERO generar un UUID único y aleatorio que se almacene en las cookies del dispositivo que lo está generando

PARA que este UUID pueda ser utilizado posteriormente en el registro de un equipo informático.

Observaciones:

- El UUID generado debe ser único y aleatorio para garantizar la identificación precisa y evitar conflictos con otros equipos informáticos.
- El almacenamiento del UUID en las cookies del dispositivo permitirá asociar el dispositivo con el sistema de manera sencilla y persistente.
- Es importante garantizar la seguridad del UUID almacenado en las cookies para evitar manipulaciones o accesos no autorizados.
- El sistema debe validar que las cookies están habilitadas en el dispositivo antes de intentar almacenar el UUID.
- Los usuarios deben ser notificados de que se generará y almacenará un UUID en las cookies de su dispositivo.

Criterios de aceptación:

1. El sistema debe generar un UUID único y aleatorio cada vez que un usuario solicite esta funcionalidad.
 2. El UUID generado debe almacenarse correctamente en las cookies del dispositivo utilizado.
 3. Los usuarios deben recibir una notificación o confirmación de que el UUID ha sido generado y almacenado con éxito.
 4. El UUID almacenado debe estar disponible para su uso posterior en el registro de un equipo informático.
 5. Se realizarán pruebas para garantizar que el UUID se genera de manera única y aleatoria, y que se almacena correctamente en las cookies de diversos navegadores y dispositivos.
 6. La funcionalidad debe cumplir con las políticas de privacidad y protección de datos, asegurando que el UUID almacenado en las cookies no se comparta ni sea accesible por terceros sin autorización.
 7. Se documentará el proceso de generación y almacenamiento del UUID en las cookies, detallando los pasos técnicos.
-

Historia de usuario

Identificador: 004	Nombre: Registrar equipo informático
Usuario: Administrador	
Iteración asignada: 002	Referencia de requisito: RF004
Fecha de creación: 22/11/2023	Fecha de entrega: 22/11/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Alta

Descripción:

COMO administrador

QUIERO poder crear equipos informáticos vinculados a un único laboratorio

PARA gestionar eficientemente los recursos informáticos en cada área de trabajo.

Observaciones:

- La capacidad de crear equipos informáticos vinculados a laboratorios individuales permitirá una mejor organización y asignación de recursos dentro del sistema.
 - Cada equipo informático debe contar con atributos válidos, incluyendo un UUID único, un identificador único, un estado para determinar su habilitación y una referencia al único laboratorio al que está asociado.
 - Es esencial que la creación de equipos informáticos se realice de manera intuitiva y eficiente para el administrador del sistema.
-

Criterios de aceptación:

8. El sistema debe proporcionar una interfaz clara y accesible para que el administrador pueda crear equipos informáticos vinculados a laboratorios.
 9. Al crear un equipo informático, se deberá ingresar un UUID único, un identificador único y un estado para determinar su habilitación.
 10. El administrador debe poder seleccionar el único laboratorio al que estará asociado cada equipo informático durante el proceso de creación.
 11. Después de crear un equipo informático, este debe mostrarse correctamente en la lista de equipos con el único laboratorio al que está asociado.
 12. Se realizarán pruebas para verificar que los equipos informáticos se creen correctamente y que todos los atributos ingresados sean válidos.
 13. Se documentarán los pasos para crear equipos informáticos en el sistema para futuras referencias y mantenimiento del sistema.
-

Historia de usuario

Identificador: 005	Nombre: Editar equipo informático
Usuario: Administrador	
Iteración asignada: 002	Referencia de requisito: RF005
Fecha de creación: 23/11/2023	Fecha de entrega: 23/11/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Media

Descripción:

COMO administrador
QUIERO poder editar la información de equipos informáticos existentes
PARA asegurar que la información registrada sea precisa y actualizada.

Observaciones:

- La capacidad de editar la información de equipos informáticos existentes permitirá mantener los registros actualizados y precisos dentro del sistema.
 - Se deben proporcionar opciones para modificar los atributos de cada equipo, asegurando que los cambios realizados sean válidos y coherentes.
 - Los atributos que se pueden modificar incluyen un UUID único, un identificador único, un estado para determinar su habilitación y una referencia al único laboratorio al que está asociado.
 - Es esencial que el proceso de edición de la información de equipos informáticos sea claro y sencillo para el administrador del sistema.
-

Criterios de aceptación:

1. El sistema debe proporcionar una interfaz intuitiva para que el administrador pueda editar la información de equipos informáticos existentes.
 2. Se debe permitir la modificación de los atributos de cada equipo, incluyendo el UUID, el identificador, el estado de habilitación y la referencia a un único laboratorio.
 3. Al realizar cambios en la información de un equipo, se debe validar que los nuevos valores ingresados sean válidos y coherentes.
 4. Después de editar la información de un equipo, los cambios realizados deben reflejarse correctamente en el sistema.
 5. Se realizarán pruebas para verificar que la edición de la información de equipos informáticos se realice correctamente y que los cambios sean válidos.
 6. Se documentarán los pasos para editar la información de equipos informáticos en el sistema para futuras referencias y mantenimiento del sistema.
-

Historia de usuario

Identificador: 006	Nombre: Dar de baja equipo informático
Usuario: Administrador	
Iteración asignada: 002	Referencia de requisito: RF006
Fecha de creación: 24/11/2023	Fecha de entrega: 24/11/2023
Prioridad en el negocio (Alta/Media/Baja): Baja	Riesgo en el desarrollo (Alta/Media/Baja): Baja

Descripción:

COMO administrador

QUIERO poder eliminar un equipo informático del sistema

PARA mantener actualizada la base de datos y eliminar equipos obsoletos o no deseados.

Observaciones:

- La capacidad de eliminar equipos informáticos del sistema permitirá mantener la base de datos limpia y actualizada.
- Se debe proporcionar una opción clara y segura para que el administrador pueda eliminar equipos informáticos de manera eficiente.
- La eliminación de un equipo informático debe ser irreversible y debe tener en cuenta cualquier impacto en otros registros del sistema.
- Es esencial que el proceso de eliminación de equipos informáticos sea seguro y no cause pérdida de datos o errores en el sistema.

Criterios de aceptación:

1. El sistema debe proporcionar una opción clara en la interfaz de usuario para que el administrador pueda eliminar un equipo informático del sistema.
2. Al seleccionar la opción de eliminar, el administrador debe confirmar la acción para evitar eliminaciones accidentales.
3. Al eliminar un equipo informático, se deben eliminar todos los registros asociados de manera segura y sin dejar rastros en la base de datos.
4. La eliminación de un equipo informático no debe causar errores en otros registros del sistema ni afectar la funcionalidad general.
5. Después de eliminar un equipo informático, este ya no debe aparecer en la lista de equipos asociados al laboratorio correspondiente.
6. Se realizarán pruebas para verificar que la eliminación de equipos informáticos se realice correctamente y de manera segura.
7. Se documentará el proceso de eliminación de equipos informáticos en el sistema para futuras referencias y mantenimiento del sistema.

Historia de usuario

Identificador: 007	Nombre: Listar equipos informáticos
Usuario: Administrador	
Iteración asignada: 002	Referencia de requisito: RF007
Fecha de creación: 25/11/2023	Fecha de entrega: 25/11/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Baja

Descripción:

COMO administrador

QUIERO poder visualizar una lista de todos los equipos informáticos registrados

PARA tener un panorama completo de los recursos informáticos disponibles en el sistema.

Observaciones:

- La capacidad de visualizar una lista de todos los equipos informáticos registrados permitirá al administrador gestionar eficazmente los recursos informáticos en el sistema.
 - Se deben proporcionar opciones para buscar equipos informáticos específicos y aplicar filtros según las preferencias del usuario, como identificador, UUID, dirección IP, laboratorio al que pertenece y estado del equipo informático.
 - Es esencial que la lista de equipos informáticos sea completa, precisa y actualizada en todo momento.
-

Criterios de aceptación:

1. El sistema debe proporcionar una interfaz clara y accesible para que el administrador pueda visualizar una lista de todos los equipos informáticos registrados.
 2. Se deben mostrar todos los atributos relevantes de cada equipo informático en la lista, incluyendo el identificador, UUID, la dirección IP, el laboratorio al que pertenece y el estado del equipo.
 3. Se debe proporcionar una función de búsqueda que permita al administrador encontrar equipos informáticos específicos ingresando criterios de búsqueda concretos.
 4. Se deben proporcionar opciones de filtrado que permitan al administrador aplicar filtros según las preferencias, como el identificador, UUID, la dirección IP, laboratorio al que pertenece o el estado del equipo informático.
 5. Después de aplicar filtros o realizar una búsqueda, la lista de equipos informáticos debe actualizarse dinámicamente para mostrar los resultados correspondientes.
 6. Se realizarán pruebas para verificar que la visualización de la lista de equipos informáticos y las funciones de búsqueda y filtrado funcionen correctamente.
 7. Se documentará la funcionalidad de visualización de la lista de equipos informáticos en el sistema para futuras referencias y mantenimiento del sistema.
-

Historia de usuario

Identificador: 008	Nombre: Generar reporte de equipos informáticos
Usuario: Administrador	
Iteración asignada: 002	Referencia de requisito: RF008
Fecha de creación: 26/11/2023	Fecha de entrega: 26/11/2023
Prioridad en el negocio (Alta/Media/Baja): Media	Riesgo en el desarrollo (Alta/Media/Baja): Media

Descripción:

COMO administrador

QUIERO poder generar un reporte de los equipos informáticos registrados en el sistema

PARA realizar un análisis detallado de los equipos y facilitar su gestión y seguimiento.

Observaciones:

- La capacidad de generar un reporte de equipos informáticos permitirá al administrador obtener una visión completa de los equipos en el sistema y realizar análisis detallados.
- Se deben proporcionar opciones claras y accesibles para que el administrador pueda definir los criterios de agrupación y personalizar el reporte según sus necesidades.
- Es esencial que el reporte de equipos informáticos sea completo, preciso y esté formateado de manera legible y comprensible.

Criterios de aceptación:

1. El sistema debe proporcionar una interfaz intuitiva para que el administrador pueda generar un reporte de equipos informáticos.
2. El reporte de equipos informáticos debe incluir todos los registros relevantes, como el UUID, identificador único, estado de habilitación y referencia al laboratorio correspondiente.
3. El administrador debe poder seleccionar y definir los criterios de agrupación para el reporte de equipos informáticos, como agrupar por laboratorio, estado de habilitación, etc.
4. El reporte de equipos informáticos debe estar formateado de manera clara y legible, con la capacidad de exportar los datos si es necesario.
5. Se debe garantizar que el reporte de equipos informáticos refleje con precisión los datos registrados en el sistema y que sea actualizado en tiempo real.
6. Se realizarán pruebas para verificar que el proceso de generación de reportes funcione correctamente y que los criterios de agrupación sean efectivos.
7. Se documentará el proceso de generación de reportes de equipos informáticos para futuras referencias y mantenimiento del sistema.

Historia de usuario

Identificador: 009 **Nombre:** Imprimir reporte de equipos informáticos

Usuario: Administrador

Iteración asignada: 002 **Referencia de requisito:** RF009

Fecha de creación: 27/11/2023 **Fecha de entrega:** 27/11/2023

Prioridad en el negocio (Alta/Media/Baja): Media **Riesgo en el desarrollo (Alta/Media/Baja):** Media

Descripción:

COMO administrador

QUIERO poder imprimir o descargar en formato PDF el reporte de equipos informáticos generado PARA obtener una versión física o digital del mismo para su referencia futura.

Observaciones:

- La capacidad de imprimir o descargar en formato PDF el reporte de equipos informáticos permitirá al administrador tener una copia del reporte para su referencia futura o para compartir con otros usuarios.
- Se debe proporcionar una opción clara y accesible para que el administrador pueda imprimir o descargar el reporte en formato PDF.
- Es esencial que el reporte en formato PDF refleje de manera precisa y completa la información presentada en el reporte generado en el sistema.

Criterios de aceptación:

1. El sistema debe proporcionar una opción clara y accesible para que el administrador pueda imprimir o descargar el reporte de equipos informáticos en formato PDF.
2. El reporte de equipos informáticos en formato PDF debe incluir todos los registros relevantes, como el UUID, identificador, estado, referencia al laboratorio al que pertenece.
3. El reporte de equipos informáticos en formato PDF debe mantener el mismo formato y estructura que en la versión en línea para garantizar la consistencia y la legibilidad de la información.
4. El sistema debe poder imprimir el reporte de equipos informáticos en papel o descargarlo en formato PDF según sus preferencias.
5. Se debe garantizar que el proceso de impresión o descarga del reporte de equipos informáticos funcione correctamente en todos los navegadores y dispositivos compatibles.
6. Se realizarán pruebas para verificar que el reporte de equipos informáticos se imprima o descargue en formato PDF de manera adecuada y que todos los datos se visualicen correctamente.
7. Se documentará el proceso de impresión o descarga en formato PDF del reporte de equipos informáticos para futuras referencias y mantenimiento del sistema.

Historia de usuario

Identificador: 010	Nombre: Registrar usuario e imagen de usuario
Usuario: Administrador	
Iteración asignada: 003	Referencia de requisito: RF010
Fecha de creación: 28/11/2023	Fecha de entrega: 08/12/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Alta

Descripción:

COMO administrador

QUIERO poder crear un nuevo usuario junto con su imagen asociada en el servidor

PARA utilizarla posteriormente en el reconocimiento facial y gestionar eficientemente los usuarios del sistema.

Observaciones:

- La capacidad de crear un nuevo usuario junto con su imagen asociada en el servidor permitirá al administrador gestionar eficazmente las cuentas de usuario en el sistema.
 - Se debe proporcionar una opción clara y sencilla para que el administrador pueda cargar la imagen asociada al nuevo usuario durante el proceso de creación de la cuenta.
 - Es esencial que la imagen asociada se almacene de manera segura en el servidor para su posterior uso en el reconocimiento facial.
 - El proceso de creación de un nuevo usuario debe ser intuitivo y eficiente para el administrador.
-

Criterios de aceptación:

1. El sistema debe proporcionar una interfaz clara y accesible para que el administrador pueda crear un nuevo usuario.
 2. Durante el proceso de creación de la cuenta de usuario, se debe proporcionar una opción para que el administrador pueda cargar una imagen asociada desde su dispositivo local.
 3. La imagen subida debe de poseer al menos un rostro humano para que se considere válida.
 4. La imagen asociada al nuevo usuario debe almacenarse de manera segura en el servidor y estar vinculada correctamente a la cuenta de usuario correspondiente.
 5. Después de crear el nuevo usuario, la imagen asociada debe estar disponible para su posterior uso en el reconocimiento facial.
 6. Se realizarán pruebas para verificar que el proceso de creación de un nuevo usuario y la carga de la imagen asociada funcionen correctamente.
 7. Se documentará el proceso de creación de un nuevo usuario y la gestión de la imagen asociada en el sistema para futuras referencias y mantenimiento del sistema.
-

Historia de usuario

Identificador: 011	Nombre: Editar usuario e imagen de usuario
Usuario: Administrador	
Iteración asignada: 003	Referencia de requisito: RF011
Fecha de creación: 09/12/2023	Fecha de entrega: 10/12/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Alta

Descripción:

COMO administrador

QUIERO poder editar la información de un usuario existente, incluyendo la posibilidad de modificar su imagen

PARA mantener actualizados y precisos los datos de los usuarios y gestionar eficientemente sus cuentas.

Observaciones:

- La capacidad de editar la información de un usuario existente, incluyendo su imagen asociada en el servidor, permitirá al administrador mantener actualizados los registros de usuario en el sistema.
- Se debe proporcionar una opción clara y sencilla para que el administrador pueda modificar la imagen asociada a un usuario durante el proceso de edición de la cuenta.
- Es esencial que la imagen modificada se actualice correctamente en el servidor y esté vinculada correctamente a la cuenta de usuario correspondiente.
- El proceso de edición de la información de un usuario existente debe ser intuitivo y eficiente para el administrador.

Criterios de aceptación:

1. El sistema debe proporcionar una interfaz clara y accesible para que el administrador pueda editar la información de un usuario existente.
2. Durante el proceso de edición de la cuenta de usuario, se debe proporcionar una opción para que el administrador pueda modificar la imagen asociada desde su dispositivo local.
3. La imagen subida debe de poseer al menos un rostro humano para que se considere válida.
4. La imagen modificada asociada al usuario debe actualizarse correctamente en el servidor y estar vinculada correctamente a la cuenta de usuario correspondiente.
5. Después de editar la información de un usuario, incluyendo la imagen asociada, los cambios realizados deben reflejarse correctamente en el sistema.
6. Se realizarán pruebas para verificar que el proceso de edición de la información de un usuario y la modificación de la imagen asociada funcionen correctamente.
7. Se documentará el proceso de edición de la información de un usuario y la gestión de la imagen asociada en el sistema para futuras referencias y mantenimiento del sistema.

Historia de usuario

Identificador: 012	Nombre: Dar de baja usuario e imagen de usuario
Usuario: Administrador	
Iteración asignada: 003	Referencia de requisito: RF012
Fecha de creación: 11/12/2023	Fecha de entrega: 11/12/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Alta

Descripción:

COMO administrador

QUIERO poder eliminar un usuario del sistema, incluyendo la eliminación de su imagen asociada en el servidor

PARA mantener actualizada la base de datos de usuarios y gestionar eficientemente las cuentas de usuario.

Observaciones:

- La capacidad de eliminar un usuario del sistema, junto con su imagen asociada, permitirá al administrador mantener la base de datos de usuarios limpia y actualizada.
- Se debe proporcionar una opción clara y segura para que el administrador pueda eliminar un usuario del sistema, asegurándose de que la imagen asociada en el servidor también se elimine correctamente.
- Es esencial que la eliminación de un usuario y su imagen asociada en el servidor se realice de manera segura y sin dejar rastros en el servidor.

Criterios de aceptación:

1. El sistema debe proporcionar una interfaz clara y accesible para que el administrador pueda eliminar un usuario del sistema.
2. Al seleccionar la opción de eliminar, el administrador debe confirmar la acción para evitar eliminaciones accidentales.
3. Al eliminar un usuario, se deben eliminar todos los registros asociados de manera segura, incluyendo su imagen asociada almacenada en el servidor.
4. La eliminación de un usuario no debe causar errores en otros registros del sistema ni afectar la funcionalidad general.
5. Después de eliminar un usuario, este ya no debe aparecer en la lista de usuarios registrados en el sistema.
6. Se realizarán pruebas para verificar que la eliminación de usuarios y la eliminación de sus imágenes asociadas se realicen correctamente y de manera segura.
7. Se documentará el proceso de eliminación de usuarios y la eliminación de imágenes asociadas en el sistema para futuras referencias y mantenimiento del sistema.

Historia de usuario

Identificador: 013	Nombre: Listar usuarios
Usuario: Administrador	
Iteración asignada: 003	Referencia de requisito: RF032
Fecha de creación: 12/12/2023	Fecha de entrega: 12/12/2023
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Baja

Descripción:

COMO administrador

QUIERO poder visualizar una lista de todas las cuentas de usuarios registradas

PARA tener un panorama completo de los usuarios del sistema y gestionar eficientemente sus cuentas.

Observaciones:

- La capacidad de visualizar una lista de todas las cuentas de usuarios registradas permitirá al administrador gestionar eficazmente las cuentas de usuario en el sistema.
- Se debe proporcionar una opción clara y accesible para que el administrador pueda acceder a la lista de usuarios registrados, así como ver sus imágenes asociadas en el servidor.
- Se deben proporcionar opciones para buscar cuentas de usuario específicas y aplicar filtros según las preferencias del usuario, como nombre y correo electrónico.
- Es esencial que la lista de usuarios registrados sea completa, precisa y actualizada en todo momento.

Criterios de aceptación:

1. El sistema debe proporcionar una interfaz clara y accesible para que el administrador pueda acceder a una lista de todas las cuentas de usuarios registradas.
2. En la lista de usuarios registrados, se deben mostrar todos los atributos relevantes de cada usuario, incluyendo su nombre, dirección de correo electrónico, imagen asociada en el servidor y rol.
3. Se debe garantizar que las imágenes asociadas a cada usuario se carguen correctamente en la lista de usuarios registrados.
4. Después de acceder a la lista de usuarios registrados, el administrador debe poder realizar acciones adicionales como editar, eliminar o filtrar usuarios según sea necesario.
5. Se debe proporcionar una función de búsqueda que permita al administrador encontrar cuentas de usuario específicas ingresando criterios de búsqueda concretos.
6. Se deben proporcionar opciones de filtrado que permitan al administrador aplicar filtros según las preferencias, como nombre y correo electrónico.
7. La lista de usuarios registrados debe actualizarse dinámicamente para reflejar cualquier cambio realizado en las cuentas de usuario o después de aplicar filtros o realizar una búsqueda.
8. Se realizarán pruebas para verificar que la lista de usuarios registrados se muestre correctamente y que las imágenes asociadas se carguen sin problemas.
9. Se documentará el proceso de visualización de la lista de usuarios registrados y la gestión de imágenes asociadas en el sistema para futuras referencias y mantenimiento del sistema.

Historia de usuario

Identificador: 014	Nombre: Registrar asistencia Face ID
Usuario: Docente, Estudiante, Administrador	
Iteración asignada: 004	Referencia de requisito: RF014
Fecha de creación: 13/12/2023	Fecha de entrega: 10/01/2024
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Alta

Descripción:

COMO usuario (docente, estudiante o administrador)

QUIERO poder registrar mi asistencia utilizando la cámara web para el reconocimiento facial

PARA agilizar el proceso de registro de asistencia y garantizar su precisión.

Observaciones:

- La capacidad de registrar la asistencia mediante el reconocimiento facial permitirá a los usuarios realizar este proceso de manera rápida y precisa.
- Se debe proporcionar un botón claro y accesible en la interfaz para iniciar el proceso de registro de asistencia utilizando la cámara web.
- Es esencial que el sistema pueda reconocer y validar la identidad del usuario de manera confiable mediante el reconocimiento facial.

Criterios de aceptación:

1. El sistema debe proporcionar un botón claro y accesible en la interfaz para que los usuarios puedan iniciar el proceso de registro de asistencia mediante el reconocimiento facial.
2. Se debe validar que sólo el administrador puede registrar asistencia a cualquier hora, mientras que, tanto el docente como el estudiante, sólo pueden registrar asistencia siempre y cuando se encuentren dentro del horario establecido.
3. Se debe de validar el registro de asistencia según la configuración del laboratorio, es decir: si está habilitado el rango ip, se valida la dirección ip del equipo; si está habilitado los equipos informáticos, se valida el UUID del equipo; si están habilitados ambos, se validan ambas direcciones; y si están deshabilitadas ambas, no se valida ninguna de las dos direcciones y se registra las direcciones sin importar cuáles sean.
4. Se debe validar el registro de asistencia según la configuración del equipo informático, es decir: si está habilitado el equipo, se podrá registrar asistencia; caso contrario, no se registra asistencia y se presenta que el equipo está deshabilitado.
5. Al hacer clic en el botón de registro de asistencia, el sistema debe activar la cámara web del dispositivo del usuario y comenzar el proceso de reconocimiento facial.
6. El sistema debe poder reconocer y validar la identidad del usuario utilizando el reconocimiento facial de manera confiable.
7. Después de completar el proceso de reconocimiento facial, el sistema debe registrar la asistencia del usuario de manera automática y precisa.
8. Se debe proporcionar retroalimentación clara al usuario durante todo el proceso de registro de asistencia mediante el reconocimiento facial.
9. Se realizarán pruebas para verificar que el proceso de registro de asistencia mediante el reconocimiento facial funcione correctamente y de manera confiable.
10. Se documentará el proceso de registro de asistencia mediante el reconocimiento facial en el sistema para futuras referencias y mantenimiento del sistema.

Historia de usuario

Identificador: 015	Nombre: Listar asistencias Face ID
Usuario: Docente, Estudiante, Administrador	
Iteración asignada: 004	Referencia de requisito: RF015
Fecha de creación: 11/01/2024	Fecha de entrega: 11/01/2024
Prioridad en el negocio (Alta/Media/Baja): Alta	Riesgo en el desarrollo (Alta/Media/Baja): Baja

Descripción:

COMO usuario (docente, estudiante o administrador)
QUIERO poder visualizar una lista detallada de todas las asistencias Face ID registradas
PARA tener un seguimiento preciso de la asistencia y poder realizar análisis específicos según mis necesidades.

Observaciones:

- La capacidad de visualizar una lista detallada de todas las asistencias Face ID registradas permitirá a los usuarios tener un control completo sobre el registro de asistencia en el sistema.
- Se deben proporcionar opciones claras y accesibles para que los usuarios puedan buscar registros específicos y aplicar filtros según sus preferencias individuales.
- Es esencial que la lista de asistencias Face ID registradas sea completa, precisa y esté actualizada en todo momento.

Criterios de aceptación:

1. El sistema debe proporcionar una interfaz clara y accesible para que los usuarios puedan acceder a una lista detallada de todas las asistencias Face ID registradas.
2. En la lista de asistencias Face ID registradas, se deben mostrar todos los atributos relevantes de cada registro, como materia, responsable, usuario, fecha y hora, período académico, dirección IP, UUID, rol y laboratorio.
3. Se debe proporcionar opciones de búsqueda para que los usuarios puedan encontrar registros específicos según diferentes criterios, como materia, responsable, usuario, fecha y hora, período académico, dirección IP, UUID, rol y laboratorio.
4. Los usuarios deben poder aplicar varios filtros para refinar la lista de asistencias Face ID según sus preferencias individuales.
5. La lista de asistencias Face ID registradas debe actualizarse dinámicamente para reflejar cualquier cambio o adición de nuevos registros.
6. Se realizarán pruebas para verificar que la lista de asistencias Face ID registradas se muestre correctamente y que las opciones de búsqueda y filtrado funcionen según lo esperado.
7. Se documentará el proceso de visualización de la lista de asistencias Face ID registradas y la aplicación de filtros en el sistema para futuras referencias y mantenimiento del sistema.

Historia de usuario

Identificador: 016	Nombre: Generar reporte de asistencias Face ID
Usuario: Docente, Estudiante, Administrador	
Iteración asignada: 004	Referencia de requisito: RF016
Fecha de creación: 12/01/2024	Fecha de entrega: 12/01/2024
Prioridad en el negocio (Alta/Media/Baja): Media	Riesgo en el desarrollo (Alta/Media/Baja): Media

Descripción:

COMO usuario (docente, estudiante o administrador)
QUIERO poder generar un reporte de asistencias Face ID
PARA realizar un seguimiento detallado de la asistencia y poder analizar los datos de manera efectiva,
con la opción de agrupar los registros según sea necesario.

Observaciones:

- La capacidad de generar un reporte de asistencias Face ID permitirá a los usuarios obtener una visión completa de la asistencia en el sistema y realizar análisis detallados.
- Se deben proporcionar opciones claras y accesibles para que los usuarios puedan definir los criterios de agrupación y personalizar el reporte según sus necesidades.
- Es esencial que el reporte de asistencias Face ID sea completo, preciso y esté formateado de manera legible y comprensible.

Criterios de aceptación:

1. El sistema debe proporcionar una interfaz intuitiva para que los usuarios puedan generar un reporte de asistencias Face ID.
2. El reporte de asistencias Face ID debe incluir todos los registros relevantes, como materia, docente, estudiante, hora, periodo académico, dirección IP, UUID, rol y laboratorio.
3. Los usuarios deben poder seleccionar y definir los criterios de agrupación para el reporte de asistencias, como agrupar por materia, docente, estudiante, periodo académico, etc.
4. El reporte de asistencias Face ID debe estar formateado de manera clara y legible, con la capacidad de exportar los datos en formato PDF.
5. Se debe garantizar que el reporte de asistencias Face ID refleje con precisión los datos registrados en el sistema y que sea actualizado en tiempo real.
6. Se realizarán pruebas para verificar que el proceso de generación de reportes funcione correctamente y que los criterios de agrupación sean efectivos.
7. Se documentará el proceso de generación de reportes de asistencias Face ID para futuras referencias y mantenimiento del sistema.

Historia de usuario

Identificador: 017	Nombre: Imprimir reporte de asistencias Face ID
Usuario: Docente, Estudiante, Administrador	
Iteración asignada: 004	Referencia de requisito: RF017
Fecha de creación: 13/01/2024	Fecha de entrega: 13/01/2024
Prioridad en el negocio (Alta/Media/Baja): Media	Riesgo en el desarrollo (Alta/Media/Baja): Media

Descripción:

COMO usuario (docente, estudiante o administrador)
QUIERO poder imprimir o descargar en formato PDF el reporte de asistencias Face ID generado
PARA obtener una versión física o digital del mismo para su referencia futura.

Observaciones:

- La capacidad de imprimir o descargar el reporte de asistencias Face ID en formato PDF permitirá a los usuarios tener una copia física o digital del mismo para su referencia futura.
- Se deben proporcionar opciones claras y accesibles para que los usuarios puedan realizar esta acción de manera sencilla y sin complicaciones.
- Es esencial que el reporte de asistencias Face ID se pueda visualizar correctamente en formato PDF y que mantenga el mismo formato y estructura que en la versión en línea.

Criterios de aceptación:

1. El sistema debe proporcionar una opción clara y accesible para que los usuarios puedan imprimir o descargar en formato PDF el reporte de asistencias Face ID generado.
2. El reporte de asistencias Face ID en formato PDF debe incluir todos los registros relevantes, como materia, docente, estudiante, hora, período académico, dirección IP, UUID, rol y laboratorio.
3. El reporte de asistencias Face ID en formato PDF debe mantener el mismo formato y estructura que en la versión en línea para garantizar la consistencia y la legibilidad de la información.
4. Los usuarios deben poder imprimir el reporte de asistencias Face ID en papel o descargarlo en formato PDF según sus preferencias.
5. Se debe garantizar que el proceso de impresión o descarga del reporte de asistencias Face ID funcione correctamente en todos los navegadores y dispositivos compatibles.
6. Se realizarán pruebas para verificar que el reporte de asistencias Face ID se imprima o descargue en formato PDF de manera adecuada y que todos los datos se visualicen correctamente.
7. Se documentará el proceso de impresión o descarga del reporte de asistencias Face ID en formato PDF para futuras referencias y mantenimiento del sistema.

Para constancia y validación de la información expuesta, firma a continuación el tutor del Trabajo de Integración Curricular, en su calidad de docente guía en el desarrollo del proyecto:

Ing. Edison Leonardo Coronel Romero, Mg.Sc.



ANEXO 4: Implementación de metodología XP



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

• **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

• **Revisado y aprobado por:**

- Ing. Edison Leonardo Coronel Romero, Mg.Sc., tutor del Trabajo de Integración Curricular.

Descripción: El presente anexo detalla la implementación de la metodología XP (Extreme Programming) en el desarrollo del software del TIC. El proceso se estructuró siguiendo el plan de iteración establecido, el cual divide el desarrollo del sistema en cuatro iteraciones distintas, donde cada iteración abarca las fases principales de la metodología XP: planificación, diseño, codificación y pruebas. Cabe destacar que, la fase de lanzamiento, no se incluyó en las iteraciones, sino, que se realizó como un único evento de puesta en producción al final del ciclo de desarrollo. El enfoque iterativo e incremental permitió una adaptación continua a las necesidades del sistema, facilitando una evolución constante del software basada en los resultados de cada iteración.

IMPLEMENTACIÓN DE LA METODOLOGÍA XP

1 Primera Iteración

1.1 Planificación

La **Tabla 1** muestra las actividades específicas a desarrollar en la primera iteración.

Tabla 1. Historias de Usuario a desarrollar en la primera iteración.

HU #	Historia de usuario	Fecha de creación	Fecha de entrega
HU01	Refactorizar el sistema de control de acceso	13/11/2023	17/11/2023
HU02	Actualizar gestión de laboratorio	18/11/2023	20/11/2023

1.2 Diseño

La **Figura 1** muestra las funcionalidades de manera general a desarrollar en la primera iteración.

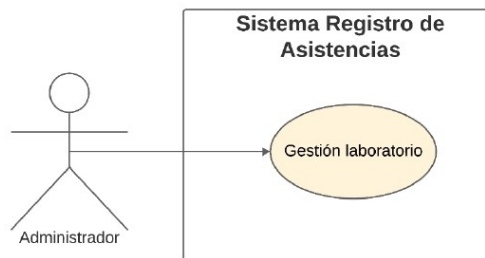


Figura 1. Casos de Uso general para la primera iteración.

1.2.1 Diagramas BPMN

A continuación, se presentan los diagramas BPMN pertenecientes a cada Historia de Usuario a desarrollar en la primera iteración.

1.2.1.1 HU01: Refactorizar el sistema de control de acceso

Es relevante señalar que el diagrama BPMN está dirigido al desarrollador y no a un usuario o rol específico del sistema, ya que esta historia de usuario se enfoca en la refactorización del sistema actual.

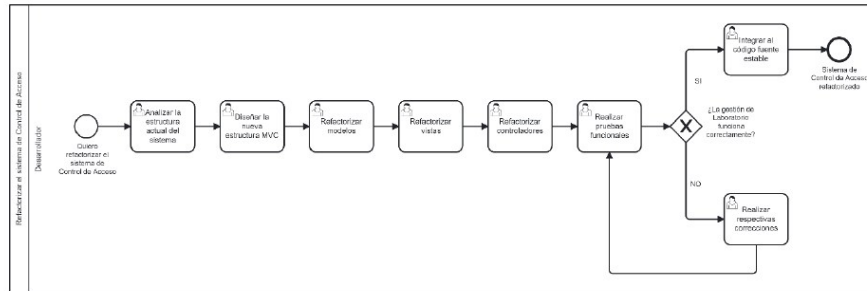


Figura 2. Diagrama BPMN de HU01.

1.2.1.2 HU02: Actualizar gestión de laboratorio

Es importante destacar que el diagrama BPMN está dirigido al desarrollador, y no a un tipo de usuario o rol del sistema. Esto se debe a que la historia de usuario en cuestión se refiere a una actualización del componente "Laboratorio" dentro del sistema actual.

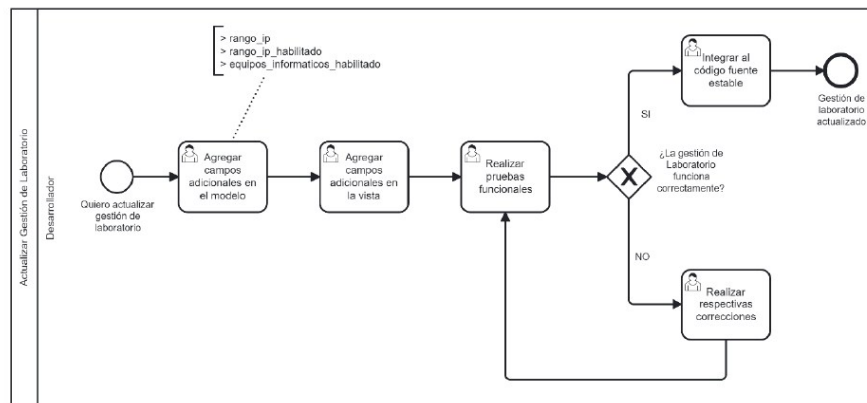


Figura 3. Diagrama BPMN de HU02.

1.2.2 Prototipos

A continuación, se presentan los prototipos pertenecientes a cada Historia de Usuario a desarrollar en la primera iteración.

1.2.2.1 HU01: Refactorizar el sistema de control de acceso

Es relevante señalar que, al tratarse de una refactorización, el prototipo refleja el patrón MVC, que es el objetivo a refactorizar. Esto se debe a que, aunque Odoos utiliza dicho patrón por defecto, en el sistema actual su implementación no es la adecuada.

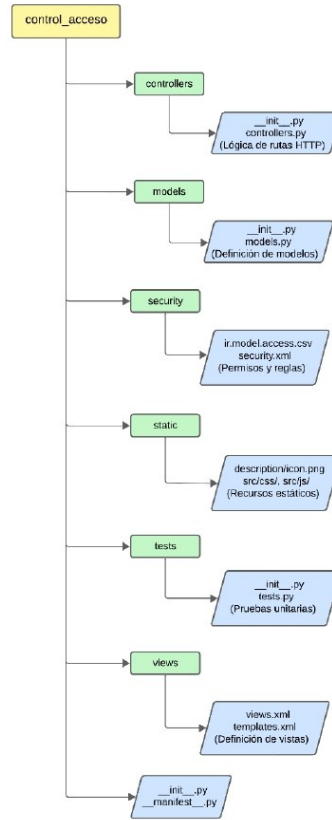


Figura 4. Prototipo para refactorizar el sistema actual en base al patrón MVC.

1.2.2.2 HU02: Actualizar gestión de laboratorio

Laboratorios

Laboratorios / Nuevo

Grabar Descartar

Número del Laboratorio

Nombre del Laboratorio

Rango de IP del Laboratorio Ejemplo: 192.168.0.110/24

Habilitar Rango IP

Habilitar Equipos Informáticos

Figura 5. Prototipo para Registrar Laboratorio.

Laboratorios

Laboratorios / Editar

Grabar Descartar

Número del Laboratorio 1

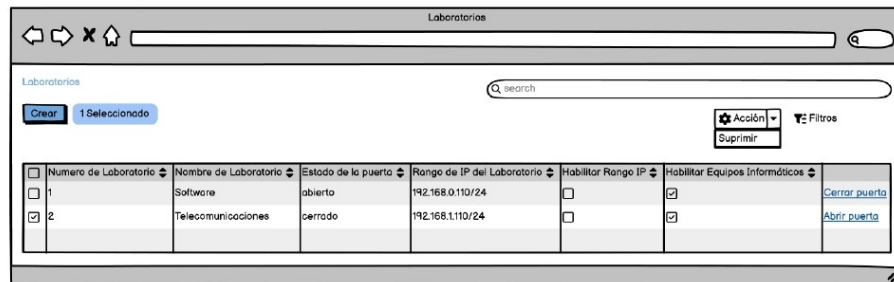
Nombre del Laboratorio Software

Rango de IP del Laboratorio 192.168.0.110/24

Habilitar Rango IP

Habilitar Equipos Informáticos

Figura 6. Prototipo para Editar Laboratorio.



<input type="checkbox"/>	Numero de Laboratorio	Nombre de Laboratorio	Estado de la puerta	Rango de IP del Laboratorio	Habilitar Rango IP	Habilitar Equipos Informáticos	
<input type="checkbox"/>	1	Software	abierta	192.168.0.110/24	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cerrar puerta
<input checked="" type="checkbox"/>	2	Telecomunicaciones	cerrada	192.168.1.110/24	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Abrir puerta

Figura 7. Prototipo para Listar y Dar de baja Laboratorio.

1.3 Codificación

A continuación, se presenta los fragmentos principales del código fuente desarrollados en la primera iteración.

1.3.1.1 HU01: Refactorizar el sistema de control de acceso

La refactorización del sistema actual depende en gran medida de la correcta implementación del patrón MVC (Modelo-Vista-Controlador). Por esta razón, todos los modelos que anteriormente se encontraban en un solo archivo fueron reorganizados en archivos separados, siguiendo el principio de independencia. Del mismo modo, las vistas fueron estructuradas de forma individual, incluyendo la vista correspondiente al menú, con el objetivo de mejorar la claridad y modularidad del código.

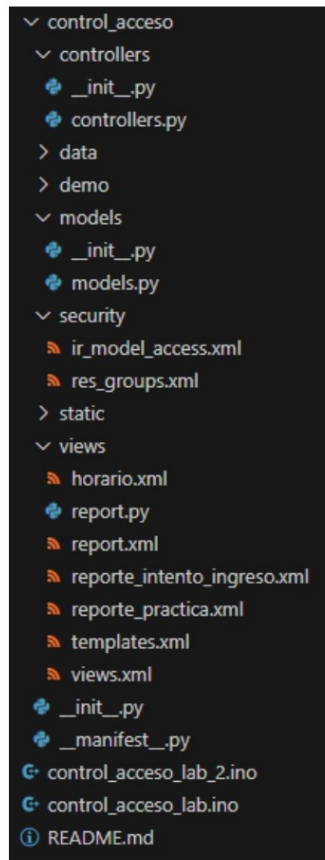


Figura 8. Estructura del módulo antes de la refactorización.

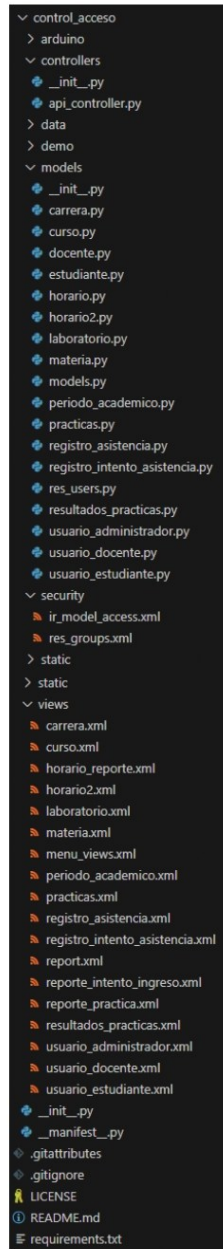


Figura 9. Estructura del módulo después de la refactorización.


```

1 # coding: utf-8
2 {
3   "name": "Control de Acceso",
4   "description": "Módulo para el control de acceso de los laboratorios en la carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja",
5   "category": "Admin",
6   "summary": "Módulo para el control de acceso de los laboratorios en la carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja",
7   "author": "Luis Jumbo Herrera",
8   "depends": ["base", "mail"],
9   "analytic": true,
10  "base_urls": ["portal"],
11  "routing": ["routing"],
12  "resources": ["resources"],
13  "web": ["web"],
14  "web_tour": ["web_tour"],
15  "messages": ["messages"],
16  "data": [
17    ],
18  "security": [
19    "security/rec_groups.xml",
20    "security/ir_modul_access.xml",
21    "views/carrera.xml",
22    "views/cron.xml",
23    "views/horario.xml",
24    "views/report.xml",
25    "views/report_practica.xml",
26    "views/report_intento_ingreso.xml",
27    "data/cron.xml",
28  ],
29  "images": ["static/description/icon.png"],
30 }

```

Figura 10. Manifest del módulo antes de la refactorización.

```

1 # coding: utf-8
2 {
3   "name": "Control de Acceso Pruebas",
4   "description": "Módulo para el control de acceso de los laboratorios en la carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja",
5   "category": "Admin",
6   "summary": "Módulo para el control de acceso de los laboratorios en la carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja",
7   "author": "Luis Jumbo Herrera",
8   "depends": ["base", "mail"],
9   "analytic": true,
10  "base_urls": ["portal"],
11  "routing": ["routing"],
12  "resources": ["resources"],
13  "web": ["web"],
14  "web_tour": ["web_tour"],
15  "messages": ["messages"],
16  "data": [
17    "security/rec_groups.xml",
18    "security/ir_modul_access.xml",
19    "views/carrera.xml",
20    "views/cron.xml",
21    "views/horario_reporte.xml",
22    "views/horario.xml",
23    "views/laboratorio.xml",
24    "views/asteris.xml",
25    "views/periodo_academico.xml",
26    "views/practica.xml",
27    "views/report_registro_asistencia_face_id.xml",
28    "views/report_registro_asistencia_face_id.xml",
29    "views/report_registro_asistencia_face_id.xml",
30    "views/report_registro_asistencia_face_id.xml",
31    "views/report_registro_asistencia_face_id.xml",
32    "views/report_registro_asistencia_face_id.xml",
33    "views/report_registro_asistencia_face_id.xml",
34    "views/report_registro_asistencia_face_id.xml",
35    "views/report_registro_asistencia_face_id.xml",
36    "views/report_registro_asistencia_face_id.xml",
37    "views/report_registro_asistencia_face_id.xml",
38    "views/report_registro_asistencia_face_id.xml",
39    "views/report_registro_asistencia_face_id.xml",
40    "views/report_registro_asistencia_face_id.xml",
41    "views/report_registro_asistencia_face_id.xml",
42    "views/report_registro_asistencia_face_id.xml",
43    "views/report_registro_asistencia_face_id.xml",
44    "views/report_registro_asistencia_face_id.xml",
45  ],
46  "images": ["static/description/icon.png"],
47  "category": "Administration",
48  "application": true,
49  "installable": true,
50  "license": "GPL-3",
51 }

```

Figura 11. Manifest del módulo después de la refactorización.

```

1 class Carrera(models.Model):
2     _name = 'controlacceso.carrera'
3     _description = 'registro de carreras'
4     name = fields.CharField('Carrera', required=True)
5     @api.onchange('name')
6     def validarCarrera(self):
7         lista = self.env['controlacceso.carrera'].search([])
8         for i in range(len(lista)):
9             if lista[i].name == self.name:
10                raise ValidationError('La carrera con el nombre: '+ self.name+ ', ya esta registrada')
11
12 class Curso(models.Model):
13     _name = 'controlacceso.curso'
14     name = fields.CharField('Nombre curso', required=True)
15     carrera_id = fields.Many2one(
16         "controlacceso.carrera", string="Carrera",
17         default=lambda self: self.env['controlacceso.carrera'].search([], limit=1),
18         ondelete="cascade")
19     carrera = fields.CharField(related='carrera_id.name', string="Nombre Carrera",)
20
21     @api.onchange('name')
22     def validarCurso(self):
23         print("ingreso al validar curso")
24         listaCursos = self.env['controlacceso.curso'].search([])
25         print("entro al validar curso")
26         for i in range(len(listaCursos)):
27             print(self.name)
28             print(listaCursos[i].name)
29             if listaCursos[i].name == self.name:
30                 raise ValidationError('El curso con el nombre: '+ self.name+ ', ya esta registrado')
31
32 class Materia(models.Model):
33     _name = 'controlacceso.materia'
34     name = fields.CharField('Nombre materia', required=True)
35     carrera_id = fields.Many2one(
36         "controlacceso.carrera", string="Carrera",
37         default=lambda self: self.env['controlacceso.carrera'].search([], limit=1),
38         ondelete="cascade")
39
40 class Docente(models.Model):
41     _name = 'controlacceso.docente'
42     cedula = fields.CharField('Cedula', required=True)
43     name = fields.CharField('Nombres y Apellidos', required=True)
44     correo_docente = fields.CharField('Correo', required=True)
45     id_tarjeta = fields.CharField('Tarjeta', required=True)
46     carrera_id = fields.Many2one(
47         "controlacceso.carrera", string="Carrera",
48         default=lambda self: self.env['controlacceso.carrera'].search([], limit=1),
49         ondelete="cascade")
50
51     @api.onchange('correo_docente')
52     def validate_mail(self):
53         if self.correo_docente:
54             match = re.match('.*@unl.edu.ec', self.correo_docente)
55             if match == None:
56                 raise ValidationError('Solo se aceptan direcciones de correo pertenecientes a @unl.edu.ec')
57
58 class Estudiante(models.Model):
59     _name = 'controlacceso.estudiante'
60     cedula = fields.CharField('Cedula', required=True)
61     name = fields.CharField('Nombres y Apellidos', required=True)
62     id_tarjeta = fields.CharField('Tarjeta', required=True)
63     carrera_id = fields.Many2one(
64         "controlacceso.carrera", string="Carrera",
65         default=lambda self: self.env['controlacceso.carrera'].search([], limit=1),
66         ondelete="cascade")
67     curso_id = fields.Many2one(
68         "controlacceso.curso", string="curso",
69         default=lambda self: self.env['controlacceso.curso'].search([], limit=1),
70         ondelete="cascade")
71

```

Figura 12. Fragmentos del archivo de Modelos antes de la refactorización.

```
1 from odoo import fields, models, api, _
2 from odoo.exceptions import ValidationError
3
4 class Carrera(models.Model):
5     _name = 'controlacceso.carrera'
6     _description = 'Registro de carreras'
7     name = fields.Char('Carrera', required=True)
8
9     @api.onchange('name')
10    def validarCarrera(self):
11        lista = self.env['controlacceso.carrera'].search([])
12        for i in range(len(lista)):
13            if lista[i].name == self.name:
14                raise ValidationError('La carrera con el nombre: '+ self.name+ ', ya esta registrada')
```

Figura 13. Ejemplo del modelo "Carrera" después de la refactorización.

```
1 from odoo import fields, models
2
3 class Materia(models.Model):
4     _name = 'controlacceso.materia'
5     _description = 'Registro de materias'
6     name = fields.Char('Nombre materia', required=True)
7     carrera_id = fields.Many2one(
8         "controlacceso.carrera", string="Carrera",
9         default=lambda self: self.env['controlacceso.carrera'].search([], limit=1),
10        ondelete="cascade")
```

Figura 14. Ejemplo del modelo "Materia" después de la refactorización.

```
1 # -*- coding: utf-8 -*-  
2  
3 from . import models
```

Figura 15. Importaciones de los modelos antes de la refactorización.

```
1 # -*- coding: utf-8 -*-  
2  
3 from . import carrera  
4 from . import curso  
5 from . import docente  
6 from . import estudiante  
7 from . import horario  
8 from . import horario2  
9 from . import laboratorio  
10 from . import materia  
11 from . import periodo_academico  
12 from . import practicas  
13 from . import registro_asistencia  
14 from . import registro_intento_asistencia  
15 from . import res_users  
16 from . import resultados_practicas  
17 from . import usuario_administrador  
18 from . import usuario_docente  
19 from . import usuario_estudiante  
20 from . import registro_asistencia_face_id  
21 from . import equipo_informatico
```

Figura 16. Importaciones de los modelos después de la refactorización.


```

1 class VistaMenu {
2   constructor() {
3     // ...
4     // ...
5     // ...
6     // ...
7     // ...
8     // ...
9     // ...
10    // ...
11    // ...
12    // ...
13    // ...
14    // ...
15    // ...
16    // ...
17    // ...
18    // ...
19    // ...
20    // ...
21    // ...
22    // ...
23    // ...
24    // ...
25    // ...
26    // ...
27    // ...
28    // ...
29    // ...
30    // ...
31    // ...
32    // ...
33    // ...
34    // ...
35    // ...
36    // ...
37    // ...
38    // ...
39    // ...
40    // ...
41    // ...
42    // ...
43    // ...
44    // ...
45    // ...
46    // ...
47    // ...
48    // ...
49    // ...
50    // ...
51    // ...
52    // ...
53    // ...
54    // ...
55    // ...
56    // ...
57    // ...
58    // ...
59    // ...
60    // ...
61    // ...
62    // ...
63    // ...
64    // ...
65    // ...
66    // ...
67    // ...
68    // ...
69    // ...
70    // ...
71    // ...
72    // ...
73    // ...
74    // ...
75    // ...
76    // ...
77    // ...
78    // ...
79    // ...
80    // ...
81    // ...
82    // ...
83    // ...
84    // ...
85    // ...
86    // ...
87    // ...
88    // ...
89    // ...
90    // ...
91    // ...
92    // ...
93    // ...
94    // ...
95    // ...
96    // ...
97    // ...
98    // ...
99    // ...
100   }
101 }
  
```

Figura 18. Fragmento de la Vista después de la refactorización “Menú”.

1.3.1.2 HU02: Actualizar gestión de laboratorio

La actualización del componente [Laboratorio], compete el agregar los campos necesarios para poder manejar el rango de IP del laboratorio y los equipos informáticos asociados al mismo.

```

1 class Laboratorio {
2   constructor() {
3     // ...
4     // ...
5     // ...
6     // ...
7     // ...
8     // ...
9     // ...
10    // ...
11    // ...
12    // ...
13    // ...
14    // ...
15    // ...
16    // ...
17    // ...
18    // ...
19    // ...
20    // ...
21    // ...
22    // ...
23    // ...
24    // ...
25    // ...
26    // ...
27    // ...
28    // ...
29    // ...
30    // ...
31    // ...
32    // ...
33    // ...
34    // ...
35    // ...
36    // ...
37    // ...
38    // ...
39    // ...
40    // ...
41    // ...
42    // ...
43    // ...
44    // ...
45    // ...
46    // ...
47    // ...
48    // ...
49    // ...
50    // ...
51    // ...
52    // ...
53    // ...
54    // ...
55    // ...
56    // ...
57    // ...
58    // ...
59    // ...
60    // ...
61    // ...
62    // ...
63    // ...
64    // ...
65    // ...
66    // ...
67    // ...
68    // ...
69    // ...
70    // ...
71    // ...
72    // ...
73    // ...
74    // ...
75    // ...
76    // ...
77    // ...
78    // ...
79    // ...
80    // ...
81    // ...
82    // ...
83    // ...
84    // ...
85    // ...
86    // ...
87    // ...
88    // ...
89    // ...
90    // ...
91    // ...
92    // ...
93    // ...
94    // ...
95    // ...
96    // ...
97    // ...
98    // ...
99    // ...
100   }
  
```

Figura 19. Modelo de Laboratorio antes de la actualización.

```

1 from odoo import fields, api, models
2 from odoo.exceptions import ValidationError
3 from datetime import datetime
4 import pytz
5 import re
6 import ipaddress
7 import logging
8
9 _logger = logging.getLogger(__name__)
10
11 class Laboratorio(models.Model):
12     _name = 'controlacceso.lab'
13     _description = 'Registro de laboratorios'
14
15     name = fields.Char('Número de Laboratorio', required=True)
16     nombre_lab = fields.Char('Nombre del Laboratorio', required=True)
17     estado = fields.Char('Estado de la puerta', required=True, default="cerrado")
18     rango_ip = fields.Text('Rango de IP del Laboratorio', required=True)
19     rango_ip_habilitado = fields.Boolean('Habilitar Rango IP', required=True, default=False)
20     equipos_informaticos_habilitado = fields.Boolean('Habilitar Equipos Informáticos', required=True, default=True)
21
22     @api.constrains('rango_ip')
23     def _check_valid_ip_range(self):
24         """
25         Valida que el rango de direcciones IP ingresado sea válido.
26         """
27         for record in self:
28             if record.rango_ip:
29                 try:
30                     # Intenta crear un objeto de red a partir del rango IP
31                     network = ipaddress.IPv4Network(record.rango_ip, strict=False)
32                     _logger.info(network)
33                 except (ipaddress.AddressValueError, ValueError):
34                     raise ValidationError("El rango de direcciones IP no es válido.")
35
36     def botonconfirm(self):
37         p = self.env.context
38         nombre = p.get('active_id')
39         buscar = self.env['controlacceso.lab'].search([('name', '=', nombre)]).update({'estado': "abierto"})
40
41         utc_now = datetime.utcnow()
42
43         local_tz = pytz.timezone('America/Bogota')
44
45         local_now = utc_now.replace(tzinfo=pytz.utc).astimezone(local_tz)
46
47         day_of_week = local_now.strftime('%A')
48
49         date_str = local_now.strftime('%Y %m %d')
50
51         time_str = local_now.strftime('%H:%M')
52         print("esto se presenta", day_of_week, date_str, time_str)
53
54     @api.onchange('name')
55     def validarLab(self):
56         lista = self.env['controlacceso.lab'].search([])
57         for i in range(len(lista)):
58             print(lista[i].name)
59             if lista[i].name == self.name:
60                 raise ValidationError("El nombre el laboratorio: '+self.name+' ya esta registrado")
61
62     @api.depends('nombre_lab')
63     def name_get(self):
64         result = []
65         for record in self:
66             name = record.nombre_lab
67             result.append((record.id, name))
68         return result
69

```

Figura 20. Modelo de Laboratorio después de la actualización.

```

1 <record id="model_access_lab" model="ir.model.access">
2   <field name="name" model.access.lab/>field
3   <field name="model_id" ref="control_acceso.model_controlacceso_lab"/>
4   <field name="group_id" ref="base.group_system"/>
5   <field name="perm_read" eval="1"/>
6   <field name="perm_write" eval="1"/>
7   <field name="perm_create" eval="1"/>
8   <field name="perm_unlink" eval="1"/>
9 </record>
10 <record id="model_access_lab_admin" model="ir.model.access">
11   <field name="name" model.access.lab.admin/>field
12   <field name="model_id" ref="control_acceso.model_controlacceso_lab"/>
13   <field name="group_id" ref="control_acceso.res_groups_administrador"/>
14   <field name="perm_read" eval="1"/>
15   <field name="perm_write" eval="1"/>
16   <field name="perm_create" eval="1"/>
17   <field name="perm_unlink" eval="1"/>
18 </record>

```

Figura 21. Permisos de Laboratorio “Administrador”.

```

1 <record id="model_access_lab_docente" model="ir.model.access">
2   <field name="name" model.access.lab/>field
3   <field name="model_id" ref="control_acceso.model_controlacceso_lab"/>
4   <field name="group_id" ref="control_acceso.res_groups_docentes"/>
5   <field name="perm_read" eval="1"/>
6   <field name="perm_write" eval="0"/>
7   <field name="perm_create" eval="0"/>
8   <field name="perm_unlink" eval="0"/>
9 </record>

```

Figura 22. Permisos de Laboratorio “Docente”.

```

1 <record id="model_access_lab_estudiante" model="ir.model.access">
2   <field name="name" model.access.lab.estudiante/>field
3   <field name="model_id" ref="control_acceso.model_controlacceso_lab"/>
4   <field name="group_id" ref="control_acceso.res_groups_alumnos"/>
5   <field name="perm_read" eval="1"/>
6   <field name="perm_write" eval="0"/>
7   <field name="perm_create" eval="0"/>
8   <field name="perm_unlink" eval="0"/>
9 </record>

```

Figura 23. Permisos de Laboratorio “Estudiante”.


```

1 <?xml version="1.0" encoding="utf-8"?>
2 <odoo>
3 <data>
4 <!-- Acción de ventana Lab -->
5 <record id="view_tree_lab" model="ir.ui.view">
6 <field name="name">view.tree.lab</field>
7 <field name="model">controlacceso.lab</field>
8 <field name="arch" type="xml">
9 <tree>
10 <field name="name"/>
11 <field name="nombre_lab"/>
12 <field name="estado"/>
13 <field name="rango_ip"/></field>
14 <field name="rango_ip_habilitado"/>
15 <field name="equipos_informaticos_habilitado"/>
16 <button name="botonconfirm" type="object" string="Abrir puerta" class="oe_highlight" context="{ 'active_id': name }"/>
17 </field>
18 </tree>
19 </record>
20 <record id="view_form_lab" model="ir.ui.view">
21 <field name="name">view.form.lab</field>
22 <field name="model">controlacceso.lab</field>
23 <field name="arch" type="xml">
24 <form>
25 <sheet>
26 <group>
27 <field name="name" style="width:50%;"/>
28 </group>
29 <group>
30 <field name="nombre_lab" style="width:50%;"/>
31 </group>
32 <group>
33 <field name="rango_ip" widget="char" style="width:50%;" placeholder="Ejemplo: 192.168.0.110/24"/>
34 <field name="rango_ip_habilitado"/>
35 <field name="equipos_informaticos_habilitado"/>
36 </group>
37 </sheet>
38 </form>
39 </field>
40 </record>
41 <record id="action_view_lab" model="ir.actions.act_window">
42 <field name="name">Laboratorios</field>
43 <field name="res_model">controlacceso.lab</field>
44 <field name="view_mode">tree,form</field>
45 </record>
46 <!-- Fin Acción de ventana Lab -->
47 </data>
48 </odoo>
49

```

Figura 24. Vista de Laboratorio.

1.4 Pruebas

La

Tabla 2 muestra las pruebas unitarias a desarrollar en la primera iteración.

Tabla 2. Pruebas unitarias a ejecutar en la primera iteración

ID	Descripción	Modelo	Datos de Entrada	Salida Esperada	Resultado
PU001	Validar que un laboratorio se	Laboratorio	➤ name: LAB002	Un registro exitoso del laboratorio	Aprobada
			➤ nombre_lab: Laboratorio de Química		

	registra correctamente		<ul style="list-style-type: none"> ➤ rango_ip: 192.168.2.0/24 ➤ rango_ip_habilitado: True ➤ equipos_informaticos_habilitado: False 	LAB002 con los campos correspondientes.	
PU002	Validar que no se puede registrar un laboratorio con IP no válida	Laboratorio	<ul style="list-style-type: none"> ➤ name: LAB003 ➤ nombre_lab: Laboratorio con IP no válida ➤ rango_ip: 300.168.1.0/24 (IP no válida) 	Error de validación debido al formato incorrecto de la IP (ValidationError).	Aprobada
PU003	Validar que un laboratorio se edita correctamente	Laboratorio	<ul style="list-style-type: none"> ➤ nombre_lab: Laboratorio de Fisica Avanzada ➤ estado: abierto 	El laboratorio LAB001 cambia su nombre a "Laboratorio de Fisica Avanzada" y su estado a "abierto".	Aprobada
PU004	Validar que no se puede editar un laboratorio con IP no válida	Laboratorio	<ul style="list-style-type: none"> ➤ rango_ip: 500.168.1.0/24 (IP no válida) 	Error de validación debido a la IP no válida (ValidationError).	Aprobada
PU005	Validar que un laboratorio se puede dar de baja (eliminar) correctamente	Laboratorio	<ul style="list-style-type: none"> ➤ laboratorio.id: ID del laboratorio a eliminar 	El laboratorio se elimina correctamente, y no aparece al hacer un search.	Aprobada
PU006	Validar que se listan correctamente todos los laboratorios	Laboratorio	<ul style="list-style-type: none"> ➤ N/A 	El método de listado devuelve más de 0 laboratorios, incluyendo el laboratorio LAB001.	Aprobada

A continuación, se presentan las seis pruebas unitarias desarrolladas en la primera iteración.

```
1 class TestLaboratorio(TransactionCase):
2
3     def setUp(self):
4         super(TestLaboratorio, self).setUp()
5         # Creamos un laboratorio para las pruebas
6         self.laboratorio_data = {
7             'name': 'LAB001',
8             'nombre_lab': 'Laboratorio de Física',
9             'estado': 'cerrado',
10            'rango_ip': '192.168.1.0/24',
11            'rango_ip_habilitado': True,
12            'equipos_informaticos_habilitado': True
13        }
14        self.laboratorio = self.env['controlacceso.lab'].create(self.laboratorio_data)
15
16    def test_creacion_exitosa(self):
17        """Prueba que un laboratorio se registra correctamente"""
18        nuevo_lab = self.env['controlacceso.lab'].create({
19            'name': 'LAB002',
20            'nombre_lab': 'Laboratorio de Química',
21            'estado': 'cerrado',
22            'rango_ip': '192.168.2.0/24',
23            'rango_ip_habilitado': True,
24            'equipos_informaticos_habilitado': False
25        })
26        self.assertTrue(nuevo_lab)
27        self.assertEqual(nuevo_lab.name, 'LAB002')
28        self.assertEqual(nuevo_lab.nombre_lab, 'Laboratorio de Química')
29
30    def test_creacion_fallida(self):
31        """Prueba que un laboratorio no se registra correctamente por validación"""
32        with self.assertRaises(ValidationError):
33            # Intentar crear un laboratorio con un rango de IP no válido
34            self.env['controlacceso.lab'].create({
35                'name': 'LAB003',
36                'nombre_lab': 'Laboratorio con IP no válida',
37                'estado': 'cerrado',
38                'rango_ip': '300.168.1.0/24', # IP inválida
39                'rango_ip_habilitado': True,
40                'equipos_informaticos_habilitado': True
41            })
42
43    def test_edicion_exitosa(self):
44        """Prueba que un laboratorio se edita correctamente"""
45        self.laboratorio.write({
46            'nombre_lab': 'Laboratorio de Física Avanzada',
47            'estado': 'abierto'
48        })
49        self.assertEqual(self.laboratorio.nombre_lab, 'Laboratorio de Física Avanzada')
50        self.assertEqual(self.laboratorio.estado, 'abierto')
51
52    def test_edicion_fallida(self):
53        """Prueba que no se puede editar un laboratorio con datos no válidos"""
54        with self.assertRaises(ValidationError):
55            # Intentar editar el laboratorio con un rango de IP no válido
56            self.laboratorio.write({
57                'rango_ip': '500.168.1.0/24' # IP inválida
58            })
59
60    def test_dar_de_baja_laboratorio(self):
61        """Prueba que un laboratorio se puede dar de baja correctamente (eliminar)"""
62        # Intentar dar de baja (eliminar) el laboratorio creado en setUp
63        self.laboratorio.unlink()
64
65        # Verificar que el laboratorio ya no existe
66        laboratorio_eliminado = self.env['controlacceso.lab'].search([('id', '=', self.laboratorio.id)])
67        self.assertFalse(laboratorio_eliminado)
68
69    def test_listado_laboratorios(self):
70        """Prueba que se puede listar correctamente todos los laboratorios"""
71        laboratorios = self.env['controlacceso.lab'].search([])
72        self.assertTrue(len(laboratorios) > 0)
73        self.assertIn(self.laboratorio, laboratorios)
```

Figura 25. Clase de las pruebas unitarias para Laboratorio

```
1 def test_creacion_exitosa(self):
2     """Prueba que un laboratorio se registra correctamente"""
3     nuevo_lab = self.env['controlacceso.lab'].create({
4         'name': 'LAB002',
5         'nombre_lab': 'Laboratorio de Quimica',
6         'estado': 'cerrado',
7         'rango_ip': '192.168.2.0/24',
8         'rango_ip_habilitado': True,
9         'equipos_informaticos_habilitado': False
10    })
11    self.assertTrue(nuevo_lab)
12    self.assertEqual(nuevo_lab.name, 'LAB002')
13    self.assertEqual(nuevo_lab.nombre_lab, 'Laboratorio de Química')
```

Figura 26. Prueba unitaria 1 (PU01), el Laboratorio se crea exitosamente

```
1 def test_creacion_fallida(self):
2     """Prueba que un laboratorio no se registra correctamente por validación"""
3     with self.assertRaises(ValidationError):
4         # Intentar crear un laboratorio con un rango de IP no válido
5         self.env['controlacceso.lab'].create({
6             'name': 'LAB003',
7             'nombre_lab': 'Laboratorio con IP no válida',
8             'estado': 'cerrado',
9             'rango_ip': '300.168.1.0/24', # IP inválida
10            'rango_ip_habilitado': True,
11            'equipos_informaticos_habilitado': True
12        })
```

Figura 27. Prueba unitaria 2 (PU02), el Laboratorio no se registra por dirección IP no válida

```
1 def test_edicion_exitosa(self):
2     """Prueba que un laboratorio se edita correctamente"""
3     self.laboratorio.write({
4         'nombre_lab': 'Laboratorio de Fisica Avanzada',
5         'estado': 'abierto'
6     })
7     self.assertEqual(self.laboratorio.nombre_lab, 'Laboratorio de Fisica Avanzada')
8     self.assertEqual(self.laboratorio.estado, 'abierto')
```

Figura 28. Prueba unitaria 3 (PU03), el Laboratorio se edita exitosamente

```
1 def test_edicion_fallida(self):
2     """Prueba que no se puede editar un laboratorio con datos no válidos"""
3     with self.assertRaises(ValidationError):
4         # Intentar editar el laboratorio con un rango de IP no válido
5         self.laboratorio.write({
6             'rango_ip': '500.168.1.0/24' # IP inválida
7         })
```

Figura 29. Prueba unitaria 4 (PU04), el Laboratorio no se edita por dirección IP no válida

```
1 def test_dar_de_baja_laboratorio(self):
2     """Prueba que un laboratorio se puede dar de baja correctamente (eliminar)"""
3     # Intentar dar de baja (eliminar) el laboratorio creado en setUp
4     self.laboratorio.unlink()
5
6     # Verificar que el laboratorio ya no existe
7     laboratorio_eliminado = self.env['controlacceso.lab'].search([('id', '=', self.laboratorio.id)])
8     self.assertFalse(laboratorio_eliminado)
```

Figura 30. Prueba unitaria 5 (PU05), el Laboratorio se da de baja exitosamente

```

1 def test_listado_laboratorios(self):
2     """Prueba que se puede listar correctamente todos los laboratorios"""
3     laboratorios = self.env['controlacceso.lab'].search([])
4     self.assertTrue(len(laboratorios) > 0)
5     self.assertIn(self.laboratorio, laboratorios)

```

Figura 31. Prueba unitaria 6 (PU06), se listan todos los Laboratorios exitosamente

A continuación, se presentan los resultados obtenidos de las pruebas unitarias en la primera iteración.

```

2024-10-15 09:05:16,650 7193 INFO odoo_tests odoo.modules.loading: Module control_acceso loaded in 10.13s (incl. 0.23s test), 1613 queries (+61 test)
2024-10-15 09:05:16,650 7193 INFO odoo_tests odoo.modules.loading: 31 modules loaded in 10.86s, 1613 queries (+61 extra)
2024-10-15 09:05:17,784 7193 INFO odoo_tests odoo.modules.loading: Modules loaded.
2024-10-15 09:05:17,797 7193 INFO odoo_tests odoo.modules.registry: Registry loaded in 13.815s
2024-10-15 09:05:17,798 7193 INFO odoo_tests odoo.service.server: loading test file /opt/odoo/Integracion_Control_Acceso_Face_Recognition/control_acceso/tests/test_laboratorio.py
2024-10-15 09:05:17,808 7193 INFO odoo_tests odoo.service.server: running tests odoo.addons.control_acceso.tests.test_laboratorio.
2024-10-15 09:05:17,809 7193 INFO odoo_tests odoo.addons.control_acceso.tests.test_laboratorio: Starting TestLaboratorio.test_creacion_exitosa ...
2024-10-15 09:05:17,819 7193 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 09:05:17,823 7193 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.2.0/24
2024-10-15 09:05:17,825 7193 INFO odoo_tests odoo.addons.control_acceso.tests.test_laboratorio: Starting TestLaboratorio.test_creacion_fallida ...
2024-10-15 09:05:17,833 7193 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 09:05:17,839 7193 INFO odoo_tests odoo.addons.control_acceso.tests.test_laboratorio: Starting TestLaboratorio.test_dar_de_baja_laboratorio ...
2024-10-15 09:05:17,847 7193 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 09:05:17,861 7193 INFO odoo_tests odoo.models.unlink: User #1 deleted controlacceso.lab records with IDs: [49]
2024-10-15 09:05:17,865 7193 INFO odoo_tests odoo.addons.control_acceso.tests.test_laboratorio: Starting TestLaboratorio.test_edicion_exitosa ...
2024-10-15 09:05:17,870 7193 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 09:05:17,877 7193 INFO odoo_tests odoo.addons.control_acceso.tests.test_laboratorio: Starting TestLaboratorio.test_edicion_fallida ...
2024-10-15 09:05:17,883 7193 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 09:05:17,890 7193 INFO odoo_tests odoo.addons.control_acceso.tests.test_laboratorio: Starting TestLaboratorio.test_listado_laboratorios ...
2024-10-15 09:05:17,898 7193 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 09:05:17,999 7193 INFO odoo_tests odoo.service.server: Starting post tests
2024-10-15 09:05:18,000 7193 INFO odoo_tests odoo.service.server: 0 post-tests in 0.00s, 0 queries
2024-10-15 09:05:18,000 7193 INFO odoo_tests odoo.tests.runner: 0 failed, 0 error(s) of 12 tests when loading database 'odoo_tests'
2024-10-15 09:05:18,001 7193 INFO odoo_tests odoo.service.server: Initiating shutdown
2024-10-15 09:05:18,001 7193 INFO odoo_tests odoo.service.server: Hit CTRL-C again or send a second signal to force the shutdown.
2024-10-15 09:05:18,260 7193 INFO odoo_tests odoo.sql_db: ConnectionPool(used=0/count=67/max=64): Closed 2 connections

```

Figura 32. Ejecución de las pruebas unitarias de la segunda iteración.

2 Segunda Iteración

2.1 Planificación

La **Tabla 3** muestra las actividades específicas a desarrollar en la segunda iteración.

Tabla 3. Historias de Usuario a desarrollar en la segunda iteración.

HU #	Historia de usuario	Fecha de creación	Fecha de entrega
HU03	Obtener UUID	21/11/2023	21/11/2023
HU04	Registrar equipo informático	22/11/2023	22/11/2023
HU05	Editar equipo informático	23/11/2023	23/11/2023
HU06	Dar de baja equipo informático	24/11/2023	24/11/2023
HU07	Listar equipos informáticos	25/11/2023	25/11/2023
HU08	Generar reporte de equipos informáticos	26/11/2023	26/11/2023
HU09	Imprimir reporte de equipos informáticos	27/11/2023	27/11/2023

2.2 Diseño

La **Figura 33** muestra las funcionalidades de manera general a desarrollar en la segunda iteración.

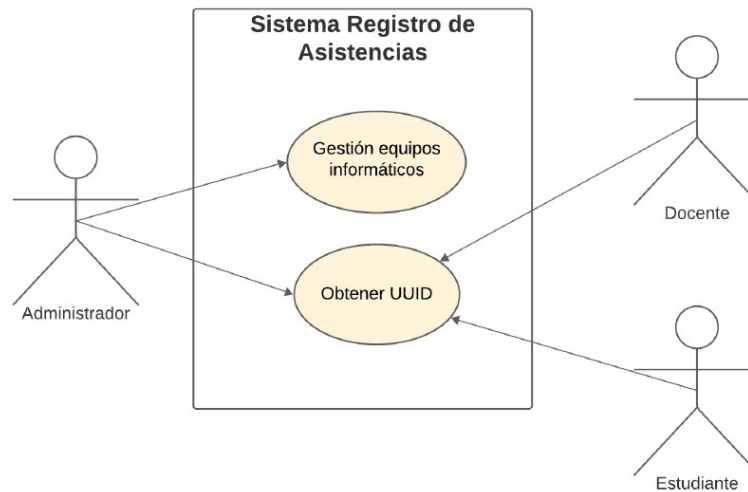


Figura 33. Casos de Uso general para la segunda iteración.

2.2.1 Diagramas BPMN

A continuación, se presentan los diagramas BPMN pertenecientes a cada Historia de Usuario a desarrollar en la segunda iteración.

2.2.1.1 HU03: Obtener UUID

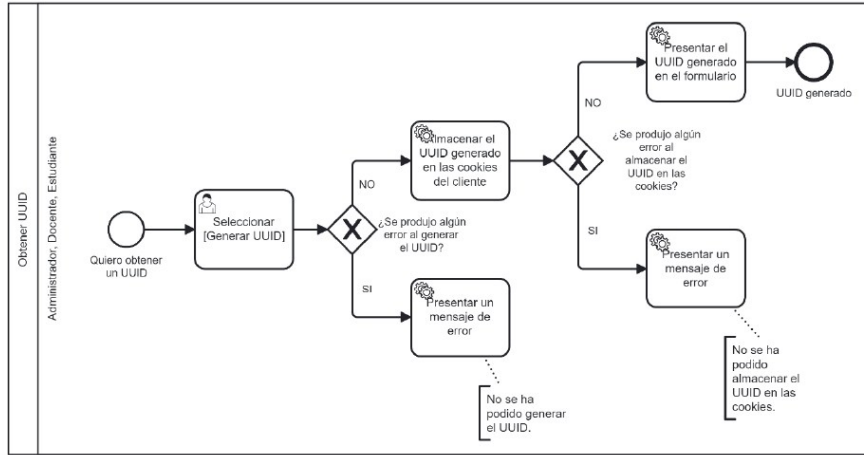


Figura 34. Diagrama BPMN de HU03.

2.2.1.2 HU04: Registrar equipo informático

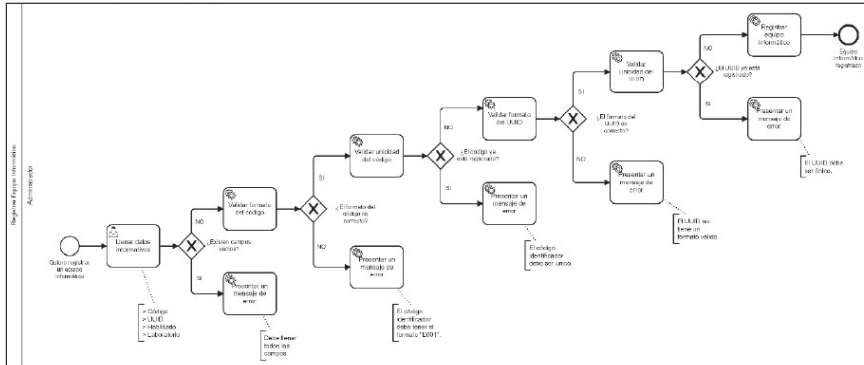


Figura 35. Diagrama BPMN de HU04.

2.2.1.3 HU05: Editar equipo informático

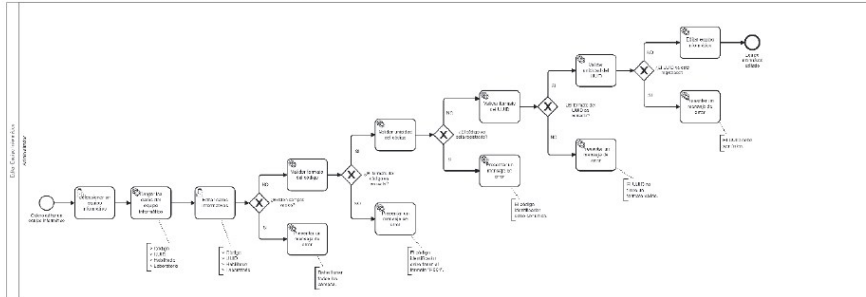


Figura 36. Diagrama BPMN de HU05.

2.2.1.4 HU06: Dar de baja equipo informático

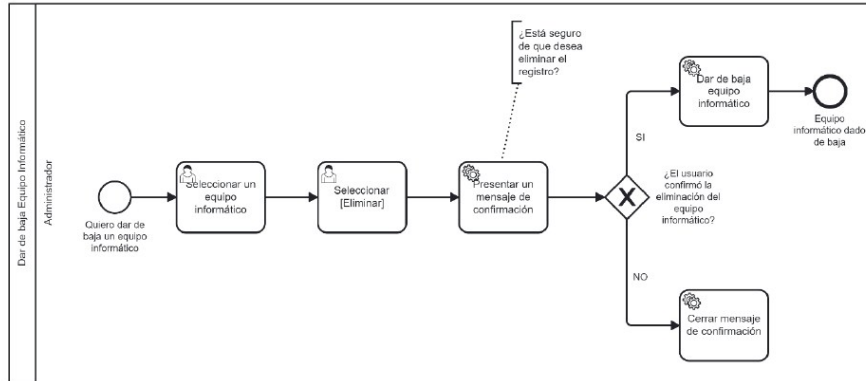


Figura 37. Diagrama BPMN de HU06.

2.2.1.5 HU07: Listar equipo informático

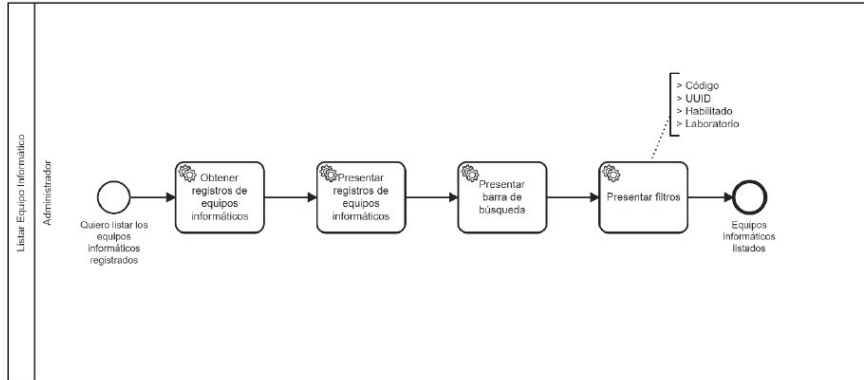


Figura 38. Diagrama BPMN de HU07.

2.2.1.6 HU08: Generar reporte de equipo informático y HU09: Imprimir reporte de equipo informático

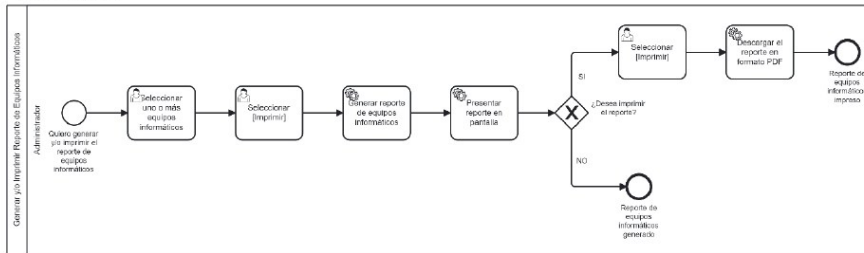


Figura 39. Diagrama BPMN de HU08 y HU09.

2.2.2 Prototipos

A continuación, se presentan los prototipos pertenecientes a cada Historia de Usuario a desarrollar en la segunda iteración.

2.2.2.1 HU03: Obtener UUID

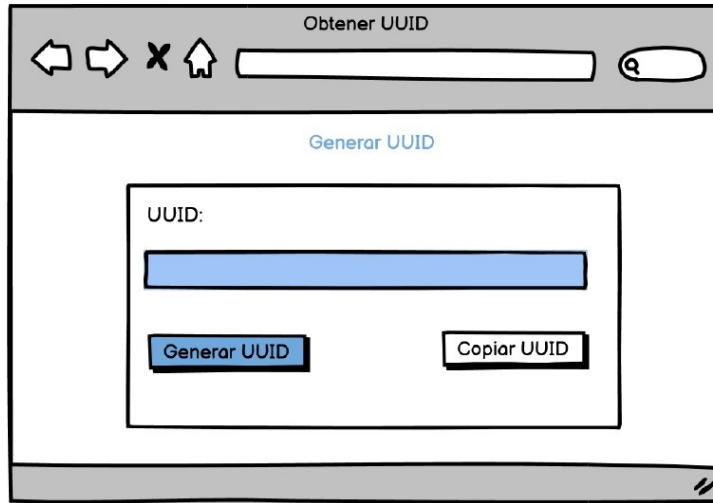


Figura 40. Prototipo para Registrar Equipo Informático.

2.2.2.2 HU04: Registrar equipo informático

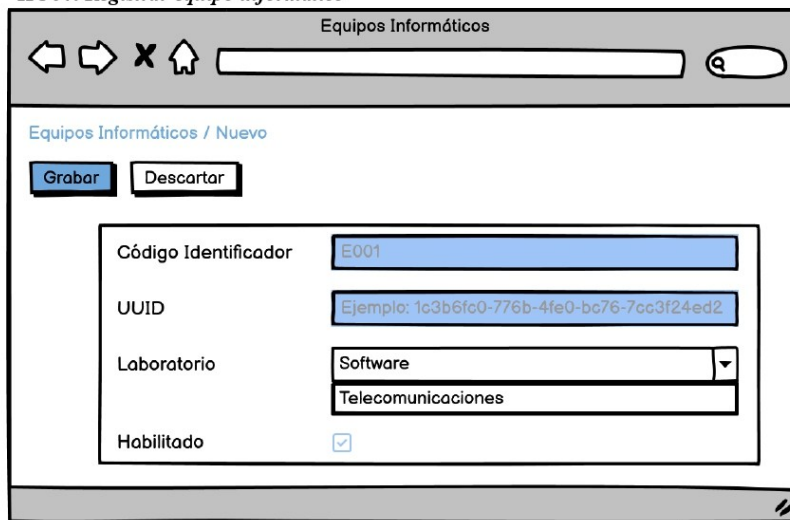
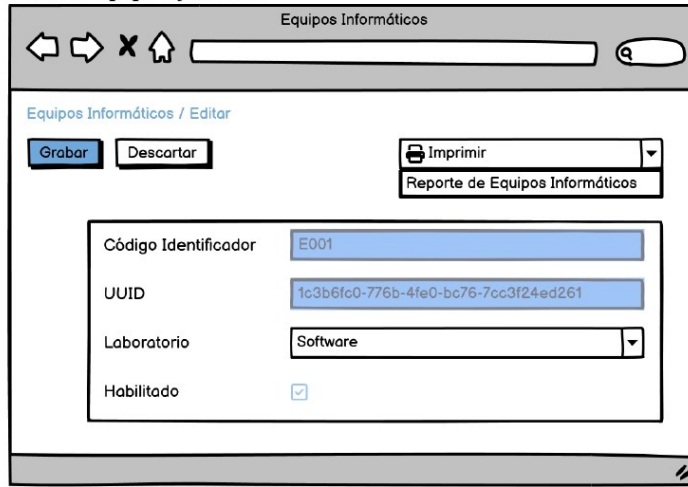


Figura 41. Prototipo para Registrar Equipo Informático.

2.2.2.3 HU05: Editar equipo informático



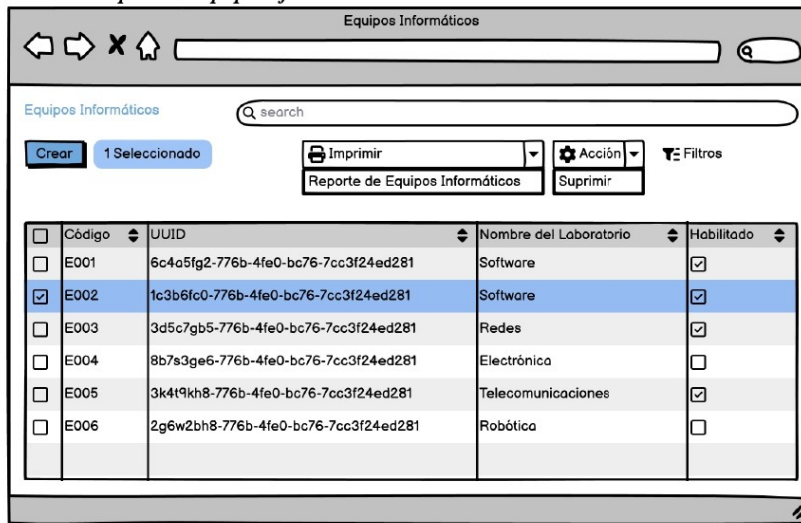
Equipos Informáticos

Equipos Informáticos / Editar

Código Identificador:
 UUID:
 Laboratorio:
 Habilitado:

Figura 42. Prototipo para Editar Equipo Informático.

2.2.2.4 HU06: Dar de baja equipo informático - HU07: Listar equipo informático - HU08: Generar reporte de equipo informático



Equipos Informáticos

Equipos Informáticos

1 Seleccionado

<input type="checkbox"/>	Código	UUID	Nombre del Laboratorio	Habilitado
<input type="checkbox"/>	E001	6c4a5fg2-776b-4fe0-bc76-7cc3f24ed281	Software	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	E002	1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281	Software	<input checked="" type="checkbox"/>
<input type="checkbox"/>	E003	3d5c7gb5-776b-4fe0-bc76-7cc3f24ed281	Redes	<input checked="" type="checkbox"/>
<input type="checkbox"/>	E004	8b7s3ge6-776b-4fe0-bc76-7cc3f24ed281	Electrónica	<input type="checkbox"/>
<input type="checkbox"/>	E005	3k4t9kh8-776b-4fe0-bc76-7cc3f24ed281	Telecomunicaciones	<input checked="" type="checkbox"/>
<input type="checkbox"/>	E006	2g6w2bh8-776b-4fe0-bc76-7cc3f24ed281	Robótica	<input type="checkbox"/>

Figura 43. Prototipo para Dar de baja, Listar y Generar Reporte de Equipo Informático.

2.2.2.5 HU09: Imprimir reporte de equipo informático

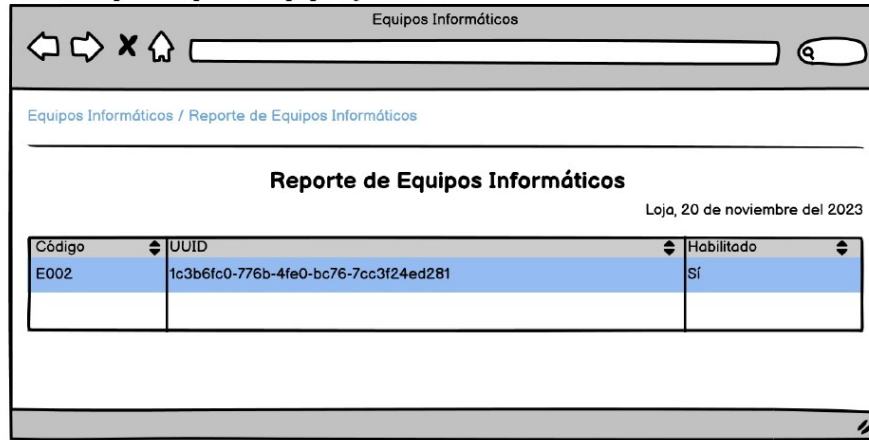


Figura 44. Prototipo para Imprimir Reporte de Equipo Informático.

2.3 Codificación

A continuación, se presenta los fragmentos principales del código fuente desarrollados en la segunda iteración.

```

1 from odoo import models, fields, api
2 from odoo.exceptions import ValidationError
3 import re
4 import uuid
5
6 class EquipoInformatico(models.Model):
7     _name = 'controlacceso.equipoinformatico'
8     _description = 'Equipos Informáticos'
9
10    codigo_identificador = fields.Char(string='Código Identificador', required=True, index=True, copy=False, readonly=True, default='Nuevo')
11    uuid = fields.Char(string='UUID', required=True)
12    habilitado = fields.Boolean(string='Habilitado', default=True)
13    laboratorio_id = fields.Many2one('controlacceso.lab', string='Laboratorio', ondelete='cascade', required=True)
14
15    nombre_laboratorio = fields.Char(related='laboratorio_id.nombre_lab', string='Nombre del Laboratorio', store=True)
16
17    # Campo adicional para notas o descripción
18    descripcion = fields.Text(string='Descripción', help='Notas o comentarios adicionales sobre el equipo informático.')
19
20    @api.constrains('codigo_identificador')
21    def validar_codigo_identificador(self):
22        for equipo in self:
23            # Verificar el formato del código identificador (ejemplo: E001)
24            if not re.match(r'^[A-Za-z]{d(3)}$', equipo.codigo_identificador):
25                raise ValidationError('El código identificador debe tener el formato "E001".')
26
27            # Verificar la unicidad del código identificador
28            if self.search([('codigo_identificador', '=', equipo.codigo_identificador), ('id', '!=', equipo.id)]):
29                raise ValidationError('El código identificador debe ser único.')
30
31    @api.model
32    def create(self, vals):
33        # Generar el código identificador incremental
34        if vals.get('codigo_identificador', 'Nuevo') == 'Nuevo':
35            # Usar una secuencia para generar el código incremental
36            seq = self.env['ir.sequence'].next_by_code('controlacceso.equipoinformatico') or '/'
37            vals['codigo_identificador'] = f'E{seq.zfill(3)}' # Formato E001, E002, etc.
38
39        return super(EquipoInformatico, self).create(vals)
40
41    @api.constrains('uuid')
42    def validar_uuid(self):
43        for equipo in self:
44            try:
45                # Verificar si el formato del UUID es válido
46                uuid_obj = uuid.UUID(equipo.uuid, version=4) # Solo acepta UUID versión 4
47            except ValueError:
48                raise ValidationError('El UUID no tiene un formato válido.')
49
50            # Verificar si el UUID ya existe en otros equipos
51            if self.search([('uuid', '=', equipo.uuid), ('id', '!=', equipo.id)]):
52                raise ValidationError('El UUID debe ser único. Por favor, vuelva a generar un nuevo UUID.')

```

Figura 45. Modelo de Equipo Informático.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <root>
3    <data>
4      <!-- Vista del formulario de equipos informáticos -->
5      <record id="view_form_equipoinformatico" model="in.ui.view">
6        <field name="name" equipoinformatico.form />
7        <field name="model" controlacceso.equipoinformatico />
8        <field name="arch" type="xml">
9          <form>
10             <sheet>
11               <group>
12                 <field name="codigo_identificador" widget="char" style="width:50%; placeholder="Ejemplo: E001" invisible="1"/>
13                 <field name="uid" widget="char" style="width:50%; placeholder="Ejemplo: e08ac07-6ac3-4a48-8065-ccca2f9ae5b3"/>
14                 <field name="descripcion" widget="char" placeholder="Notas o comentarios adicionales sobre el equipo informático."/>
15                 <field name="laboratorio_id"/>
16                 <field name="habilitado"/>
17               </group>
18             </sheet>
19           </form>
20         </field>
21       </record>
22
23      <!-- Vista del árbol de equipos informáticos -->
24      <record id="view_tree_equipoinformatico" model="in.ui.view">
25        <field name="name" equipoinformatico.tree />
26        <field name="model" controlacceso.equipoinformatico />
27        <field name="arch" type="xml">
28          <tree>
29            <field name="codigo_identificador"/>
30            <field name="uid"/>
31            <field name="descripcion"/>
32            <field name="nombre_laboratorio"/>
33            <field name="habilitado"/>
34          </tree>
35        </field>
36      </record>
37
38      <!-- Vista de búsqueda de equipos informáticos -->
39      <record id="view_search_equipoinformatico" model="in.ui.view">
40        <field name="name" equipoinformatico.search />
41        <field name="model" controlacceso.equipoinformatico />
42        <field name="arch" type="xml">
43          <search>
44            <field name="codigo_identificador" string="Código Identificador"/>
45            <field name="uid" string="UID"/>
46            <field name="descripcion" string="Descripción"/>
47            <field name="nombre_laboratorio" string="Laboratorio"/>
48            <field name="habilitado" string="Habilitado"/>
49          </search>
50        </field>
51      </record>
52
53      <!-- Acción para abrir la vista de equipos informáticos -->
54      <record id="action_view_equipoinformatico" model="in.actions.act_window">
55        <field name="name" Equipos Informáticos />
56        <field name="res_model" controlacceso.equipoinformatico />
57        <field name="view_mode" tree,form />
58        <field name="view_id" ref="view_tree_equipoinformatico"/>
59        <field name="search_view_id" ref="view_search_equipoinformatico"/>
60        <field name="help" type="html">
61          <p class="oe_view_nocontent_create">
62            Click para crear un nuevo equipo informático.
63          </p>
64        </field>
65      </record>
66    </data>
67  </root>
68

```

Figura 46. Vista de Equipo Informático.

```

1 <odoo>
2
3 <data>
4 <record id="equipoinformatico.reporte" model="ir.actions.report">
5 <field name="name">Reporte de Equipos Informáticos</field>
6 <field name="model">controlacceso.equipoinformatico</field>
7 <field name="report_type">web-html</field>
8 <field name="report_name">control_acceso.report_equipoinformatico</field>
9 <field name="report_file">control_acceso.report_equipoinformatico</field>
10 <field name="binding_model_id" ref="model_controlacceso_equipoinformatico"/>
11 <field name="binding_type">report</field>
12 </record>
13
14 <template id="report_equipoinformatico">
15 <t t-call="web.html_container">
16 <t t-call="web.external_layout">
17 <center>
18 <h1>Reporte de Equipos Informáticos</h1>
19 </center>
20 <p class="text-right">Loja, <span t-esc="context_timestamp(datetime.datetime.now()).strftime('%d de %B del %Y')"/></p>
21 <div class="page">
22 <div style="font-family: 'arial', font-size: 16pt; color: #000000">
23 <xpath expr="." position="inside">
24 <style>
25 table, td, th {
26 border: 1px solid;
27 }
28
29 table {
30 width: 100%;
31 border-collapse: collapse;
32 }
33 </style>
34 </xpath>
35 <table class="table table-striped" id="tablets">
36 <thead>
37 <tr>
38 <th>Codigo Identificador</th>
39 <th>UUID</th>
40 <th>Habilitado</th>
41 </tr>
42 </thead>
43 <tr t-foreach="docs" t-as="equipo">
44 <td t-esc="equipo.codigo_identificador"></td>
45 <td t-esc="equipo.uuid"></td>
46 <td t-if="equipo.habilitado">Si</td><td t-else="">No</td></tr>
47 </tr>
48 </table>
49 </div>
50 </div>
51 </t>
52 </template>
53
54 <record id="paperformat_frenchcheck" model="report.paperformat">
55 <field name="name">French Bank Check</field>
56 <field name="default" eval="True"/>
57 <field name="format">custom</field>
58 <field name="page_height">80</field>
59 <field name="page_width">175</field>
60 <field name="orientation">Portrait</field>
61 <field name="margin_top">3</field>
62 <field name="margin_bottom">3</field>
63 <field name="margin_left">3</field>
64 <field name="margin_right">3</field>
65 <field name="header_line" eval="False"/>
66 <field name="header_spacing">3</field>
67 <field name="dpi">80</field>
68 </record>
69 </data>
70 </odoo>
71

```

Figura 47. Vista del Reporte de Equipo Informático.


```

1 <!-- EQUIPO INFORMATICO-->
2 <record id="model_access equipoinformatico_admin" model="ir.model.access">
3   <field name="name">model.access.equipoinformatico.admin</field>
4   <field name="model_id" ref="control_acceso.model_controlacceso_equipoinformatico"/>
5   <field name="group_id" ref="control_acceso.res_groups_administrador"/>
6   <field name="perm_read" eval="1"/>
7   <field name="perm_write" eval="1"/>
8   <field name="perm_create" eval="1"/>
9   <field name="perm_unlink" eval="1"/>
10 </record>
11 <record id="model_acces_view_equipoinformatico_admin" model="ir.rule">
12   <field name="name">model.acces.view.equipoinformatico.admin</field>
13   <field name="model_id" ref="control_acceso.model_controlacceso_equipoinformatico"/>
14   <field name="groups" eval="[ (4, ref('&apos;control_acceso.res_groups_administrador&apos;))]"/>
15   <field name="perm_read" eval="1"/>
16   <field name="perm_write" eval="1"/>
17   <field name="perm_create" eval="1"/>
18   <field name="perm_unlink" eval="1"/>
19   <field name="domain_force">[(1, &apos;=&apos;, 1)]</field>
20 </record>
21 <!-- EQUIPO INFORMATICO-->

```

Figura 48. Permisos de Equipo Informático “Administrador”.

2.4 Pruebas

La **Tabla 4** muestra las pruebas unitarias a desarrollar en la primera iteración.

Tabla 4. Pruebas unitarias a ejecutar en la segunda iteración

ID	Descripción	Modelo	Datos de Entrada	Salida Esperada	Resultado
PU007	Validar que un equipo informático se registre correctamente	Equipo Informático	<ul style="list-style-type: none"> ➤ codigo_identificador: 'E001' ➤ uuid: '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281' ➤ laboratorio_id: ID del laboratorio ➤ habilitado: True 	El equipo se registra correctamente con el código 'E001' y UUID '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281'.	Aprobada
PU008	Validar que un equipo informático no se registre con código inválido	Equipo Informático	<ul style="list-style-type: none"> ➤ codigo_identificador: '001' ➤ uuid: '7ffcbe04-a18b-419a-a0cb-cbbd32154f47' ➤ laboratorio_id: ID del laboratorio ➤ habilitado: True 	Error de validación por formato incorrecto del código identificador ('001' no sigue el patrón 'E001').	Aprobada
PU009	Validar que un equipo informático se edite correctamente	Equipo Informático	<ul style="list-style-type: none"> ➤ codigo_identificador: 'E002' ➤ uuid: '598da705-e6bd-4eb3-b44b-af3952a62736' 	El equipo se edita correctamente, cambiando el código identificador a 'E002' y el UUID a '598da705-e6bd-	Aprobada

				4eb3-b44b-af3952a62736'.	
PU010	Validar que un equipo informático no se edite con UUID inválido	Equipo Informático	➤ uuid: 'ZZ:ZZ:ZZ:ZZ'	Error de validación por formato incorrecto del UUID ('ZZ:ZZ:ZZ:ZZ' no sigue el patrón de un UUID válido).	Aprobada
PU011	Validar que un equipo informático se pueda dar de baja	Equipo Informático	➤ equipo.id: ID del equipo a eliminar	El equipo se elimina correctamente y ya no aparece en el sistema.	Aprobada
PU012	Validar que se puedan listar los equipos informáticos	Equipo Informático	➤ N/A	Todos los equipos registrados son listados correctamente, incluyendo el equipo con código 'E001'.	Aprobada

A continuación, se presentan las seis pruebas unitarias desarrolladas en la segunda iteración.

```

1 from odoo.tests.common import TransactionCase
2 from odoo.exceptions import ValidationError
3
4 class TestEquipoInformatico(TransactionCase):
5
6     def setUp(self):
7         super(TestEquipoInformatico, self).setUp()
8         # Crear un laboratorio para asociar a los equipos informaticos
9         self.laboratorio_data = {
10             'name': 'LAB001',
11             'nombre_lab': 'Laboratorio de Fisica',
12             'estado': 'cerrado',
13             'rango_ip': '192.168.1.0/24',
14             'rango_ip_habilitado': True,
15             'equipos_informaticos_habilitado': True
16         }
17         self.laboratorio = self.env['controlacceso.lab'].create(self.laboratorio_data)
18
19         # Datos del equipo informatico
20         self.equipo_data = {
21             'codigo_identificador': 'E001',
22             'direccion_mac': 'AA:BB:CC:DD:EE:FF',
23             'laboratorio_id': self.laboratorio.id,
24             'habilitado': True
25         }
26
27     def test_creacion_exitosa(self):
28         """Prueba que un equipo informático se registra correctamente"""
29         nuevo_equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
30         self.assertTrue(nuevo_equipo)
31         self.assertEqual(nuevo_equipo.codigo_identificador, 'E001')
32         self.assertEqual(nuevo_equipo.direccion_mac, 'AA:BB:CC:DD:EE:FF')
33
34     def test_creacion_fallida(self):
35         """Prueba que un equipo informático no se registra correctamente por validación"""
36         with self.assertRaises(ValidationError):
37             # Intentar crear un equipo con un formato de código identificador no válido
38             self.env['controlacceso.equipoinformatico'].create({
39                 'codigo_identificador': '001', # Código identificador inválido
40                 'direccion_mac': 'AA:BB:CC:DD:EE:FF',
41                 'laboratorio_id': self.laboratorio.id,
42                 'habilitado': True
43             })
44
45     def test_edicion_exitosa(self):
46         """Prueba que un equipo informático se edita correctamente"""
47         equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
48         equipo.write({
49             'codigo_identificador': 'E002',
50             'direccion_mac': 'FF:EE:DD:CC:BB:AA',
51         })
52         self.assertEqual(equipo.codigo_identificador, 'E002')
53         self.assertEqual(equipo.direccion_mac, 'FF:EE:DD:CC:BB:AA')
54
55     def test_edicion_fallida(self):
56         """Prueba que no se puede editar un equipo informático con datos no válidos"""
57         equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
58         with self.assertRaises(ValidationError):
59             # Intentar editar el equipo con un formato de dirección MAC no válido
60             equipo.write({
61                 'direccion_mac': 'ZZ:ZZ:ZZ:ZZ:ZZ:ZZ' # Dirección MAC inválida
62             })
63
64     def test_dar_de_baja_equipo(self):
65         """Prueba que un equipo informático se puede dar de baja correctamente (eliminar)"""
66         equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
67         equipo.unlink()
68
69         # Verificar que el equipo ya no existe
70         equipo_eliminado = self.env['controlacceso.equipoinformatico'].search([('id', '=', equipo.id)])
71         self.assertFalse(equipo_eliminado)
72
73     def test_listado_equipos(self):
74         """Prueba que se puede listar correctamente todos los equipos informaticos"""
75         equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
76         equipos = self.env['controlacceso.equipoinformatico'].search([])
77         self.assertTrue(len(equipos) > 0)
78         self.assertIn(equipo, equipos)
79

```

Figura 49. Clase de las pruebas unitarias para Equipo Informático

```
1 def test_creacion_exitosa(self):
2     """Prueba que un equipo informático se registra correctamente"""
3     nuevo_equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
4     self.assertTrue(nuevo_equipo)
5     self.assertEqual(nuevo_equipo.codigo_identificador, 'E001')
6     self.assertEqual(nuevo_equipo.direccion_mac, 'AA:BB:CC:DD:EE:FF')
```

Figura 50. Prueba unitaria 7 (PU07), el Equipo Informático se crea exitosamente

```
1 def test_creacion_fallida(self):
2     """Prueba que un equipo informático no se registra correctamente por validación"""
3     with self.assertRaises(ValidationError):
4         # Intentar crear un equipo con un formato de código identificador no válido
5         self.env['controlacceso.equipoinformatico'].create({
6             'codigo_identificador': '001', # Código identificador inválido
7             'direccion_mac': 'AA:BB:CC:DD:EE:FF',
8             'laboratorio_id': self.laboratorio.id,
9             'habilitado': True
10        })
```

Figura 51. Prueba unitaria 8 (PU08), el Equipo Informático no se registra por código identificador no válido

```
1 def test_edicion_exitosa(self):
2     """Prueba que un equipo informático se edita correctamente"""
3     equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
4     equipo.write({
5         'codigo_identificador': 'E002',
6         'direccion_mac': 'FF:EE:DD:CC:BB:AA',
7     })
8     self.assertEqual(equipo.codigo_identificador, 'E002')
9     self.assertEqual(equipo.direccion_mac, 'FF:EE:DD:CC:BB:AA')
```

Figura 52. Prueba unitaria 9 (PU09), el Equipo Informático se edita exitosamente

```
1 def test_edicion_fallida(self):
2     """Prueba que no se puede editar un equipo informático con datos no válidos"""
3     equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
4     with self.assertRaises(ValidationError):
5         # Intentar editar el equipo con un formato de dirección MAC no válido
6         equipo.write({
7             'direccion_mac': 'ZZ:ZZ:ZZ:ZZ:ZZ:ZZ' # Dirección MAC inválida
8         })
```

Figura 53. Prueba unitaria 10 (PU10), el Equipo Informático no se edita por dirección MAC no válida

```
1 def test_dar_de_baja_equipo(self):
2     """Prueba que un equipo informático se puede dar de baja correctamente (eliminar)"""
3     equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
4     equipo.unlink()
5
6     # Verificar que el equipo ya no existe
7     equipo_eliminado = self.env['controlacceso.equipoinformatico'].search([('id', '=', equipo.id)])
8     self.assertFalse(equipo_eliminado)
```

Figura 54. Prueba unitaria 11 (PU11), el Equipo Informático se da de baja exitosamente

```
1 def test_listadoEquipos(self):
2     """Prueba que se puede listar correctamente todos los equipos informáticos"""
3     equipo = self.env['controlacceso.equipoinformatico'].create(self.equipo_data)
4     equipos = self.env['controlacceso.equipoinformatico'].search([])
5     self.assertTrue(len(equipos) > 0)
6     self.assertIn(equipo, equipos)
```

Figura 55. Prueba unitaria 12 (PU12), se listan todos los Equipos Informáticos exitosamente

A continuación, se presentan los resultados obtenidos de las pruebas unitarias en la segunda iteración.

```

2024-10-15 08:58:06,140 6857 INFO odoo_tests odoo.modules.loading: loading control_acceso/data/cron.xml
2024-10-15 08:58:06,178 6857 INFO odoo_tests odoo.modules.loading: Module control_acceso: loading demo
2024-10-15 08:58:06,354 6857 INFO odoo_tests odoo.http: HTTP Configuring static files
2024-10-15 08:58:06,366 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_creacion_exitosa ...
2024-10-15 08:58:06,384 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:06,396 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_creacion_fallida ...
2024-10-15 08:58:06,402 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:06,409 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_dar_de_baja_equipo ...
2024-10-15 08:58:06,416 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:06,442 6857 INFO odoo_tests odoo.models.unlink: User #1 deleted control_acceso.equipo_informatico records with IDs: [15]
2024-10-15 08:58:06,446 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_edicion_exitosa ...
2024-10-15 08:58:06,453 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:06,478 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_edicion_fallida ...
2024-10-15 08:58:06,476 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:06,489 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_listadoequipos ...
2024-10-15 08:58:06,497 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:06,668 6857 INFO odoo_tests odoo.modules.loading: Module control_acceso loaded in 10.23s (incl. 0.24s test), 1613 quer
ies (+86 test)
2024-10-15 08:58:06,668 6857 INFO odoo_tests odoo.modules.loading: 31 modules loaded in 11.00s, 1613 queries (+86 extra)
2024-10-15 08:58:07,734 6857 INFO odoo_tests odoo.modules.loading: Modules loaded.
2024-10-15 08:58:07,744 6857 INFO odoo_tests odoo.modules.registry: Registry loaded in 14.307s
2024-10-15 08:58:07,744 6857 INFO odoo_tests odoo.service.server: loading test file /opt/odoo/Integracion_Control_Acceso_Face_Recogniti
on/control_acceso/tests/test_equipo_informatico.py
2024-10-15 08:58:07,755 6857 INFO odoo_tests odoo.service.server: running tests odoo.addons.control_acceso.tests.test_equipo_informatic
o
2024-10-15 08:58:07,756 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_creacion_exitosa ...
2024-10-15 08:58:07,765 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24

```

Figura 56. Ejecución de las pruebas unitarias de la segunda iteración, parte 1.

```

ies (+86 test)
2024-10-15 08:58:06,668 6857 INFO odoo_tests odoo.modules.loading: 31 modules loaded in 11.00s, 1613 queries (+86 extra)
2024-10-15 08:58:07,734 6857 INFO odoo_tests odoo.modules.loading: Modules loaded.
2024-10-15 08:58:07,744 6857 INFO odoo_tests odoo.modules.registry: Registry loaded in 14.307s
2024-10-15 08:58:07,744 6857 INFO odoo_tests odoo.service.server: loading test file /opt/odoo/Integracion_Control_Acceso_Face_Recogniti
on/control_acceso/tests/test_equipo_informatico.py
2024-10-15 08:58:07,755 6857 INFO odoo_tests odoo.service.server: running tests odoo.addons.control_acceso.tests.test_equipo_informatic
o
2024-10-15 08:58:07,756 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_creacion_exitosa ...
2024-10-15 08:58:07,765 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:07,776 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_creacion_fallida ...
2024-10-15 08:58:07,784 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:07,799 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_dar_de_baja_equipo ...
2024-10-15 08:58:07,824 6857 INFO odoo_tests odoo.models.unlink: User #1 deleted control_acceso.equipo_informatico records with IDs: [21]
2024-10-15 08:58:07,828 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_edicion_exitosa ...
2024-10-15 08:58:07,836 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:07,859 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_edicion_fallida ...
2024-10-15 08:58:07,863 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:07,877 6857 INFO odoo_tests odoo.addons.control_acceso.tests.test_equipo_informatico: Starting TestEquipoInformatico.t
est_listadoequipos ...
2024-10-15 08:58:07,885 6857 INFO odoo_tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-15 08:58:08,006 6857 INFO odoo_tests odoo.service.server: Starting post tests
2024-10-15 08:58:08,008 6857 INFO odoo_tests odoo.service.server: 0 post-tests in 0.00s, 0 queries
2024-10-15 08:58:08,008 6857 INFO odoo_tests odoo.tests.runner: 0 failed, 0 error(s) or 12 tests when loading database 'odoo_tests'
2024-10-15 08:58:08,009 6857 INFO odoo_tests odoo.service.server: Initiating shutdown
2024-10-15 08:58:08,009 6857 INFO odoo_tests odoo.service.server: Hit CTRL-C again or send a second signal to force the shutdown.
2024-10-15 08:58:08,243 6857 INFO odoo_tests odoo.sql_db: ConnectionPool(used=0/count=8/max=64): Closed 2 connections

```

Figura 57. Ejecución de las pruebas unitarias de la segunda iteración, parte 2.

3 Tercera Iteración

3.1 Planificación

La **Tabla 5** muestra las actividades específicas a desarrollar en la tercera iteración.

Tabla 5. Historias de Usuario a desarrollar en la tercera iteración.

HU #	Historia de usuario	Fecha de creación	Fecha de entrega
HU10	Registrar usuario e imagen de usuario	28/11/2023	08/12/2023
HU11	Editar usuario e imagen de usuario	09/12/2023	10/12/2023
HU12	Dar de baja usuario e imagen de usuario	11/12/2023	11/12/2023
HU13	Listar usuarios	12/12/2023	12/12/2023

3.2 Diseño

La **Figura 58** muestra las funcionalidades de manera general a desarrollar en la tercera iteración.

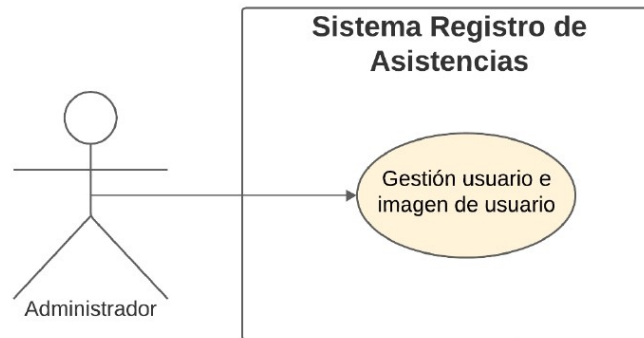


Figura 58. Casos de Uso general para la tercera iteración.

3.2.1 Diagramas BPMN

A continuación, se presentan los diagramas BPMN pertenecientes a cada Historia de Usuario a desarrollar en la tercera iteración.

3.2.1.1 HU10: Registrar usuario e imagen de usuario

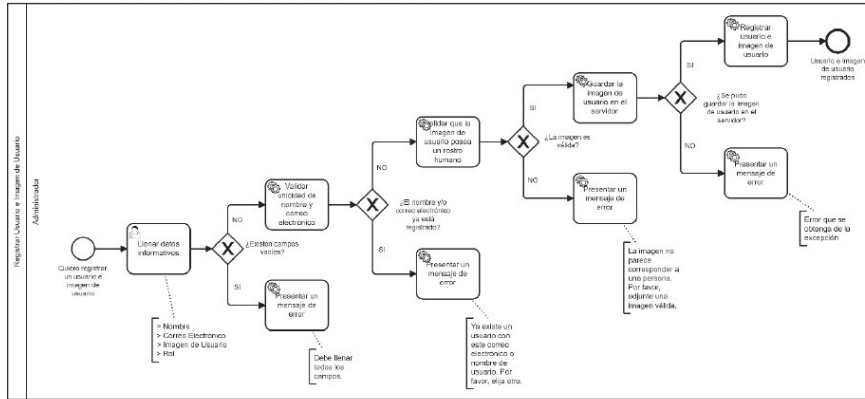


Figura 59. Diagrama BPMN de HU10.

3.2.1.2 HU11: Editar usuario e imagen de usuario

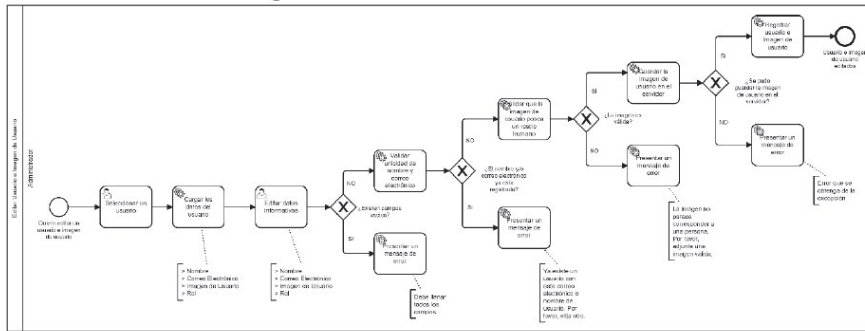


Figura 60. Diagrama BPMN de HU11.

3.2.1.3 HU12: Dar de baja usuario e imagen de usuario

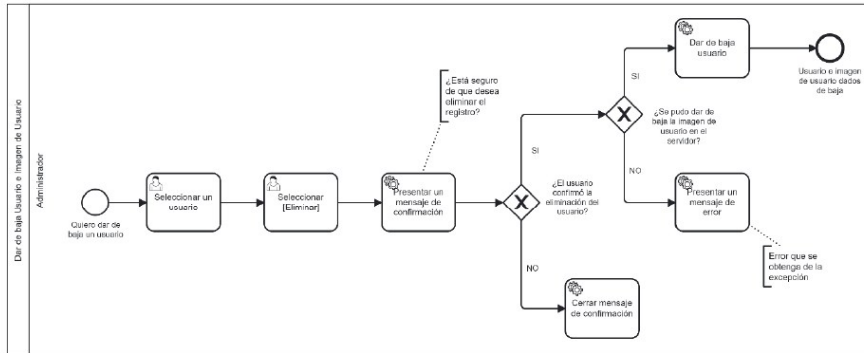


Figura 61. Diagrama BPMN de HU12.

3.2.1.4 HU13: Listar usuarios

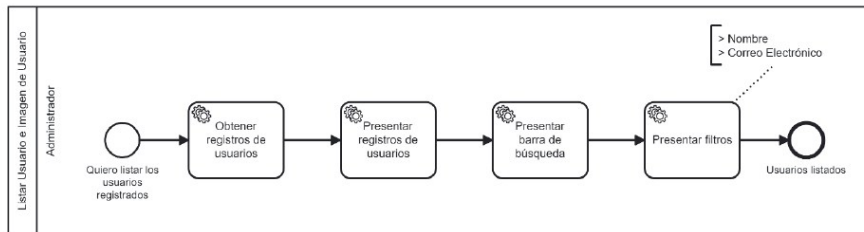


Figura 62. Diagrama BPMN de HU13.

3.2.2 Prototipos

A continuación, se presentan los prototipos pertenecientes a cada Historia de Usuario a desarrollar en la tercera iteración.

3.2.2.1 HU09: Registrar usuario e imagen de usuario

The screenshot shows a web browser window titled 'Usuarios'. The address bar contains a search icon and a search input field. Below the browser window, the page header reads 'Usuarios / Nuevo'. There are two buttons: 'Grabar' (highlighted in blue) and 'Descartar'. The main form area contains three input fields: 'Nombre' (empty), 'Usuario' (empty), and 'Permisos'. The 'Permisos' section has three checkboxes: 'Administrador' (checked), 'Docentes' (unchecked), and 'Alumnos' (unchecked). To the right of the input fields is a placeholder image labeled 'Imagen' showing a simple black outline of a person's head and shoulders.

Figura 63. Prototipo para Registrar Usuario e Imagen de Usuario.

3.2.2.2 HU10: Editar usuario e imagen de usuario

The screenshot shows a web browser window titled 'Usuarios'. The address bar contains a search icon and a search input field. Below the browser window, the page header reads 'Usuarios / Editor'. There are two buttons: 'Grabar' (highlighted in blue) and 'Descartar'. The main form area contains three input fields: 'Nombre' (containing 'Luis Jumbo'), 'Usuario' (containing 'luis.a.jumbo.h@uni.edu.ec'), and 'Permisos'. The 'Permisos' section has three checkboxes: 'Administrador' (checked), 'Docentes' (unchecked), and 'Alumnos' (unchecked). To the right of the input fields is a placeholder image labeled 'Imagen' showing a photograph of a man with dark hair wearing a dark shirt.

Figura 64. Prototipo para Editar Usuario e Imagen de Usuario.

3.2.2.3 HU11: Dar de baja usuario e imagen de usuario - HU12: Listar usuarios

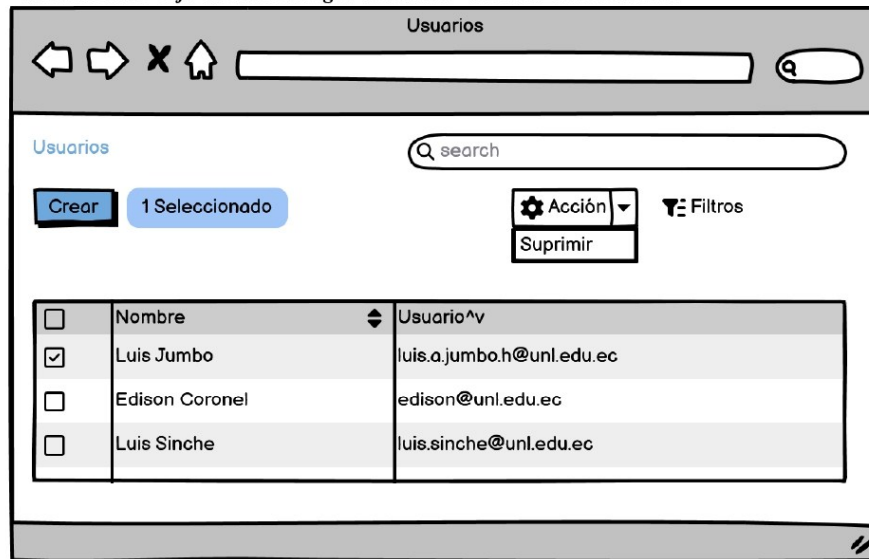


Figura 65. Prototipo para Dar de baja, Listar y Listar Usuario e Imagen de Usuario.

3.3 Codificación

A continuación, se presenta los fragmentos principales del código fuente desarrollados en la tercera iteración.


```

1 <record id="model_access_res_user_Admin" model="ir.model.access">
2   <field name="name">model.access.res.user/</field>
3   <field name="model_id" ref="base.model_res_users"/>
4   <field name="group_id" ref="control_acceso.res_groups_administrador"/><!--Administracion/Ajustes-->
5   <field name="perm_create" eval="1"/>
6   <field name="perm_write" eval="1"/>
7   <field name="perm_unlink" eval="1"/>
8   <field name="perm_read" eval="1"/>
9 </record>

```

Figura 67. Permisos de Usuarios “Administrador”.

3.4 Pruebas

La **Tabla 6** muestra las pruebas unitarias a desarrollar en la tercera iteración.

Tabla 6. Pruebas unitarias a ejecutar en la tercera iteración

ID	Descripción	Modelo	Datos de Entrada	Salida Esperada	Resultado
PU013	Validar que un usuario se registre correctamente	Usuario	<ul style="list-style-type: none"> ➤ name: 'usuario_test' ➤ login: 'test@example.com' ➤ image_1920: b64 	El usuario se registra correctamente con el nombre 'usuario_test'.	Aprobada
PU014	Validar que un usuario no se registre sin imagen válida	Usuario	<ul style="list-style-type: none"> ➤ name: 'usuario_test' ➤ login: 'test@example.com' ➤ image_1920: '' 	Error de validación por falta de imagen válida.	Aprobada
PU015	Validar que un usuario se edite correctamente	Usuario	<ul style="list-style-type: none"> ➤ name: 'usuario_editado' 	El usuario se edita correctamente y su nombre cambia a 'usuario_editado'.	Aprobada
PU016	Validar que un usuario no se edite con nombre duplicado	Usuario	<ul style="list-style-type: none"> ➤ name: 'usuario_duplicado' 	Error de validación por nombre duplicado.	Aprobada
PU017	Validar que un usuario se puede dar de baja	Usuario	<ul style="list-style-type: none"> ➤ usuario.id: ID del usuario a eliminar 	El usuario se elimina correctamente y ya no aparece en el sistema.	Aprobada
PU018	Validar que se puedan listar usuarios	Usuario	<ul style="list-style-type: none"> ➤ N/A 	Todos los usuarios registrados son	Aprobada

listados
correctamente.

A continuación, se presentan las seis pruebas unitarias desarrolladas en la tercera iteración.

```
1 from odoo.tests.common import TransactionCase
2 from odoo.exceptions import ValidationError
3 import base64
4
5 class TestUsuario_E_ImagenDeUsuario(TransactionCase):
6
7     def setUp(self):
8         super(TestUsuario_E_ImagenDeUsuario, self).setUp()
9         # Crear un usuario para las pruebas
10        self.usuario_data = {
11            'name': 'usuario_test',
12            'login': 'test@example.com',
13            'image_1920': base64.b64encode(b'Imagen_de_prueba').decode('utf-8')
14        }
15        self.usuario = self.env['res.users'].create(self.usuario_data)
16
17    def test_creacion_exitosa(self):
18        """Prueba que un usuario se registre correctamente"""
19        usuario_data = {
20            'name': 'usuario_test',
21            'login': 'test@example.com',
22            'image_1920': base64.b64encode(b'Imagen_de_prueba').decode('utf-8'),
23        }
24        nuevo_usuario = self.env['res.users'].create(usuario_data)
25        self.assertTrue(nuevo_usuario)
26        self.assertEqual(nuevo_usuario.name, 'usuario_test')
27
28    def test_creacion_fallida_imagen_invalida(self):
29        """Prueba que un usuario no se registre con una imagen inválida"""
30        usuario_data = {
31            'name': 'usuario_test',
32            'login': 'test@example.com',
33            'image_1920': '', # Imagen vacía
34        }
35        with self.assertRaises(ValidationError):
36            self.env['res.users'].create(usuario_data)
37
38    def test_edicion_exitosa(self):
39        """Prueba que un usuario se edite correctamente"""
40        self.usuario.write({'name': 'usuario_editado'})
41        self.assertEqual(self.usuario.name, 'usuario_editado')
42
43    def test_edicion_fallida_nombre_duplicado(self):
44        """Prueba que no se pueda editar un usuario con nombre duplicado"""
45        otro_usuario = self.env['res.users'].create({
46            'name': 'usuario_duplicado',
47            'login': 'duplicado@example.com',
48            'image_1920': base64.b64encode(b'Imagen_de_prueba').decode('utf-8'),
49        })
50        with self.assertRaises(ValidationError):
51            self.usuario.write({'name': 'usuario_duplicado'})
52
53    def test_dar_de_baja_usuario(self):
54        """Prueba que un usuario se puede dar de baja correctamente"""
55        self.usuario.unlink()
56        usuario_eliminado = self.env['res.users'].search([('id', '=', self.usuario.id)])
57        self.assertFalse(usuario_eliminado)
58
59
60    def test_listado_usuarios(self):
61        """Prueba que se pueden listar correctamente todos los usuarios"""
62        usuarios = self.env['res.users'].search([])
63        self.assertTrue(len(usuarios) > 0)
64
65
```

Figura 68. Clase de las pruebas unitarias para Equipo Informático

```
1 def test_creacion_exitosa(self):
2     """Prueba que un usuario se registre correctamente"""
3     usuario_data = {
4         'name': 'usuario_test',
5         'login': 'test@example.com',
6         'image_1920': base64.b64encode(b'imagen_de_prueba').decode('utf-8'),
7     }
8     nuevo_usuario = self.env['res.users'].create(usuario_data)
9     self.assertTrue(nuevo_usuario)
10    self.assertEqual(nuevo_usuario.name, 'usuario_test')
```

Figura 69. Prueba unitaria 13 (PU13), el Usuario se crea exitosamente

```
1 def test_creacion_fallida_imagen_invalida(self):
2     """Prueba que un usuario no se registre con una imagen inválida"""
3     usuario_data = {
4         'name': 'usuario_test',
5         'login': 'test@example.com',
6         'image_1920': '', # Imagen vacia
7     }
8     with self.assertRaises(ValidationError):
9         self.env['res.users'].create(usuario_data)
```

Figura 70. Prueba unitaria 14 (PU14), el Usuario no se registra por no subir la imagen de usuario


```
1 def test_edicion_exitosa(self):
2     """Prueba que un usuario se edite correctamente"""
3     self.usuario.write({'name': 'usuario_editado'})
4     self.assertEqual(self.usuario.name, 'usuario_editado')
```

Figura 71. Prueba unitaria 15 (PU15), el Usuario se edita exitosamente

```
1 def test_edicion_fallida_nombre_duplicado(self):
2     """Prueba que no se pueda editar un usuario con nombre duplicado"""
3     otro_usuario = self.env['res.users'].create({
4         'name': 'usuario_duplicado',
5         'login': 'duplicado@example.com',
6         'image_1920': base64.b64encode(b'imagen_de_prueba').decode('utf-8'),
7     })
8     with self.assertRaises(ValidationError):
9         self.usuario.write({'name': 'usuario_duplicado'})
```

Figura 72. Prueba unitaria 16 (PU16), el Usuario no se edita por nombre duplicado

```
1 def test_dar_de_baja_usuario(self):
2     """Prueba que un usuario se puede dar de baja correctamente"""
3     self.usuario.unlink()
4     usuario_eliminado = self.env['res.users'].search([('id', '=', self.usuario.id)])
5     self.assertFalse(usuario_eliminado)
```

Figura 73. Prueba unitaria 17 (PU17), el Usuario se da de baja exitosamente

```

1 def test_listado_usuarios(self):
2     """Prueba que se pueden listar correctamente todos los usuarios"""
3     usuarios = self.env['res.users'].search([])
4     self.assertTrue(len(usuarios) > 0)

```

Figura 74. Prueba unitaria 18 (PU18), se listan todos los Usuarios exitosamente

A continuación, se presentan los resultados obtenidos de las pruebas unitarias en la tercera iteración.

```

2024-10-15 09:53:52,811 10030 INFO odoo.tests.odoo.modules.loading: loading control_acceso/data/cron.xml
2024-10-15 09:53:52,843 10030 INFO odoo.tests.odoo.modules.loading: Module control_acceso: loading demo
2024-10-15 09:53:52,892 10030 INFO odoo.tests.odoo.http: HTTP Configuring static files
2024-10-15 09:53:53,003 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E
ImageneUsuario.test_creacion_exitosa ...
2024-10-15 09:53:53,431 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E
ImageneUsuario.test_creacion_fallida_imagen_invalida ...
2024-10-15 09:53:54,047 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E
ImageneUsuario.test_dar_de_baja_usuario ...
2024-10-15 09:53:54,664 10030 INFO odoo.tests.odoo.models.unlink: User #1 deleted mail.channel.partner records with IDs: [155]
2024-10-15 09:53:54,770 10030 INFO odoo.tests.odoo.models.unlink: User #1 deleted res.users records with IDs: [172]
2024-10-15 09:53:54,775 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E
ImageneUsuario.test_edicion_exitosa ...
2024-10-15 09:53:55,177 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E
ImageneUsuario.test_edicion_fallida_nombre_duplicado ...
2024-10-15 09:53:56,245 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E
ImageneUsuario.test_listado_usuarios ...
2024-10-15 09:53:56,742 10030 INFO odoo.tests.odoo.modules.loading: Module control_acceso loaded in 13.47s (incl. 3.68s test), 1613 que
ries (+488 test)
2024-10-15 09:53:56,743 10030 INFO odoo.tests.odoo.modules.loading: 31 modules loaded in 14.24s, 1613 queries (+488 extra)
2024-10-15 09:53:57,672 10030 INFO odoo.tests.odoo.modules.loading: Modules loaded.
2024-10-15 09:53:57,679 10030 INFO odoo.tests.odoo.modules.registry: Registry loaded in 17.304s
2024-10-15 09:53:57,680 10030 INFO odoo.tests.odoo.service.server: loading test file /opt/odoo/Integracion_Control_Acceso_Face_Recognit
ion/control_acceso/tests/test_usuario_e_imagen_de_usuario.py
2024-10-15 09:53:57,683 10030 INFO odoo.tests.odoo.service.server: running tests odoo.addons.control_acceso.tests.test_usuario_e_imagen
de_usuario.
2024-10-15 09:53:57,690 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E
ImageneUsuario.test_creacion_exitosa ...
2024-10-15 09:53:58,092 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E
ImageneUsuario.test_creacion_fallida_imagen_invalida ...
2024-10-15 09:53:58,774 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E
ImageneUsuario.test_dar_de_baja_usuario ...
2024-10-15 09:53:59,468 10030 INFO odoo.tests.odoo.models.unlink: User #1 deleted mail.channel.partner records with IDs: [162]
2024-10-15 09:53:59,558 10030 INFO odoo.tests.odoo.models.unlink: User #1 deleted res.users records with IDs: [179]
2024-10-15 09:53:59,564 10030 INFO odoo.tests.odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario_E

```

Figura 75. Ejecución de las pruebas unitarias de la tercera iteración, parte 1.

```

ImagenDeUsuario.test edicion exitosa ...
2024-10-15 09:53:55,177 10030 INFO odoo_tests odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario E
ImagenDeUsuario.test edicion fallida nombre duplicado ...
2024-10-15 09:53:56,245 10030 INFO odoo_tests odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario E
ImagenDeUsuario.test listado usuarios ...
2024-10-15 09:53:56,742 10030 INFO odoo_tests odoo.modules.loading: Module control_acceso loaded in 13.47s (incl. 3.68s test), 1613 que
ries (+488 test)
2024-10-15 09:53:56,743 10030 INFO odoo_tests odoo.modules.loading: 31 modules loaded in 14.24s, 1613 queries (+488 extra)
2024-10-15 09:53:57,672 10030 INFO odoo_tests odoo.modules.loading: Modules loaded.
2024-10-15 09:53:57,679 10030 INFO odoo_tests odoo.modules.registry: Registry loaded in 17.394s
2024-10-15 09:53:57,680 10030 INFO odoo_tests odoo.service.server: loading test file /opt/odoo/Integracion_Control_Acceso_Face_Recognit
ion/control_acceso/tests/test_usuario_e_imagen_de_usuario.py
2024-10-15 09:53:57,688 10030 INFO odoo_tests odoo.service.server: running tests odoo.addons.control_acceso.tests.test_usuario_e_imagen
_de_usuario.
2024-10-15 09:53:57,690 10030 INFO odoo_tests odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario E
ImagenDeUsuario.test creacion exitosa ...
2024-10-15 09:53:58,092 10030 INFO odoo_tests odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario E
ImagenDeUsuario.test creacion fallida imagen invalida ...
2024-10-15 09:53:58,774 10030 INFO odoo_tests odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario E
ImagenDeUsuario.test dar de baja usuario ...
2024-10-15 09:53:59,458 10030 INFO odoo_tests odoo.models.unlink: User #1 deleted mail.channel.partner records with IDs: [162]
2024-10-15 09:53:59,558 10030 INFO odoo_tests odoo.models.unlink: User #1 deleted res.users records with IDs: [179]
2024-10-15 09:53:59,564 10030 INFO odoo_tests odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario E
ImagenDeUsuario.test edicion exitosa ...
2024-10-15 09:53:59,970 10030 INFO odoo_tests odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario E
ImagenDeUsuario.test edicion fallida nombre duplicado ...
2024-10-15 09:54:01,060 10030 INFO odoo_tests odoo.addons.control_acceso.tests.test_usuario_e_imagen_de_usuario: Starting TestUsuario E
ImagenDeUsuario.test listado usuarios ...
2024-10-15 09:54:01,511 10030 INFO odoo_tests odoo.service.server: Starting post tests
2024-10-15 09:54:01,512 10030 INFO odoo_tests odoo.service.server: 0 post-tests in 0.00s, 0 queries
2024-10-15 09:54:01,513 10030 INFO odoo_tests odoo.tests.runner: 0 failed, 0 error(s) of 12 tests when loading database 'odoo_tests'
2024-10-15 09:54:01,513 10030 INFO odoo_tests odoo.service.server: Initiating shutdown
2024-10-15 09:54:01,513 10030 INFO odoo_tests odoo.service.server: Hit CTRL-C again or send a second signal to force the shutdown.
2024-10-15 09:54:01,517 10030 INFO odoo_tests odoo.sql_db: ConnectionPool(used=0/count=0/max=64): Closed 2 connections

```

Figura 76. Ejecución de las pruebas unitarias de la tercera iteración, parte 2.

4 Cuarta Iteración

4.1 Planificación

La **Tabla 7** muestra las actividades específicas a desarrollar en la cuarta iteración.

Tabla 7. Historias de Usuario a desarrollar en la cuarta iteración.

HU #	Historia de usuario	Fecha de creación	Fecha de entrega
HU14	Registrar asistencia Face ID	13/12/2023	10/01/2024
HU15	Listar asistencias Face ID	11/12/2023	11/01/2024
HU16	Generar reporte de asistencias Face ID	12/01/2024	12/01/2024
HU17	Imprimir reporte de asistencias Face ID	13/01/2024	13/01/2024

4.2 Diseño

La **Figura 33** muestra las funcionalidades de manera general a desarrollar en la cuarta iteración.



Figura 77. Casos de Uso general para la cuarta iteración.

4.2.1 Diagramas BPMN

A continuación, se presentan los diagramas BPMN pertenecientes a cada Historia de Usuario a desarrollar en la cuarta iteración.

4.2.1.1 HU14: Registrar asistencia Face ID

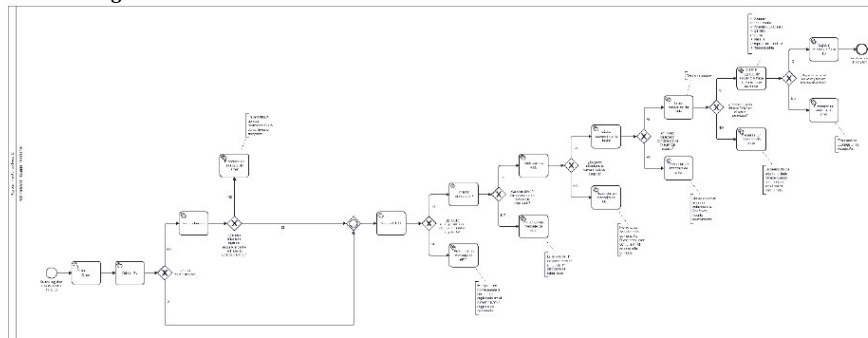


Figura 78. Diagrama BPMN de HU14.

4.2.1.2 HU15: Listar asistencias Face ID

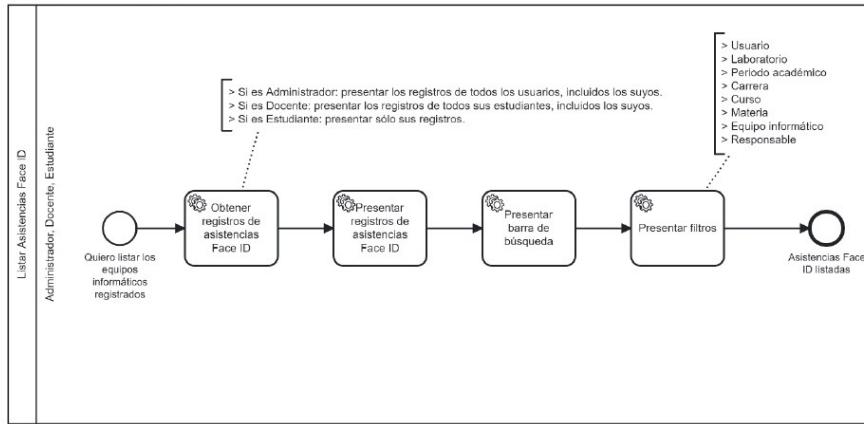


Figura 79. Diagrama BPMN de HU15.

4.2.1.3 HU16: Generar reporte de asistencias Face ID - HU17: Imprimir reporte de asistencias Face ID

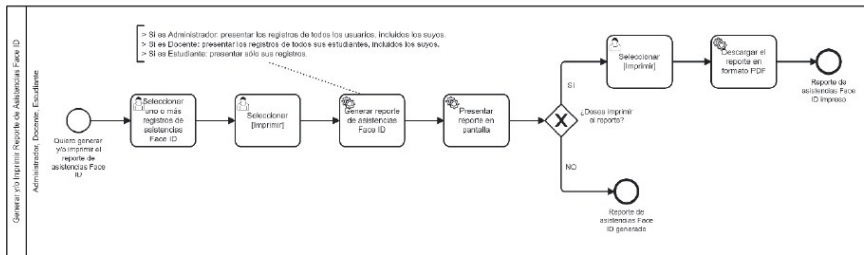


Figura 80. Diagrama BPMN de HU16 y HU17.

4.2.2 Prototipos

A continuación, se presentan los prototipos pertenecientes a cada Historia de Usuario a desarrollar en la cuarta iteración.

4.2.2.1 HU14: Registrar asistencia Face ID

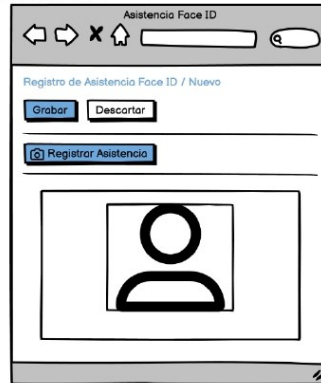


Figura 81. Prototipo para Registrar Asistencia Face ID.

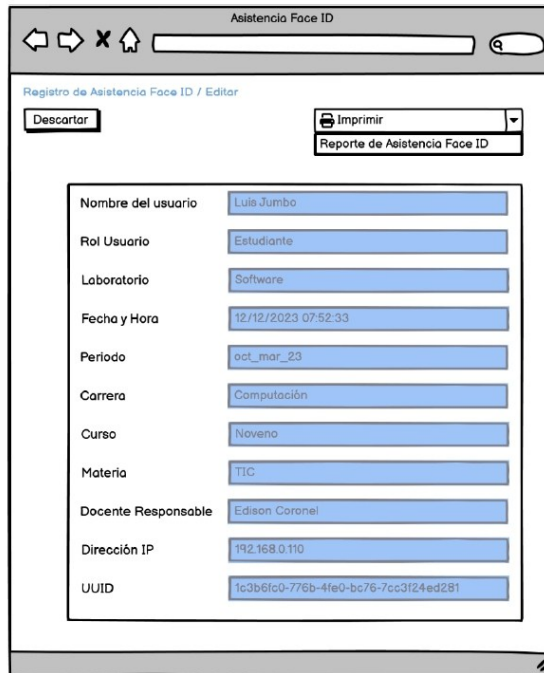
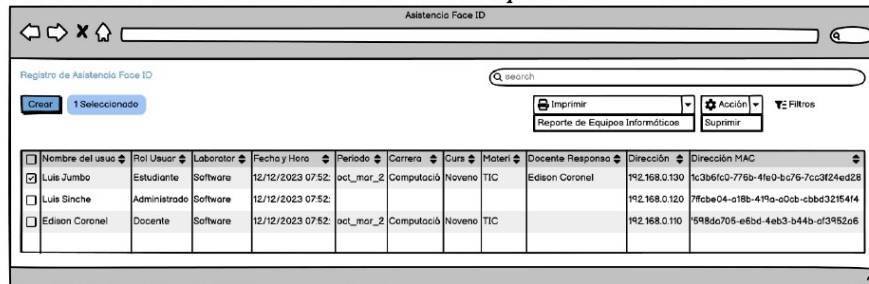


Figura 82. Prototipo para Visualizar Asistencia Face ID.

4.2.2.2 **HU15: Listar asistencias Face ID - HU16: Generar reporte de asistencias Face ID**



Asistencia Face ID

Registro de Asistencia Face ID

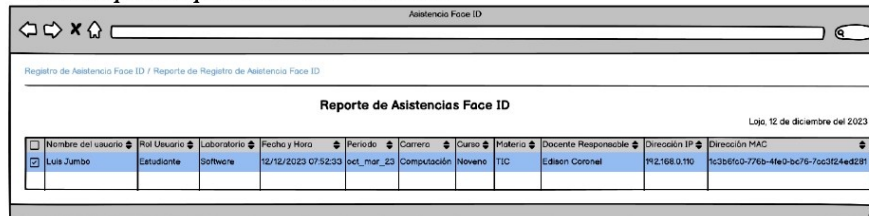
Crear 1 Seleccionado

Imprimir Reporte de Equipos Informáticos Acción Suprimir Filtros

Nombre del usua	Rol Usuario	Laborator	Fecha y Hora	Periodo	Carrera	Curso	Materi	Docente Responsa	Dirección	Dirección MAC
<input type="checkbox"/> Luis Jumbo	Estudiante	Software	12/12/2023 07:52	oct_mar_2	Computació	Noveno	TIC	Edison Coronel	192.168.0.130	1c3b6f60-776b-4fe0-bc76-7cc3f24ed28
<input type="checkbox"/> Luis Sinche	Administrado	Software	12/12/2023 07:52	oct_mar_2	Computació	Noveno	TIC		192.168.0.120	7f6be04-c18b-419a-c0ab-cbbd3216414
<input type="checkbox"/> Edison Coronel	Docente	Software	12/12/2023 07:52	oct_mar_2	Computació	Noveno	TIC		192.168.0.110	5988dc705-e6bd-4eb3-b44b-cf3952c6

Figura 83. Prototipo para Listar y Generar Reporte de Asistencia Face ID.

4.2.2.3 **HU17: Imprimir reporte de asistencias Face ID**



Asistencia Face ID

Registro de Asistencia Face ID / Reporte de Registro de Asistencia Face ID

Reporte de Asistencias Face ID

Loja, 12 de diciembre del 2023

Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	Dirección MAC
<input checked="" type="checkbox"/> Luis Jumbo	Estudiante	Software	12/12/2023 07:52:33	oct_mar_23	Computación	Noveno	TIC	Edison Coronel	192.168.0.110	1c3b6f60-776b-4fe0-bc76-7cc3f24ed28f

Figura 84. Prototipo para Imprimir Reporte de Asistencia Face ID.

4.3 **Codificación**

A continuación, se presenta los fragmentos principales del código fuente desarrollados en la cuarta iteración.

```

1 from odoo import api, fields, http, models
2 from odoo.exceptions import ValidationError
3
4 class RegistroAsistenciaFaceID(models.Model):
5     _name = 'controlacceso.registroasistenciafaceid'
6     _description = 'Registro de Asistencia Face ID'
7
8     # Campos de IDS (relaciones con otros modelos)
9     usuario_id = fields.Many2one('res.users', string='Usuario', ondelete='cascade', required=False)
10    lab_id = fields.Many2one('controlacceso.lab', string='Laboratorio', ondelete='cascade', required=False)
11    periodo_id = fields.Many2one('controlacceso.periodoacademico', string='Periodo académico', ondelete='cascade')
12    carrera_id = fields.Many2one('controlacceso.carrera', string='Carrera', ondelete='cascade')
13    curso_id = fields.Many2one('controlacceso.curso', string='Curso', ondelete='cascade')
14    materia_id = fields.Many2one('controlacceso.materia', string='Materia', ondelete='cascade')
15    equipo_informatico_id = fields.Many2one('controlacceso.equipoinformatico', string='Equipo Informático', ondelete='cascade')
16    responsable = fields.Many2one('controlacceso.docente2', string='Docente Responsable', ondelete='cascade')
17
18    # Campos Char relacionados (son los que va a visualizar el usuario)
19    nombre_usuario = fields.Char(related='usuario_id.name', string='Nombre del usuario', store=True)
20    nombre_lab = fields.Char(related='lab_id.nombre_lab', string='Laboratorio', store=True)
21    nombre_periodo = fields.Char(related='periodo_id.name', string='Periodo', store=True)
22    nombre_carrera = fields.Char(related='carrera_id.name', string='Carrera', store=True)
23    nombre_curso = fields.Char(related='curso_id.name', string='Curso', store=True)
24    nombre_materia = fields.Char(related='materia_id.name', string='Materia', store=True)
25    nombre_equipo_informatico = fields.Char(related='equipo_informatico_id.codigo_identificador', string='Equipo Informático', store=True)
26    nombre_responsable = fields.Char(related='responsable.name.name', string='Docente Responsable', store=True)
27    rol = fields.Selection([(Administrador, Administrador), (Docente, Docente)], ('Estudiante', 'Estudiante')), string='Rol Usuario', required=False)
28    fecha_y_hora = fields.Datetime(string='Fecha y Hora', required=False)
29    direccion_ip = fields.Char(string='Dirección IP', required=False)
30    uuid = fields.Char(string='UUID', required=False)

```

Figura 85. Modelo de Registro Asistencia Face ID.


```

1 <?xml version="1.0" encoding="utf-8"?>
2 <odoo>
3   <data>
4     <!-- Vista del formulario de registro de asistencia face ID -->
5     <record id="view_form_registro_asistencia_face_id" model="ir.ui.view">
6       <field name="name" registroasistenciafaceid.form/>
7       <field name="model" controlacceso.registroasistenciafaceid/>
8       <field name="arch" type="xml">
9         <form>
10          <sheet>
11            <group>
12              <field name="nombre_usuario" widget="char" readonly="1"/>
13              <field name="rol" widget="char" readonly="1"/>
14              <field name="nombre_lab" widget="char" readonly="1"/>
15              <field name="fecha_y_hora" widget="char" readonly="1"/>
16              <field name="nombre_periodo" widget="char" readonly="1"/>
17              <field name="nombre_carrera" widget="char" readonly="1"/>
18              <field name="nombre_curso" widget="char" readonly="1"/>
19              <field name="nombre_materia" widget="char" readonly="1"/>
20              <field name="nombre_responsable" widget="char" readonly="1"/>
21              <field name="direccion_ip" widget="char" readonly="1"/>
22              <field name="usid" widget="char" readonly="1"/>
23            </group>
24          </sheet>
25        </form>
26      </field>
27    </record>
28
29    <!-- Vista del árbol de registro de asistencia Face ID -->
30    <record id="view_tree_registro_asistencia_face_id" model="ir.ui.view">
31      <field name="name" registroasistenciafaceid.tree/>
32      <field name="model" controlacceso.registroasistenciafaceid/>
33      <field name="arch" type="xml">
34        <tree>
35          <field name="nombre_usuario"/>
36          <field name="rol"/>
37          <field name="nombre_lab"/>
38          <field name="fecha_y_hora"/>
39          <field name="nombre_periodo"/>
40          <field name="nombre_carrera"/>
41          <field name="nombre_curso"/>
42          <field name="nombre_materia"/>
43          <field name="nombre_responsable"/>
44          <field name="direccion_ip"/>
45          <field name="usid" string="UID"/>
46        </tree>
47      </field>
48    </record>
49
50    <!-- Vista de búsqueda de registro de asistencia Face ID -->
51    <record id="view_search_registro_asistencia_face_id" model="ir.ui.view">
52      <field name="name" registroasistenciafaceid.search/>
53      <field name="model" controlacceso.registroasistenciafaceid/>
54      <field name="arch" type="xml">
55        <search>
56          <field name="nombre_usuario" string="Usuario"/>
57          <field name="rol" string="Rol Usuario"/>
58          <field name="nombre_lab" string="Laboratorio"/>
59          <field name="fecha_y_hora" string="Fecha y Hora"/>
60          <field name="nombre_periodo" string="Período Académico"/>
61          <field name="nombre_carrera" string="Carrera"/>
62          <field name="nombre_curso" string="Curso"/>
63          <field name="nombre_materia" string="Materia"/>
64          <field name="nombre_responsable" string="Docente Responsable"/>
65          <field name="direccion_ip" string="Dirección IP"/>
66          <field name="usid" string="UID"/>
67        </search>
68      </field>
69    </record>
70
71    <!-- Acción para abrir la vista de registro de asistencia face ID -->
72    <record id="action_view_registroasistenciafaceid" model="ir.actions.act_window">
73      <field name="name" Registro de Asistencia Face ID/>
74      <field name="res_model" controlacceso.registroasistenciafaceid/>
75      <field name="view_mode" tree,form/>
76      <field name="view_id" ref="control_acceso.view_tree_registro_asistencia_face_id"/>
77      <field name="search_view_id" ref="control_acceso.view_search_registro_asistencia_face_id"/>
78      <field name="help" type="html">
79        <p class="oe_view_nocontent_create">
80          Click para crear un nuevo registro de asistencia.
81        </p>
82      </field>
83    </record>
84  </data>
85 </odoo>
86

```

Figura 86. Vista de Registro Asistencia Face ID.

```

1 </tbody>
2 </table>
3 </table>
4 </table>
5 </table>
6 </table>
7 </table>
8 </table>
9 </table>
10 </table>
11 </table>
12 </table>
13 </table>
14 </table>
15 </table>
16 </table>
17 </table>
18 </table>
19 </table>
20 </table>
21 </table>
22 </table>
23 </table>
24 </table>
25 </table>
26 </table>
27 </table>
28 </table>
29 </table>
30 </table>
31 </table>
32 </table>
33 </table>
34 </table>
35 </table>
36 </table>
37 </table>
38 </table>
39 </table>
40 </table>
41 </table>
42 </table>
43 </table>
44 </table>
45 </table>
46 </table>
47 </table>
48 </table>
49 </table>
50 </table>
51 </table>
52 </table>
53 </table>
54 </table>
55 </table>
56 </table>
57 </table>
58 </table>
59 </table>
60 </table>
61 </table>
62 </table>
63 </table>
64 </table>
65 </table>
66 </table>
67 </table>
68 </table>
69 </table>
70 </table>
71 </table>
72 </table>
73 </table>
74 </table>
75 </table>
76 </table>
77 </table>
78 </table>
79 </table>
80 </table>
81 </table>
82 </table>
83 </table>
84 </table>
85 </table>
86 </table>
87 </table>

```

Figura 87. Vista del Reporte de Registro Asistencia Face ID.

```

1 <!--=====FACE
2 ID=====-->
3 <record id="model_access_view_registroasistenciafaceid_admin" model="ir.rule">
4   <field name="name">model.access.view_registroasistenciafaceid.admin</field>
5   <field name="model_id" ref="control_acceso.model_controlacceso_registroasistenciafaceid"/>
6   <field name="groups" eval="(4, ref('&apos;control_acceso.res_groups_administrador&apos;'))"/>
7   <field name="perm_read" eval="1"/>
8   <field name="perm_write" eval="0"/>
9   <field name="perm_create" eval="1"/>
10  <field name="perm_unlink" eval="0"/>
11  <field name="domain_force">[(1, &apos;=&apos;, 1)]</field>
12 </record>
13 <record id="model_access_registroasistenciafaceid" model="ir.model.access">
14   <field name="name">model.access_registroasistenciafaceid</field>
15   <field name="model_id" ref="control_acceso.model_controlacceso_registroasistenciafaceid"/>
16   <field name="group_id" ref="base.group_system"/>
17   <field name="perm_read" eval="1"/>
18   <field name="perm_write" eval="0"/>
19   <field name="perm_create" eval="1"/>
20   <field name="perm_unlink" eval="0"/>
21 </record><!--=====FACE
22 ID=====-->
  
```

Figura 88. Permisos de Registro Asistencia Face ID “Administrador”.

```

1 <!--=====FACE
2 ID=====-->
3 <record id="model_access_view_registroasistenciafaceid_docentes" model="ir.model.access">
4   <field name="name">model.access.view_registroasistenciafaceid_docentes</field>
5   <field name="model_id" ref="control_acceso.model_controlacceso_registroasistenciafaceid"/>
6   <field name="group_id" ref="control_acceso.res_groups_docentes"/>
7   <field name="perm_read" eval="1"/>
8   <field name="perm_write" eval="0"/>
9   <field name="perm_create" eval="1"/>
10  <field name="perm_unlink" eval="0"/>
11 </record>
12 <record id="model_access_view_registroasistenciafaceid_docente" model="ir.rule">
13   <field name="name">model.access.view_registroasistenciafaceid.docente</field>
14   <field name="model_id" ref="control_acceso.model_controlacceso_registroasistenciafaceid"/>
15   <field name="groups" eval="(4, ref('&apos;control_acceso.res_groups_docentes&apos;'))"/>
16   <field name="perm_read" eval="1"/>
17   <field name="perm_write" eval="0"/>
18   <field name="perm_create" eval="1"/>
19   <field name="perm_unlink" eval="0"/>
20   <field name="domain_force">[&apos;=&apos;, (&apos;responsable&apos;, &apos;=&apos;, user.name), (&apos;usuario_id&apos;, &apos;=&apos;,
21   user.name)]</field>
22 </record><!--=====FACE
23 ID=====-->
  
```

Figura 89. Permisos de Registro Asistencia Face ID “Docente”.

```

1  <!--=====FACE
2  ID=====-->
3  <record id="model_access_view_registroasistenciafaceid_estudiante" model="ir.model.access">
4  <field name="name">model.access.view_registroasistenciafaceid_estudiante</field>
5  <field name="model_id" ref="control_acceso.model_controlacceso_registroasistenciafaceid"/>
6  <field name="group_id" ref="control_acceso.res_groups_alumnos"/>
7  <field name="perm_read" eval="1"/>
8  <field name="perm_write" eval="0"/>
9  <field name="perm_create" eval="1"/>
10 <field name="perm_unlink" eval="0"/>
11 </record>
12 <record id="model_access_view_registroasistenciafaceid_estudiantes" model="ir.model.access">
13 <field name="name">model.access.view_registroasistenciafaceid_estudiantes</field>
14 <field name="model_id" ref="control_acceso.model_controlacceso_registroasistenciafaceid"/>
15 <field name="group_id" ref="control_acceso.res_groups_alumnos"/>
16 <field name="perm_read" eval="1"/>
17 <field name="perm_write" eval="0"/>
18 <field name="perm_create" eval="1"/>
19 <field name="perm_unlink" eval="0"/>
20 </record>
21 <record id="model_acces_view_registroasistenciafaceid_estudiantes" model="ir.rule">
22 <field name="name">model.acces.view_registroasistenciafaceid_estudiantes</field>
23 <field name="model_id" ref="control_acceso.model_controlacceso_registroasistenciafaceid"/>
24 <field name="groups" eval="[(4, ref('&apos;control_acceso.res_groups_alumnos&apos;)))]"/>
25 <field name="perm_read" eval="1"/>
26 <field name="perm_write" eval="0"/>
27 <field name="perm_create" eval="1"/>
28 <field name="perm_unlink" eval="0"/>
29 <field name="domain_force">[(&apos;nombre_usuario&apos;, &apos;=&apos;, user.name)]</field>
30 </record><!--=====FACE
31 ID=====-->

```

Figura 90. Permisos de Registro Asistencia Face ID “Estudiante”.

```

1 <html version="1.0" encoding="UTF-8" ?>
2 <template xmlns="preserve">
3   <!-- t-navegacion: control: acceso: facialID -->
4   <!-- t-call-assets: web: assets: backend: t-35= false -->
5   <script type="text/javascript" src="https://cdn.jsdelivr.net/npm/sweetalert2@11"/></script>
6   <script type="text/javascript" src="https://cdn.jsdelivr.net/npm/bootstrap@3.3/dist/css/bootstrap.min.css"/></script>
7   <script type="text/javascript" src="https://cdn.jsdelivr.net/npm/bootstrap@3.3/dist/js/bootstrap.bundle.min.js"/></script>
8
9   <div class="container-fluid p-4" style="max-height: 100vh; overflow-y: auto;">
10     <!-- Titulo -->
11     <div class="text-center mb-4">
12       <h1 class="h2">Registro de Asistencia con Face ID (H5)</h1>
13     </div>
14
15     <!-- Contenedor principal -->
16     <div class="row p-4">
17       <!-- Columna izquierda: Formulario -->
18       <div class="col-lg-6 col-md-12">
19         <div class="card shadow-sm">
20           <div class="card-header bg-primary text-white">
21             <h3 class="mb-0 text-white">Datos del Usuario (H5)</h3>
22           </div>
23           <div class="card-body">
24             <form id="facial-form">
25               <!-- Grupo de campos -->
26               <div class="mb-3 d-flex align-items-center">
27                 <label for="nombre_usuario" class="form-label me-3 flex-shrink-0" style="width: 30%;>Usuario(</label>
28                 <input type="text" id="nombre_usuario" class="form-control"
29                 name="nombre_usuario" t-att-value="state.asistenciaIdDict.nombre_usuario" | "" readonly="readonly"/>
30               </div>
31               <div class="mb-3 d-flex align-items-center">
32                 <label for="rol" class="form-label me-3 flex-shrink-0" style="width: 30%;>Rol(</label>
33                 <input type="text" id="rol" class="form-control"
34                 name="rol" t-att-value="state.asistenciaIdDict.rol" | "" readonly="readonly"/>
35               </div>
36               <div class="mb-3 d-flex align-items-center">
37                 <label for="nombre_laboratorio" class="form-label me-3 flex-shrink-0" style="width: 30%;>Laboratorio(</label>
38                 <input type="text" id="nombre_laboratorio" class="form-control"
39                 name="nombre_laboratorio" t-att-value="state.asistenciaIdDict.nombre_laboratorio" | "" readonly="readonly"/>
40               </div>
41               <div class="mb-3 d-flex align-items-center">
42                 <label for="fecha_y_hora_zona_horaria_actual" class="form-label me-3 flex-shrink-0" style="width: 30%;>Fecha y hora(</label>
43                 <input type="text" id="fecha_y_hora_zona_horaria_actual" class="form-control"
44                 name="fecha_y_hora_zona_horaria_actual" t-att-value="state.asistenciaIdDict.fecha_y_hora_zona_horaria_actual" | "" readonly="readonly"/>
45               </div>
46               <div class="mb-3 d-flex align-items-center">
47                 <label for="direccion_ip" class="form-label me-3 flex-shrink-0" style="width: 30%;>Dirección IP(</label>
48                 <input type="text" id="direccion_ip" class="form-control"
49                 name="direccion_ip" t-att-value="state.asistenciaIdDict.direccion_ip" | "" readonly="readonly"/>
50               </div>
51               <div class="mb-3 d-flex align-items-center">
52                 <label for="uid" class="form-label me-3 flex-shrink-0" style="width: 30%;>UID(</label>
53                 <input type="text" id="uid" class="form-control"
54                 name="uid" t-att-value="state.asistenciaIdDict.uid" | "" readonly="readonly"/>
55               </div>
56             </form>
57           </div>
58         </div>
59
60         <!-- Columna derecha: Video y Camara -->
61         <div class="col-lg-6 col-md-12">
62           <div class="card shadow-sm">
63             <div class="card-header bg-primary text-white">
64               <h3 class="mb-0 text-white">Reconocimiento Facial (H5)</h3>
65             </div>
66             <div class="card-body text-center">
67               <!-- Contenedor de video -->
68               <div id="video-container" class="mb-3">
69                 <video id="video" width="100%" class="rounded border" autoplay="true" playsinline="true"/></video>
70                 <canvas id="canvasOutput" width="540" height="380" style="display:none"/></canvas>
71               </div>
72               <!-- Boton para registrar asistencia -->
73               <button id="btn-registrar-asistencia" class="btn btn-primary btn-lg w-100" t-on-click="handleRegister">
74                 Registrar Asistencia
75               </button>
76             </div>
77           </div>
78
79         </div>
80       </div>
81     </div>
82   </div>
83 </template>
84 </compiatos>

```

Figura 91. Template para el Registro Asistencia Face ID.

```
1  /**
2   * Función para manejar el flujo al hacer click en "Registrar Asistencia".
3   */
4   async handleRegister() {
5     console.log("Iniciando flujo de registro de asistencia...");
6
7     try {
8       const button = document.getElementById("btn-registrar-asistencia");
9       this.disableButton(button);
10
11       // Leer el UUID desde las cookies
12       const uuid_recibido = this.getCookieValue("equipo_informatico_uuid");
13
14       if (!uuid_recibido) {
15         this.showNotification(-1, "No se encontró el UUID en las cookies. Asegúrate de haber configurado el dispositivo correctamente.");
16         this.enableButton(button);
17         return;
18       }
19
20       // Llamada inicial al endpoint para obtener datos del usuario
21       const userData = await this.rpc("/api/iniciar_validacion_horario", {
22         method: "POST",
23         args: [],
24         kwargs: {
25           uuid_recibido: uuid_recibido, // Enviar el UUID al controlador
26         },
27       });
28
29       if (userData.codigo === 1 && userData.asistencia_dict) {
30         // Actualizar el estado con los datos recibidos
31         this.state.records = [userData.asistencia_dict];
32         this.state.asistencia_dict = userData.asistencia_dict;
33         // CARGAR LOS DATOS QUE SE VAN A PRESENTAR EN EL FORMULARIO
34
35         // Mostrar notificación y proceder con el reconocimiento facial
36         this.showNotification(1, "Datos cargados. Procediendo al reconocimiento facial...");
37         await this.startFaceRecognition(userData.asistencia_dict, button);
38       } else {
39         this.showNotification(-1, userData.mensaje || "Error al obtener datos del usuario.");
40         this.enableButton(button);
41       }
42     } catch (error) {
43       const button = document.getElementById("btn-registrar-asistencia");
44       console.error("Error durante el flujo de registro:", error);
45       this.showNotification(-1, "Error inesperado al procesar el registro.");
46       this.enableButton(button);
47     }
48   }
49 }
```

Figura 92. Función JS para manejar todo el flujo cuando se hace click en el botón, y enviar a validar el horario en el backend de Odo.

```
1 /**
2  * Inicia el reconocimiento facial utilizando la cámara del usuario.
3  */
4  async startFaceRecognition(asistenciaDict, button) {
5      console.log("Iniciando reconocimiento facial...");
6
7      const video = document.getElementById("video");
8      const canvas = document.getElementById("canvasOutput");
9      const context = canvas.getContext("2d");
10
11      try {
12          const stream = await navigator.mediaDevices.getUserMedia({ video: true });
13          video.srcObject = stream;
14          await video.play();
15
16          const frames = await this.captureFrames(video, canvas, context);
17
18          await this.processFaceRecognition(frames, asistenciaDict, video, context, canvas, button);
19      } catch (error) {
20          this.showNotification(-1, "No se pudo acceder a la cámara. Verifica los permisos.");
21          console.error("Error de cámara:", error);
22          this.cleanupResources(video, context, canvas, button);
23      }
24  }
```

Figura 93. Función JS para abrir la cámara web y empezar a transmitir en vivo lo capturado por la misma.

```
1 /**
2  * Captura frames del video para el reconocimiento facial.
3  */
4  captureFrames(video, canvas, context) {
5      return new Promise((resolve) => {
6          const frames = [];
7          let frameCount = 0;
8          const maxFrames = 24;
9
10         const captureInterval = setInterval(() => {
11             context.drawImage(video, 0, 0, canvas.width, canvas.height);
12             frames.push(canvas.toDataURL("image/jpeg"));
13             frameCount++;
14
15             if (frameCount >= maxFrames) {
16                 clearInterval(captureInterval);
17                 resolve(frames);
18             }
19         }, 200);
20     });
21 }
```

Figura 94. Función JS para capturar los frames del rostro del usuario.

```

1  /**
2   * Procesa el reconocimiento facial enviando los frames capturados al backend.
3   */
4   async processFaceRecognition(frames, asistenciaDict, video, context, canvas, button) {
5     console.log("Procesando reconocimiento facial...");
6
7     try {
8       const response = await this.rpc('/api/iniciar_reconocimiento_facial', {
9         method: "POST",
10        args: [],
11        kwargs: {
12          images_data_urls: { accumulatedFrames: frames },
13          asistencia_dict: asistenciaDict,
14        },
15      });
16
17      if (response.codigo === 1 && response.match && response.registro_creado) {
18        this.showNotification(1, response.mensaje);
19        this.cleanupResources(video, context, canvas, button);
20
21        // Redirigir a la acción existente
22        await this.actionService.doAction("control_acceso.action_view_registroasistenciafaceid");
23      } else {
24        this.showNotification(-1, response.mensaje || "Error en el reconocimiento facial.");
25        this.cleanupResources(video, context, canvas, button);
26      }
27    } catch (error) {
28      this.showNotification(-1, "Error durante el reconocimiento facial.");
29      console.error("Error en reconocimiento facial:", error);
30      this.cleanupResources(video, context, canvas, button);
31    }
32  }

```

Figura 95. Función JS para enviar los frames obtenidos del rostro del usuario al backend de Odo.

```

1  /**
2   * Muestra notificaciones al usuario.
3   */
4   showNotification(codigo, mensaje, temporizador = null, position = "top-end") {
5     const title = codigo === 1 ? "Éxito" : "Error";
6     const icon = codigo === 1 ? "success" : "error";
7
8     Swal.fire({
9       title,
10      text: mensaje,
11      icon,
12      timer: temporizador,
13      timerProgressBar: !temporizador,
14      position,
15      showConfirmButton: !temporizador,
16    });
17  }

```

Figura 96. Función JS para mostrar las notificaciones en el navegador.


```
1 def success_response(self, codigo, mensaje, asistencia_dict=None, match=None, registro_creado=None, informacion_red=None, uuid=None):
2     response_data = {}
3     if codigo:
4         response_data['codigo'] = codigo
5     if mensaje:
6         response_data['mensaje'] = mensaje
7     if asistencia_dict:
8         response_data['asistencia_dict'] = asistencia_dict
9     if match:
10        response_data['match'] = match
11    if registro_creado:
12        response_data['registro_creado'] = registro_creado
13    if informacion_red:
14        response_data['informacion_red'] = informacion_red
15    if uuid:
16        response_data['uuid'] = uuid
17    return response_data
18
19 def error_response(self, codigo, mensaje):
20     # Verificar si estamos en un entorno de prueba
21     if os.getenv('AMBIENTE') == 'test':
22         return {'codigo': codigo, 'mensaje': mensaje}
23     return {'codigo': codigo, 'mensaje': mensaje}
```

Figura 97. Función en el controlador de Registro de Asistencia Face ID para el manejo de respuestas negativas (errores) y positivas.

```

1 @http.route('/api/iniciar_validacion_horario', type='json', methods=['POST'], auth='user', csrf=False)
2 def iniciar_validacion_horario(self, **kwargs):
3     try:
4
5         usuario_actual = request.env.user
6         # Capturar el contenido completo de kwargs
7         kwargs_data = kwargs.get('kwargs')
8
9         # Obtener el UUID
10        uuid_recibido = kwargs_data.get('uuid_recibido')
11
12        if not uuid_recibido:
13            return self.error_response(-1, 'El UUID no fue proporcionado.')
14
15        _logger.info(f'UUID recibido: {uuid_recibido}')
16
17        # Validar si el usuario SI es administrador
18        if usuario_actual.has_group('control_acceso.res_groups_administrador'):
19            # Si es administrador, puede continuar sin validar horarios
20            asistencia_dict = self.validar_uuid_sin_horario(usuario_actual, uuid_recibido)
21            if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
22                # Caso de error
23                return asistencia_dict
24            else:
25                # Caso exitoso
26                asistencia_dict = self.validar_direccion_ip_con_rango_ip_de_laboratorio(asistencia_dict)
27
28                if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
29                    # Caso de error
30                    return asistencia_dict
31                else:
32                    return self.success_response(1, "Datos validados para el administrador", asistencia_dict)
33
34        # Validar si el usuario SI es docente
35        elif usuario_actual.has_group('control_acceso.res_groups_docentes'):
36            # Lógica para docentes
37            asistencia_dict = self.validar_horario("Docente")
38
39            if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
40                # Caso de error
41                return asistencia_dict
42            else:
43                # Caso exitoso
44                asistencia_dict = self.validar_uuid_con_horario(asistencia_dict, uuid_recibido)
45
46                if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
47                    # Caso de error
48                    _logger.error(f'Error detectado MAC CON HORARIO: {asistencia_dict["mensaje"]}')
49                    return asistencia_dict
50                else:
51                    asistencia_dict = self.validar_direccion_ip_con_rango_ip_de_laboratorio(asistencia_dict)
52                    if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
53                        # Caso de error
54                        _logger.error(f'Error detectado DIRECCION IP CON HORARIO: {asistencia_dict["mensaje"]}')
55                        return asistencia_dict
56                    else:
57                        return self.success_response(1, "Datos validados para el administrador", asistencia_dict)
58
59        elif usuario_actual.has_group('control_acceso.res_groups_alumnos'):
60            # Lógica para estudiantes
61            asistencia_dict = self.validar_horario("Estudiante")
62
63            if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
64                # Caso de error
65                _logger.error(f'Error detectado HORARIO: {asistencia_dict["mensaje"]}')
66                return asistencia_dict
67            else:
68                # Caso exitoso
69                asistencia_dict = self.validar_uuid_con_horario(asistencia_dict, uuid_recibido)
70
71                if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
72                    # Caso de error
73                    _logger.error(f'Error detectado MAC CON HORARIO: {asistencia_dict["mensaje"]}')
74                    return asistencia_dict
75                else:
76                    asistencia_dict = self.validar_direccion_ip_con_rango_ip_de_laboratorio(asistencia_dict)
77                    if 'codigo' in asistencia_dict and asistencia_dict['codigo'] == -1 and 'mensaje' in asistencia_dict:
78                        # Caso de error
79                        _logger.error(f'Error detectado DIRECCION IP CON HORARIO: {asistencia_dict["mensaje"]}')
80                        return asistencia_dict
81                    else:
82                        return self.success_response(1, "Datos validados para el administrador", asistencia_dict)
83            else:
84                return self.error_response(-1, "Horario NO Válido, el usuario no pertenece a ningún grupo en el sistema")
85        except Exception as e:
86            return self.error_response(-1, str(e))

```

Figura 98. Endpoint en el controlador de Registro de Asistencia Face ID para “Validar Horario” según el rol de usuario.

```
1 def validar_horario(self, rol):
2     local_now = self._obtener_hora_local()
3     fecha_y_hora = self._convertir_fecha_hora(local_now)
4     day_of_week, time_str = self._obtener_dia_y_hora(local_now)
5
6     if rol == "Estudiante":
7         estudiante = request.env['controlacceso.usuarioestudiante'].search([('name', '=', request.env.user.name)])
8         for curso in estudiante.curso_ids:
9             horario_actual = self._obtener_horarios_de_usuario(controlacceso.horario2, 'controlacceso.usuarioestudiante', 'curso_id', day_of_week, time_str, local_now, fecha_y_hora)
10            if 'codigo' in horario_actual and horario_actual['codigo'] == -1 and 'mensaje' in horario_actual:
11                return horario_actual
12            else:
13                return self._create_asistencia_dict(horario_actual, day_of_week, fecha_y_hora, rol)
14
15        elif rol == "Docente":
16            horario_actual = self._obtener_horarios_de_usuario(controlacceso.horario2, 'controlacceso.docente2', 'docente_id', day_of_week, time_str, local_now, fecha_y_hora)
17            if 'codigo' in horario_actual and horario_actual['codigo'] == -1 and 'mensaje' in horario_actual:
18                return horario_actual
19            else:
20                return self._create_asistencia_dict(horario_actual, day_of_week, fecha_y_hora, rol)
```

Figura 99. Función en el controlador de Registro de Asistencia Face ID para para “Validar Horario” según el rol de usuario.

```

1 @http.route('/api/initial_reconocimiento_facial', type='json', methods=['POST'], auth='user', csrf=False)
2 def initial_reconocimiento_facial(self, request):
3     try:
4         # Capturar el contenido recibido de la request
5         kwargs_data = kwargs.get('kwargs')
6
7         # Recibir de objeto 'asistencia_dict'
8         asistencia_dict = kwargs_data.get('asistencia_dict')
9
10        # Acceder al objeto 'imagenes_data_urls'
11        imagenes_data_urls = kwargs_data.get('imagenes_data_urls')
12
13        # Validar que haya frames
14        if not asistencia_dict:
15            return self.error_response(-1, "No se recibió el directorio con los datos de usuario")
16
17        # Validar que haya frames
18        if not imagenes_data_urls:
19            return self.error_response(-1, "No se recibieron los frames para el reconocimiento facial")
20
21        asistencia_dict['diccionario'] = {}
22        imagenes_data_urls['diccionario'] = {}
23
24        # Recorrer de que es un diccionario
25        if not isinstance(asistencia_dict, dict):
26            try:
27                asistencia_dict['diccionario'] = json.loads(asistencia_dict)
28            except Exception as e:
29                return self.error_response(-1, f"Error al procesar asistencia_dict: {e}")
30        else:
31            asistencia_dict['diccionario'] = asistencia_dict
32
33        # Obtener imagenes_data_urls directamente
34        imagenes_data_urls['diccionario'] = imagenes_data_urls if imagenes_data_urls else {}
35
36        # Obtener el arreglo de imagenes desde el diccionario
37        accumulated_frames = imagenes_data_urls['diccionario'].get('accumulated_frames', [])
38
39        # Convertir las imagenes decodificadas en arrays numpy
40        imagenes_decoded = [base64.b64decode(image_data_url.split(',')[1]) for image_data_url in accumulated_frames]
41        imagenes_np = [cv.cvtColor(image_decoded[i].decode('utf-8'), cv.COLOR_BGR2GRAY) for image_decoded in imagenes_decoded]
42
43        # Lista para almacenar los resultados de la detección de vida
44        live_inputs = []
45        # Procesar cada imagen para la detección de vida
46        for image_np in imagenes_np:
47            # Rescaleamos la imagen a tamaño
48            resized_image = cv.resize(image_np, (100, 100))
49            # Convertir la imagen a escala de grises
50            gray_image = cv.cvtColor(resized_image, cv.COLOR_BGR2GRAY)
51            # Agregar la imagen preprocesada a la lista de resultados de la detección de vida
52            live_inputs.append(gray_image)
53
54        usuario_actual = request.user.username
55        nombre_usuario = usuario_actual.name
56        ruta_imagen_usuario = os.path.join(PATH_PROYECTO_IMAGES, nombre_usuario + ".png")
57        if (os.path.exists(ruta_imagen_usuario)):
58            imagen_usuario = face_recognition.load_image_file(ruta_imagen_usuario)
59            imagen_usuario_encoding = face_recognition.face_encodings(imagen_usuario)[0]
60        else:
61            return self.error_response(-1, "No se encontró la imagen de usuario en el servidor")
62
63        # Procesar cada imagen para realizar la comparación facial
64        for image_np in imagenes_np:
65            face_encodings = face_recognition.face_encodings(image_np)
66            if len(face_encodings) > 0:
67                match = face_recognition.compare_faces([imagen_usuario_encoding], face_encodings[0])
68                if match[0]:
69                    try:
70                        _logger.info("Datos enviados a la API externa: {live_inputs}")
71
72                        # Convertir cada ndarray en una lista
73                        live_inputs_serialized = [live_input.tolist() for live_input in live_inputs]
74
75                        # Enviar al API externo
76                        response = requests.post(
77                            URL_API_EXTERNAL,
78                            json={"live_inputs_serialized": live_inputs_serialized})
79
80                        match_status = json.loads(response.text).get('match')
81                        if response.status_code == 200 and match_status == True:
82                            registro_creado = self.crear_registro_facial(asistencia_dict['diccionario'])
83                            return self.success_response(codigo=1, mensaje="Identidad encontrada y registro creado exitosamente", match=True, registro_creado=registro_creado.id)
84                        return self.error_response(-1, "La detección de vida ha fallado, no se detecta vida en el rostro comparado.")
85                    except Exception as e:
86                        _logger.error(f"Error al conectar con la API externa: {e}")
87                        return self.error_response(-1, str(e))
88
89        # Si no se encuentra ninguna coincidencia en ninguna de las imagenes
90        return self.success_response(codigo=-1, mensaje="No se encontró ninguna coincidencia. Por favor, intente nuevamente.", match=False)
91
92    except Exception as e:
93        return json.dumps({
94            "status": "error",
95            "mensaje": str(e)
96        })

```

Figura 100. Endpoint en el controlador de Registro de Asistencia Face ID para "Iniciar Reconocimiento Facial".

```

1 def livness(self):
2     model = Sequential()
3     model.add(Conv3D(32, kernel_size=(3, 3, 3), activation='relu', input_shape=(24,100,100,1)))
4     model.add(Conv3D(64, (3, 3, 3), activation='relu'))
5     model.add(MaxPooling3D(pool_size=(2, 2, 2)))
6     model.add(Conv3D(64, (3, 3, 3), activation='relu'))
7     model.add(MaxPooling3D(pool_size=(2, 2, 2)))
8     model.add(Conv3D(64, (3, 3, 3), activation='relu'))
9     model.add(MaxPooling3D(pool_size=(2, 2, 2)))
10    model.add(Dropout(0.25))
11    model.add(Flatten())
12    model.add(Dense(128, activation='relu'))
13    model.add(Dropout(0.5))
14    model.add(Dense(2, activation='softmax'))
15    model.load_weights(PATH_MODULO_MODEL)
16    return model

```

Figura 101. Función en el controlador de Registro de Asistencia Face ID para cargar el modelo de detección de vida.

```

1 def crear_registro_face_id(self, asistencia_dict_diccionario):
2     registro_creado = request.env['controlacceso.registroasistenciafaceid'].sudo().create({
3         'usuario_id': asistencia_dict_diccionario.get('usuario_id', False),
4         'rol': asistencia_dict_diccionario.get('rol', False),
5         'lab_id': asistencia_dict_diccionario.get('laboratorio_id', False),
6         'fecha_y_hora': asistencia_dict_diccionario.get('fecha_y_hora', ''),
7         'direccion_ip': asistencia_dict_diccionario.get('direccion_ip', ''),
8         'periodo_id': asistencia_dict_diccionario.get('periodo_id', False),
9         'carrera_id': asistencia_dict_diccionario.get('carrera_id', False),
10        'curso_id': asistencia_dict_diccionario.get('curso_id', False),
11        'materia_id': asistencia_dict_diccionario.get('materia_id', False),
12        'responsable': asistencia_dict_diccionario.get('responsable', False),
13        'uuid': asistencia_dict_diccionario.get('uuid', False),
14    })
15    return registro_creado

```

Figura 102. Función en el controlador de Registro de Asistencia Face ID para crear un nuevo registro en la base de datos.

4.4 Pruebas

La **Tabla 8** muestra las pruebas unitarias a desarrollar en la cuarta iteración.

Tabla 8. Pruebas unitarias a ejecutar en la cuarta iteración

ID	Descripción	Modelo	Datos de Entrada	Salida Esperada	Resultado
PU019	Validar que el equipo no está habilitado para el registro de asistencia	Registro Asistencia Face ID: UUID	➤ Simulación de equipo no habilitado: {'codigo': -1, 'mensaje': 'El equipo no se encuentra habilitado para el registro de asistencia.'}	Respuesta de error: {'codigo': -1, 'mensaje': 'El equipo no se encuentra habilitado para el registro de asistencia.'}	Aprobada
PU020	Validar que el equipo no está registrado para el registro de asistencia	Registro Asistencia Face ID: UUID	➤ Simulación de equipo no registrado: {'codigo': -1, 'mensaje': 'El equipo no se encuentra registrado para el registro de asistencia.'}	Respuesta de error para equipo no registrado: {'codigo': -1, 'mensaje': 'El equipo no se encuentra registrado para el registro de asistencia.'}	Aprobada
PU021	Validar que no se encontraron interfaces de red	Registro Asistencia Face ID: UUID	➤ Simulación de equipo sin interfaces de red: {'codigo': -1, 'mensaje': 'No se encontraron interfaces de red.'}	Respuesta de error para equipo sin interfaces de red: {'codigo': -1, 'mensaje': 'No se encontraron interfaces de red.'}	Aprobada
PU022	Validar registro exitoso de un equipo habilitado para el registro de asistencia	Registro Asistencia Face ID: UUID	➤ Simulación de equipo habilitado: {'usuario_id': 1, 'laboratorio_id': 2, 'equipo_informatico_id': 3, 'rol': 'Administrador', 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281', 'fecha_y_hora': '2024-10-15 10:00:00'}	Simulación de equipo habilitado: {'usuario_id': 1, 'laboratorio_id': 2, 'equipo_informatico_id': 3, 'rol': 'Administrador', 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281', 'fecha_y_hora': '2024-10-15 10:00:00'}	Aprobada
PU023	Validar que la dirección IP del equipo no pertenece al laboratorio asociado para el registro de asistencia.	Registro Asistencia Face ID: Dirección IP	➤ Simulación de equipo no pertenece al laboratorio asociado: {'codigo': -1, 'mensaje': 'La dirección IP no pertenece al rango de IP asociada al laboratorio.'}	Respuesta de error: {'codigo': -1, 'mensaje': 'La dirección IP no pertenece al rango de IP asociada al laboratorio.'}	Aprobada
PU024	Validar que no se puede	Registro Asistencia	➤ Simulación de equipo no se puede obtener la dirección IP: {'codigo': -1,	Respuesta de error: {'codigo': -1,	Aprobada

	obtener la dirección IP del equipo para el registro de asistencia.	Face ID: Dirección IP	'mensaje': 'Error: No se pudo obtener la dirección IP del usuario.}'	'mensaje': 'Error: No se pudo obtener la dirección IP del usuario.}'	
PU025	Validar el registro de asistencia cuando el rango IP del laboratorio está deshabilitado.	Registro de Asistencia Face ID: Dirección IP	➤ Simulación de registro de asistencia con el rango IP del laboratorio deshabilitado: {laboratorio_id: self.laboratorio.id, 'usuario_id': self.administrador.id, 'equipo_informatico_id': self.equipo_informatico.id, 'fecha_y_hora': '2024-03-21 10:00:00', 'rol': 'Administrador', 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281'}	Diccionario con claves: usuario_id, laboratorio_id, equipo_informatico_id, fecha_y_hora, rol, uuid, direccion_ip	Aprobada
PU026	Validar el registro de asistencia cuando la dirección IP del equipo se encuentra dentro del rango IP del laboratorio.	Registro de Asistencia Face ID: Dirección IP	➤ <input type="checkbox"/> Simulación de registro de asistencia con el rango IP del laboratorio deshabilitado: {laboratorio_id: self.laboratorio.id, 'usuario_id': self.administrador.id, 'equipo_informatico_id': self.equipo_informatico.id, 'fecha_y_hora': '2024-03-21 10:00:00', 'rol': 'Administrador', 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281'}	Diccionario con claves: usuario_id, laboratorio_id, equipo_informatico_id, fecha_y_hora, rol, uuid, direccion_ip	Aprobada
PU027	Validar rechazo de registro por falta de horario de estudiante	Registro de Asistencia Face ID: Validar Horario	➤ Simulación de estudiante sin horario: {usuario_id: 1, 'rol': 'Estudiante', 'laboratorio_id': 1, 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281', 'fecha_y_hora': '2024-10-15 10:00:00'}	{'codigo': -1, 'mensaje': 'No posee ningún horario acorde al día y hora actual.'}	Aprobada
PU028	Validar rechazo de asistencia ya registrada por estudiante	Registro de Asistencia Face ID: Validar Horario	➤ Simulación de estudiante con asistencia ya registrada: {usuario_id: 1, 'rol': 'Estudiante', 'laboratorio_id': 1, 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281', 'fecha_y_hora': '2024-10-15 10:00:00'}	{'codigo': -1, 'mensaje': 'Ya ha marcado asistencia para este horario hoy.'}	Aprobada
PU029	Validar registro exitoso de estudiante con horario correcto	Registro de Asistencia Face ID: Validar Horario	➤ Simulación de estudiante con horario correcto: {usuario_id: 7, 'rol': 'Estudiante', 'laboratorio_id': 1, 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281', 'fecha_y_hora': '2024-10-15 10:00:00'}	{'usuario_id': 7, 'responsable': 1, 'curso_id': 1, 'laboratorio_id': 1, 'rol': 'Estudiante'}	Aprobada
PU030	Validar rechazo de registro por falta de horario de docente	Registro de Asistencia Face ID: Validar Horario	➤ Simulación de docente sin horario: {usuario_id: 2, 'rol': 'Docente', 'laboratorio_id': 1, 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281', 'fecha_y_hora': '2024-10-15 10:00:00'}	{'codigo': -1, 'mensaje': 'No posee ningún horario acorde al día y hora actual.'}	Aprobada

PU031	Validar rechazo de asistencia ya registrada por docente	Registro de Asistencia Face ID: Validar Horario	➤ Simulación de docente con asistencia ya registrada: {'usuario_id': 2, 'rol': 'Docente', 'laboratorio_id': 1, 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281', 'fecha_y_hora': '2024-10-15 10:00:00'}	{'codigo': -1, 'mensaje': 'Ya ha marcado asistencia para este horario hoy.'}	Aprobada
PU032	Validar registro exitoso de docente con horario correcto	Registro de Asistencia Face ID: Validar Horario	➤ Simulación de docente con horario correcto: {'usuario_id': 7, 'rol': 'Docente', 'laboratorio_id': 1, 'uuid': '1c3b6fc0-776b-4fe0-bc76-7cc3f24ed281', 'fecha_y_hora': '2024-10-15 10:00:00'}	{'usuario_id': 7, 'responsable': 1, 'curso_id': 1, 'laboratorio_id': 1, 'rol': 'Docente'}	Aprobada

A continuación, se presentan las pruebas unitarias desarrolladas en la cuarta iteración.

```

1 class TestRegistroAsistenciaFaceIDBase(TestCase):
2     # Setup común para todas las pruebas
3     def setUp(self):
4         super(TestRegistroAsistenciaFaceIDBase, self).setUp()
5         # Crear un usuario administrador de prueba
6         image_path = os.path.join(os.path.dirname(__file__), '..', 'static/test', 'imagen_para_test.jpg')
7         with open(image_path, 'rb') as img_file:
8             self.administrador_data = {
9                 'name': 'Administrador Test',
10                'login': 'admin_test@example.com',
11                'image_1920': base64.b64encode(img_file.read()).decode('utf-8'),
12                'groups_id': [(6, 0, [self.env.ref('control_acceso.res_groups_administrador').id])]
13            }
14         self.administrador = self.env['res.users'].create(self.administrador_data)
15         self.reconocimiento_facial = ReconocimientoFacial() # Crea una instancia del controlador

```

Figura 103. Clase Registro Asistencia Face ID: Base

```

1 class TestRegistroAsistenciaMAC(TestRegistroAsistenciaFaceIDBase):
2
3     # Setup específico para pruebas de MAC
4     def setUp(self):
5         super().setUp()

```


Figura 104. Clase Registro Asistencia Face ID: Dirección MAC

```

1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_direccion_mac_sin_horario')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_validar_direccion_mac_sin_horario_equipo_no_habilitado(self, mock_getenv, mock_validator):
4     # Configurar para entorno de pruebas
5     mock_getenv.return_value = 'test'
6
7     # Configurar el mock para retornar un error
8     mock_validator.return_value = {'codigo': -1, 'mensaje': 'El equipo no se encuentra habilitado para el registro de asistencia.'}
9
10    # Simular una solicitud
11    resultado = self.reconocimiento_facial.validar_direccion_mac_sin_horario(mock_validator)
12
13    # Verificar el resultado (no es un http.Response en entorno de pruebas)
14    self.assertIsInstance(resultado, dict)
15    self.assertEqual(resultado['codigo'], -1)
16    self.assertEqual(resultado['mensaje'], 'El equipo no se encuentra habilitado para el registro de asistencia.')

```

Figura 105. Prueba unitaria 19 (PU19), el equipo no está habilitado para el registro de asistencia

```

1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_direccion_mac_sin_horario')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_validar_direccion_mac_sin_horario_equipo_no_registrado(self, mock_getenv, mock_validator):
4     # Configurar para entorno de pruebas
5     mock_getenv.return_value = 'test'
6
7     # Configurar el mock para retornar un error
8     mock_validator.return_value = {'codigo': -1, 'mensaje': 'El equipo no se encuentra registrado para el registro de asistencia.'}
9
10    # Simular una solicitud
11    resultado = self.reconocimiento_facial.validar_direccion_mac_sin_horario(mock_validator)
12
13    # Verificar el resultado (no es un http.Response en entorno de pruebas)
14    self.assertIsInstance(resultado, dict)
15    self.assertEqual(resultado['codigo'], -1)
16    self.assertEqual(resultado['mensaje'], 'El equipo no se encuentra registrado para el registro de asistencia.')

```

Figura 106. Prueba unitaria 20 (PU20), el equipo no está registrado para el registro de asistencia

```

1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_direccion_mac_sin_horario')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_validar_direccion_mac_sin_interfaces_de_red(self, mock_getenv, mock_validator):
4     # Configurar para entorno de pruebas
5     mock_getenv.return_value = 'test'
6
7     # Configurar el mock para retornar un error
8     mock_validator.return_value = {'codigo': -1, 'mensaje': 'No se encontraron interfaces de red.'}
9
10    # Simular una solicitud
11    resultado = self.reconocimiento_facial.validar_direccion_mac_sin_horario(mock_validator)
12
13    # Verificar el resultado (no es un http.Response en entorno de pruebas)
14    self.assertIsInstance(resultado, dict)
15    self.assertEqual(resultado['codigo'], -1)
16    self.assertEqual(resultado['mensaje'], 'No se encontraron interfaces de red.')

```

Figura 107. Prueba unitaria 21 (PU21), no se encontraron interfaces de red

```
1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_direccion_mac_sin_horario')
2 def test_validar_direccion_mac_sin_horario_exitoso(self, mock_validar):
3     # Configuramos el mock
4     mock_usuario = MagicMock()
5     mock_usuario.id = 1
6     mock_equipo = MagicMock()
7     mock_equipo.habilitado = True
8     mock_equipo.laboratorio_id = 2
9     mock_equipo.id = 3
10
11     mock_validar.return_value = {
12         'usuario_id': 1,
13         'laboratorio_id': 2,
14         'equipo_informatico_id': 3,
15         'fecha_y_hora': '2024-16-15 10:00:00',
16         'rol': 'Administrador',
17         'direccion_mac': '08:11:22:33:44:55',
18     }
19
20     # llamar a la función
21     resultado = self.reconocimiento_facial.validar_direccion_mac_sin_horario(mock_usuario)
22
23     # Verificar el resultado
24     self.assertIsInstance(resultado, dict)
25     self.assertEqual(resultado['usuario_id'], 1)
26     self.assertEqual(resultado['laboratorio_id'], 2)
27     self.assertEqual(resultado['equipo_informatico_id'], 3)
28     self.assertEqual(resultado['rol'], 'Administrador')
29     self.assertIn('fecha_y_hora', resultado)
30     self.assertIn('direccion_mac', resultado)
```

Figura 108. Prueba unitaria 22 (PU22), registro exitoso de un equipo habilitado para el registro de asistencia

```

1 class TestRegistroAsistenciaIP(TestRegistroAsistenciaFaceIDBase):
2     # Setup específico para pruebas de IP
3     def setUp(self):
4         super().setUp()
5
6         # Crear un laboratorio de prueba
7         self.laboratorio = self.env['controlacceso.lab'].create({
8             'name': '1',
9             'nombre_lab': 'Laboratorio Test',
10            'rango_ip': '192.168.1.0/24',
11            'rango_ip_habilitado': True,
12            'equipos_informaticos_habilitado': True
13        })
14
15        # Crear un equipo informático de prueba
16        self.equipo_informatico = self.env['controlacceso.equipoinformatico'].create({
17            'codigo_identificador': 'E011',
18            'direccion_mac': 'AA:BB:CC:DD:EE:FF',
19            'laboratorio_id': self.laboratorio.id,
20            'habilitado': True
21        })
22
23        # Diccionario base de asistencia para las pruebas
24        self.asistencia_dict_base = {
25            'laboratorio_id': self.laboratorio.id,
26            'usuario_id': self.administrador.id,
27            'equipo_informatico_id': self.equipo_informatico.id,
28            'fecha_y_hora': '2024-03-21 10:00:00',
29            'rol': 'Administrador',
30            'direccion_mac': '00:11:22:33:44:55'
31        }
  
```

Figura 109. Clase Registro Asistencia Face ID: Dirección IP

```

1 @patch('odo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.reconocimiento_facial.validar_direccion_ip_con_rango_ip_de_laboratorio')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_ip_fuera_de_rango_de_ip_de_laboratorio(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente está fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     # Configurar el mock para retornar un error
10    mock_validar.return_value = {'codigo': -3, 'mensaje': 'La dirección IP no pertenece al rango de IP asociada al laboratorio.'}
11
12    # Simular una solicitud
13    resultado = self.reconocimiento_facial.validar_direccion_ip_con_rango_ip_de_laboratorio(mock_validar)
14
15    # Verificar el resultado (no es un http.Response en entorno de pruebas)
16    self.assertIsInstance(resultado, dict)
17    self.assertEqual(resultado['codigo'], -3)
18    self.assertEqual(resultado['mensaje'], 'La dirección IP no pertenece al rango de IP asociada al laboratorio.')
  
```

Figura 110. Prueba unitaria 23 (PU23), la dirección IP del equipo no pertenece al laboratorio asociado para el registro de asistencia.

```

1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id.controller.ReconocimientoFacial.validar_direccion_ip_con_rango_ip_de_laboratorio')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_ip_no_se_puede_obtener_direccion_ip(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente está fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     # Configurar el mock para retornar un error
10    mock_validar.return_value = {'codigo': -1, 'mensaje': 'Error: No se puede obtener la dirección IP del usuario.'}
11
12    # Simular una solicitud
13    resultado = self.reconocimiento_facial.validar_direccion_ip_con_rango_ip_de_laboratorio(mock_validar)
14
15    # Verificar el resultado (no es un http.Response en entorno de pruebas)
16    self.assertIsInstance(resultado, dict)
17    self.assertEqual(resultado['codigo'], -1)
18    self.assertEqual(resultado['mensaje'], 'Error: No se puede obtener la dirección IP del usuario.')
    
```

Figura 111. Prueba unitaria 24 (PU24), no se puede obtener la dirección IP del equipo para el registro de asistencia.

```

1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id.controller.ReconocimientoFacial.validar_direccion_ip_con_rango_ip_de_laboratorio')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_ip_rango_ip_deshabilitado(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente está fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     self.laboratorio.write({'equipos_informaticos_habilitado': False})
10
11    # Configurar el mock para devolver el diccionario esperado con direccion_ip añadida
12    mock_result = self.asistencia_dict_base.copy()
13    mock_result['direccion_ip'] = '192.168.1.100'
14    mock_validar.return_value = mock_result
15
16    # Simular una solicitud
17    resultado = self.reconocimiento_facial.validar_direccion_ip_con_rango_ip_de_laboratorio(self.asistencia_dict_base.copy())
18
19    # Verificar que contiene todas las claves esperadas
20    claves_esperadas = ['usuario_id', 'laboratorio_id', 'equipo_informatico_id',
21                       'fecha_y_hora', 'rol', 'direccion_mac', 'direccion_ip']
22    for clave in claves_esperadas:
23        self.assertIn(clave, resultado)
24
25    # Verificar el resultado
26    self.assertIsInstance(resultado, dict)
27    self.assertEqual(resultado['usuario_id'], self.administrador.id)
28    self.assertEqual(resultado['laboratorio_id'], self.laboratorio.id)
29    self.assertEqual(resultado['equipo_informatico_id'], self.equipo_informatico.id)
30    self.assertEqual(resultado['rol'], 'Administrador')
31    self.assertIn('fecha_y_hora', resultado)
32    self.assertIn('direccion_mac', resultado)
33    self.assertIn('direccion_ip', resultado)
    
```

Figura 112. Prueba unitaria 25 (PU25), el registro de asistencia cuando el rango IP del laboratorio está deshabilitado

```

1 @patch('odo.addons.control_acceso.controllers.registro_asistencias_face_id.controller.ReconocimientoFacial.validar_direccion_ip_con_rango_ip_de_laboratorio')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_ip_direccion_ip_dentro_del_rango_ip_de_laboratorio(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente este fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     # Configurar el mock para devolver el diccionario esperado con direccion ip añadida
10    mock_result = self.asistencia_dict_base.copy()
11    mock_result['direccion_ip'] = '192.168.1.100'
12    mock_validar.return_value = mock_result
13
14    # Simular una solicitud
15    resultado = self.reconocimiento_facial.validar_direccion_ip_con_rango_ip_de_laboratorio(self.asistencia_dict_base.copy())
16
17    # Verificar que contenga todas las claves esperadas
18    claves_esperadas = ['usuario_id', 'laboratorio_id', 'equipo_informatico_id',
19                       'fecha_y_hora', 'rol', 'direccion_mac', 'direccion_ip']
20    for clave in claves_esperadas:
21        self.assertIn(clave, resultado)
22
23    # Verificar el resultado
24    self.assertIsInstance(resultado, dict)
25    self.assertEqual(resultado['usuario_id'], self.administrador.id)
26    self.assertEqual(resultado['laboratorio_id'], self.laboratorio.id)
27    self.assertEqual(resultado['equipo_informatico_id'], self.equipo_informatico.id)
28    self.assertEqual(resultado['rol'], 'Administrador')
29    self.assertIn('fecha_y_hora', resultado)
30    self.assertIn('direccion_mac', resultado)
31    self.assertIn('direccion_ip', resultado)

```

Figura 113. Prueba unitaria 26 (PU26), el registro de asistencia cuando la dirección IP del equipo se encuentra dentro del rango IP del laboratorio.

```

1 class TestRegistroAsistenciaHorario(TestRegistroAsistenciaFaceIDBase):
2
3     def setUp(self):
4         super().setUp()

```

Figura 114. Clase Registro Asistencia Face ID: Validar Horario

```
1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_horario')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_validar_horario_estudiante_sin_horario(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente está fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     # Mock para obtener hora local y convertir fecha hora
10    self.reconocimiento_facial_obtener_hora_local = MagicMock(return_value="2023-10-16 10:00:00")
11
12    # Configurar el mock para retornar un error
13    mock_validar.return_value = {'codigo': -1, 'mensaje': 'No posee ningún horario acorde al día y hora actual.'}
14
15    # Simular una solicitud
16    resultado = self.reconocimiento_facial.validar_horario(mock_validar)
17
18    # Verificar el resultado (no es un http.Response en entorno de pruebas)
19    self.assertIsInstance(resultado, dict)
20    self.assertEqual(resultado['codigo'], -1)
21    self.assertEqual(resultado['mensaje'], 'No posee ningún horario acorde al día y hora actual.')
```

Figura 115. Prueba unitaria 27 (PU27), rechazo de registro por falta de horario de estudiante.

```
1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_horario')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_validar_horario_estudiante_asistencia_ya_registrada(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente está fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     # Mock para obtener hora local y convertir fecha hora
10    self.reconocimiento_facial_obtener_hora_local = MagicMock(return_value="2023-10-16 10:00:00")
11
12    # Configurar el mock para retornar un error
13    mock_validar.return_value = {'codigo': -1, 'mensaje': 'Ya ha marcado asistencia para este horario hoy.'}
14
15    # Simular una solicitud
16    resultado = self.reconocimiento_facial.validar_horario(mock_validar)
17
18    # Verificar el resultado (no es un http.Response en entorno de pruebas)
19    self.assertIsInstance(resultado, dict)
20    self.assertEqual(resultado['codigo'], -1)
21    self.assertEqual(resultado['mensaje'], 'Ya ha marcado asistencia para este horario hoy.')
```

Figura 116. Prueba unitaria 28 (PU28), rechazo de asistencia ya registrada por estudiante.

```

1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_horario')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_validar_horario_estudiante_exitoso(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente está fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     # Mock para _obtener_hora_local y convertir_fecha_hora
10    self.reconocimiento_facial._obtener_hora_local = MagicMock(return_value="2023-10-16 10:00:00")
11
12    # Configurar el mock para retornar un error
13    mock_validar.return_value = {
14        'usuario_id': 7,
15        'responsable': 1,
16        'curso_id': 1,
17        'laboratorio_id': 1,
18        'carrera_id': 1,
19        'periodo_id': 2,
20        'materia_id': 2,
21        'fecha_y_hora': '2024-10-01 10:20:23',
22        'rol': 'Estudiante',
23    }
24
25    # Simular una solicitud
26    resultado = self.reconocimiento_facial.validar_horario(mock_validar)
27
28    # Verificar el resultado (no es un http.Response en entorno de pruebas)
29    self.assertIsInstance(resultado, dict)
30    self.assertEqual(resultado['usuario_id'], 7)
31    self.assertEqual(resultado['responsable'], 1)
32    self.assertEqual(resultado['curso_id'], 1)
33    self.assertEqual(resultado['laboratorio_id'], 1)
34    self.assertEqual(resultado['carrera_id'], 1)
35    self.assertEqual(resultado['periodo_id'], 2)
36    self.assertEqual(resultado['materia_id'], 2)
37    self.assertIn('fecha_y_hora', resultado)
38    self.assertEqual(resultado['rol'], 'Estudiante')

```

Figura 117. Prueba unitaria 29 (PU29), registro exitoso de estudiante con horario correcto.

```

1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_horario')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_validar_horario_docente_sin_horario(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente está fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     # Mock para _obtener_hora_local y convertir_fecha_hora
10    self.reconocimiento_facial._obtener_hora_local = MagicMock(return_value="2023-10-16 10:00:00")
11
12    # Configurar el mock para retornar un error
13    mock_validar.return_value = {'codigo': -1, 'mensaje': 'No posee ningún horario acorde al día y hora actual.'}
14
15    # Simular una solicitud
16    resultado = self.reconocimiento_facial.validar_horario(mock_validar)
17
18    # Verificar el resultado (no es un http.Response en entorno de pruebas)
19    self.assertIsInstance(resultado, dict)
20    self.assertEqual(resultado['codigo'], -1)
21    self.assertEqual(resultado['mensaje'], 'No posee ningún horario acorde al día y hora actual.')

```

Figura 118. Prueba unitaria 30 (PU30), rechazo de registro por falta de horario de docente.

```

1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_horario')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_validar_horario_docente_asistencia_ya_registrada(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente está fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     # Mock para obtener hora local y convertir fecha hora
10    self.reconocimiento_facial._obtener_hora_local = MagicMock(return_value="2023-10-16 10:00:00")
11
12    # Configurar el mock para retornar un error
13    mock_validar.return_value = {'codigo': -1, 'mensaje': 'Ya ha marcado asistencia para este horario hoy.'}
14
15    # Simular una solicitud
16    resultado = self.reconocimiento_facial.validar_horario(mock_validar)
17
18    # Verificar el resultado (no es un http.Response en entorno de pruebas)
19    self.assertIsInstance(resultado, dict)
20    self.assertEqual(resultado['codigo'], -1)
21    self.assertEqual(resultado['mensaje'], 'Ya ha marcado asistencia para este horario hoy.')

```

Figura 119. Prueba unitaria 31 (PU31), rechazo de asistencia ya registrada por docente

```

1 @patch('odoo.addons.control_acceso.controllers.registro_asistencias_face_id_controller.ReconocimientoFacial.validar_horario')
2 @patch('os.getenv') # Simular la variable de entorno
3 def test_validar_horario_docente_exitoso(self, mock_getenv, mock_validar):
4     """Prueba cuando la IP del cliente está fuera del rango del laboratorio"""
5
6     # Configurar para entorno de pruebas
7     mock_getenv.return_value = 'test'
8
9     # Mock para obtener hora local y convertir fecha hora
10    self.reconocimiento_facial._obtener_hora_local = MagicMock(return_value="2023-10-16 10:00:00")
11
12    # Configurar el mock para retornar un error
13    mock_validar.return_value = {
14        'usuario_id': 7,
15        'responsable': 1,
16        'curso_id': 1,
17        'laboratorio_id': 1,
18        'carrera_id': 1,
19        'periodo_id': 2,
20        'materia_id': 2,
21        'fecha_y_hora': '2024-10-01 10:20:23',
22        'rol': 'Docente',
23    }
24
25    # Simular una solicitud
26    resultado = self.reconocimiento_facial.validar_horario(mock_validar)
27
28    # Verificar el resultado (no es un http.Response en entorno de pruebas)
29    self.assertIsInstance(resultado, dict)
30    self.assertEqual(resultado['usuario_id'], 7)
31    self.assertEqual(resultado['responsable'], 1)
32    self.assertEqual(resultado['curso_id'], 1)
33    self.assertEqual(resultado['laboratorio_id'], 1)
34    self.assertEqual(resultado['carrera_id'], 1)
35    self.assertEqual(resultado['periodo_id'], 2)
36    self.assertEqual(resultado['materia_id'], 2)
37    self.assertIn('fecha_y_hora', resultado)
38    self.assertEqual(resultado['rol'], 'Docente')

```

Figura 120. Prueba unitaria 32 (PU32), registro exitoso de docente con horario correcto.

A continuación, se presentan los resultados obtenidos de las pruebas unitarias en la cuarta iteración.

```

2024-10-01 15:35:46,481 7892 INFO odoo tests odoo.modules.loading: Module control acceso: Loading demo
2024-10-01 15:35:46,505 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo docente asistencial ya registrada ...
2024-10-01 15:35:46,999 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo docente exitoso ...
2024-10-01 15:35:47,380 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo docente sin horario ...
2024-10-01 15:35:47,681 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo estudiante asistencial ya registrada ...
2024-10-01 15:35:48,053 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo estudiante exitoso ...
2024-10-01 15:35:48,443 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_direccion_ip_de
ntro_del_rango_ip_de_laboratorio ...
2024-10-01 15:35:49,322 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:49,398 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_fuera_de_rango
_de_ip_de_laboratorio ...
2024-10-01 15:35:49,945 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:49,953 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_no_disponible ...
...
2024-10-01 15:35:50,324 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:50,335 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_no_se_puede_obt
ener_direccion_ip ...
2024-10-01 15:35:50,670 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:50,678 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_rango_ip_deshab
ilitado ...
2024-10-01 15:35:51,185 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:51,193 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIAMAC.test_validar_direccion
_mac_sin_horario_equipo_no_habilitado ...
2024-10-01 15:35:51,478 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIAMAC.test_validar_direccion
_mac_sin_horario_equipo_no_registro ...
2024-10-01 15:35:51,821 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIAMAC.test_validar_direccion
_mac_sin_horario_exitoso ...
2024-10-01 15:35:52,189 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIAMAC.test_validar_direccion
_mac_sin_interfaces_de_red ...
2024-10-01 15:35:52,694 7892 INFO odoo tests odoo.modules.loading: Module control acceso loaded in 16.06s (incl. 6.07s test), 1013 queries (+914 test)
2024-10-01 15:35:52,694 7892 INFO odoo tests odoo.modules.loading: 31 modules loaded in 17.51s, 1013 queries (+914 extra)
2024-10-01 15:35:53,539 7892 INFO odoo tests odoo.modules.loading: Modules loaded.
2024-10-01 15:35:53,542 7892 INFO odoo tests odoo.modules.registry: Registry loaded in 20.406s
2024-10-01 15:35:53,543 7892 INFO odoo tests odoo.service.server: Loading test file /opt/odoo/integracion/control_acceso/face_recognition/control_acceso/tests/test_regis
tro_asistencia_face_id.py
2024-10-01 15:35:53,546 7892 INFO odoo tests odoo.addons.base.models.ir.http: Generating routing map for key None
2024-10-01 15:35:53,551 7892 INFO odoo tests odoo.addons.base.models.ir.http: Generating routing map for key None
2024-10-01 15:35:54,090 7892 INFO odoo tests odoo.service.server: Running tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id.
2024-10-01 15:35:54,197 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo docente asistencial ya registrada ...
2024-10-01 15:35:54,270 7892 INFO odoo tests odoo.http: Bus Bug: Not available in test mode
2024-10-01 15:35:54,284 7892 INFO odoo tests werkzeug: 127.0.0.1 - - [01/Oct/2024 15:35:54] "POST /longpolling/poll HTTP/1.1" 200 - 6 0.008 10.023
2024-10-01 15:35:54,288 7892 INFO odoo tests odoo.http: Bus: Bin not available in test mode

```

Figura 121. Ejecución de las pruebas unitarias de la cuarta iteración, parte 1.


```

2024-10-01 15:35:54,296 7892 INFO odoo tests werkzeug: 127.0.0.1 - - [01/Oct/2024 15:35:54] "POST /longpolling/poll HTTP/1.1" 200 - 5 0.005 11.631
2024-10-01 15:35:54,590 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo docente exitoso ...
2024-10-01 15:35:54,918 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo docente sin horario ...
2024-10-01 15:35:55,285 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo estudiante asistencial ya registrada ...
2024-10-01 15:35:55,659 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo estudiante exitoso ...
2024-10-01 15:35:56,030 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaHorario.test_validar_hora
lo estudiante sin horario ...
2024-10-01 15:35:56,506 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_direccion_ip_de
ntro_del_rango_ip_de_laboratorio ...
2024-10-01 15:35:56,594 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:56,968 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_fuera_de_rango
_de_ip_de_laboratorio ...
2024-10-01 15:35:57,332 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:57,341 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_no_disponible ...
...
2024-10-01 15:35:57,677 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:57,685 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_no_se_puede_obt
ener_direccion_ip ...
2024-10-01 15:35:58,043 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:58,054 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIP.test_ip_rango_ip_deshab
ilitado ...
2024-10-01 15:35:58,432 7892 INFO odoo tests odoo.addons.control_acceso.models.laboratorio: 192.168.1.0/24
2024-10-01 15:35:58,463 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIAMAC.test_validar_direccion
_mac_sin_horario_equipo_no_habilitado ...
2024-10-01 15:35:58,845 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIAMAC.test_validar_direccion
_mac_sin_horario_equipo_no_registro ...
2024-10-01 15:35:59,207 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIAMAC.test_validar_direccion
_mac_sin_horario_exitoso ...
2024-10-01 15:35:59,570 7892 INFO odoo tests odoo.addons.control_acceso.tests.test_registro_asistencia_face_id: Starting TestRegistroAsistenciaIAMAC.test_validar_direccion
_mac_sin_interfaces_de_red ...
2024-10-01 15:35:59,952 7892 INFO odoo tests odoo.service.server: Starting post tests
2024-10-01 15:35:59,950 7892 INFO odoo tests odoo.service.server: 0 post tests in 0.06s, 0 queries
2024-10-01 15:35:59,950 7892 INFO odoo tests odoo.tests.runner: 0 failed, 0 error(s) of 39 tests when loading database 'odoo tests'
2024-10-01 15:35:59,950 7892 INFO odoo tests odoo.service.server: Initiating shutdown
2024-10-01 15:35:59,957 7892 INFO odoo tests odoo.service.server: Hit CTRL-C again or send a second signal to force the shutdown.
2024-10-01 15:36:00,208 7892 INFO odoo tests odoo.sql_db: ConnectionPool(used=0/count=nan/s4): Closed 2 connections

```

Figura 122. Ejecución de las pruebas unitarias de la cuarta iteración, parte 2.

Para constancia y validación de la información expuesta, firma a continuación el tutor del Trabajo de Integración Curricular, en su calidad de docente guía en el desarrollo del proyecto:

Ing. Edison Leonardo Coronel Romero, Mg.Sc.	 Firma electrónica por: EDISON LEONARDO CORONEL ROMERO
---	--

ANEXO 5: Resultados de la aceptación tecnológica del sistema mediante TAM



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

• **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

Descripción: El presente anexo muestra los resultados de la evaluación de la aceptación tecnológica del sistema, utilizando el Modelo TAM (Technology Acceptance Model). Se incluye el cuestionario elaborado para medir la percepción de los usuarios en relación con los constructos clave del modelo: utilidad percibida, facilidad de uso percibida, actitud hacia el uso y la intención de uso. Además, se adjunta un fragmento de las respuestas obtenidas, el cual demuestra que los participantes en la evaluación cuentan con correos institucionales de la Universidad Nacional de Loja. También se presentan capturas de los resultados generados por Google Forms, detallando la percepción del usuario respecto a cada pregunta, lo que proporciona una visión clara sobre la aceptación tecnológica del sistema por parte de los usuarios finales.

RESULTADOS DE LA ACEPTACIÓN TECNOLÓGICA DEL SISTEMA MEDIANTE EL MODELO TAM

1 Preguntas

La **Tabla 1** presenta el cuestionario elaborado para determinar la percepción del usuario sobre el sistema. Cada constructo (variable del Modelo de Aceptación Tecnológica, TAM) contiene un conjunto de preguntas que permitirán asignar ponderaciones basadas en las respuestas obtenidas para cada uno de ellos.

Tabla 1. Cuestionario para evaluar el nivel de aceptación tecnológica (TAM)

Constructo	Identificador	Preguntas
Utilidad percibida (UP)	UP1	¿Está de acuerdo que el uso del registro de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software mejora la gestión y control de los usuarios que hacen uso de dicha instalación?
	UP2	¿Está de acuerdo que el proceso de registro de asistencia en el laboratorio mediante reconocimiento facial y detección de vida es más sencillo y práctico de realizar que los métodos manuales que se llevan a cabo actualmente?
	UP3	¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio es realmente útil para controlar las asistencias de los usuarios que hacen uso del laboratorio de software?
Facilidad de uso percibida (FUP)	FUP1	¿Está de acuerdo que es fácil de utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software?
	FUP2	¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio reduce el tiempo necesario para el registro de asistencias?
	FUP3	¿Está de acuerdo que el sistema web que gestiona las asistencias y reportes en el laboratorio de software es fácil de navegar y encontrar la información necesaria?
Actitud hacia el uso (AU)	AU1	¿Se siente cómodo/a al utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software?
	AU2	¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es beneficioso para gestionar y controlar los usuarios que hacen uso del laboratorio de software?
	AU3	¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software es beneficioso para gestionar y controlar los usuarios que hacen uso del laboratorio de software?

		vida en el laboratorio de software, es una buena idea en general?
	AU4	¿Considera agradable el uso del sistema web que gestiona las asistencias y reportes en el laboratorio de software?
	AU5	¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es una opción viable y conveniente para el registro de asistencia en el laboratorio de software en comparación con el método que se lleva actualmente?
Intención de uso (IU)	IU1	¿Tiene usted la intención de utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida dentro del laboratorio de software?
	IU2	¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es una opción beneficiosa en el futuro, considerándola como una alternativa viable en el registro de asistencias dentro del laboratorio de software?
	IU3	¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida mejorará la experiencia de mantener el control de las asistencias en el laboratorio de software?

2 Resultados

La **Figura 1** presenta un fragmento de los correos institucionales de las respuestas obtenidas en la encuesta realizada mediante Google Forms. Esto, se hace con la finalidad de evidenciar la participación estudiantil de la carrera de Computación de la Universidad Nacional de Loja, garantizando así la autenticidad de las respuestas obtenidas.

Usuarios que han respondido

Correo electrónico

melissa.tuza@unl.edu.ec

juan.d.carreno@unl.edu.ec

james.romero@unl.edu.ec

alba.nieto@unl.edu.ec

jose.cartuche.r@unl.edu.ec

juan.f.castillo.e@unl.edu.ec

juan.p.ortega@unl.edu.ec

jainer.pinta@unl.edu.ec

Figura 1. Fragmento de las respuestas obtenidas

A continuación, se presentan los resultados obtenidos para cada pregunta de cada constructo.

2.1 Utilidad Percibida (UP)

¿Está de acuerdo que el uso del registro de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software mejora la ges...e los usuarios que hacen uso de dicha instalación?

42 respuestas

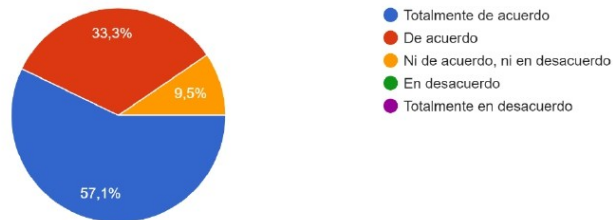


Figura 2. Pregunta UP1

¿Está de acuerdo que el proceso de registro de asistencia en el laboratorio mediante reconocimiento facial y detección de vida es más ...odos manuales que se llevan a cabo actualmente?
42 respuestas

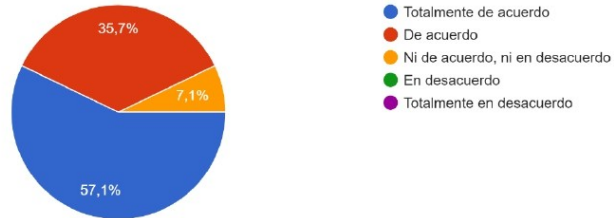


Figura 3. Pregunta UP2

¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio es realmente ...uarios que hacen uso del laboratorio de software?
42 respuestas

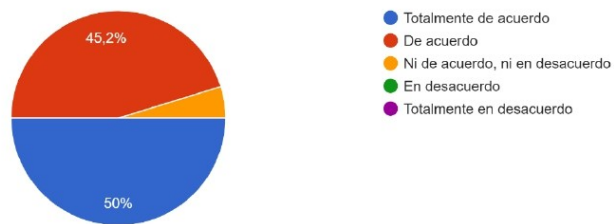


Figura 4. Pregunta UP3

2.2 Facilidad de uso percibida (FUP)

¿Está de acuerdo que es fácil de utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software?

42 respuestas

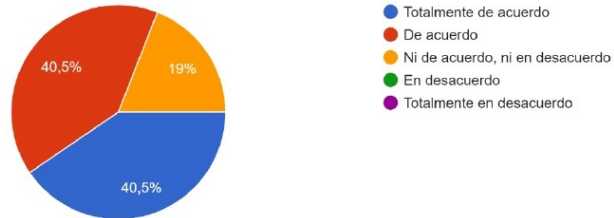


Figura 5. Pregunta FUP1

¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el lab... tiempo necesario para el registro de asistencias?

42 respuestas

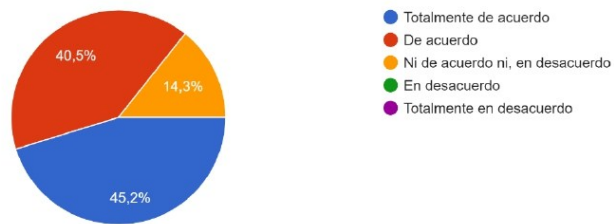


Figura 6. Pregunta FUP2

¿Está de acuerdo que el sistema web que gestiona las asistencias y reportes en el laboratorio de software es fácil de navegar y encontrar la información necesaria?

42 respuestas

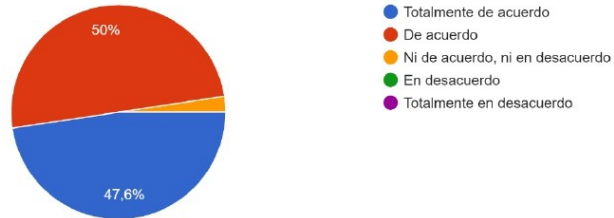


Figura 7. Pregunta FUP3

2.3 Actitud hacia el uso (AU)

¿Se siente cómodo/a al utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software?

42 respuestas

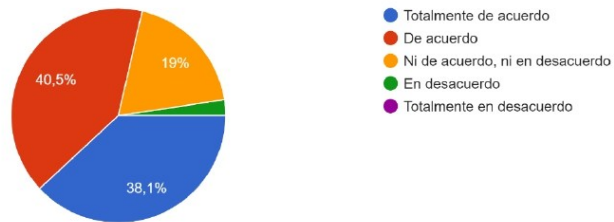


Figura 8. Pregunta AU1

¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es benef...uarios que hacen uso del laboratorio de software?
42 respuestas

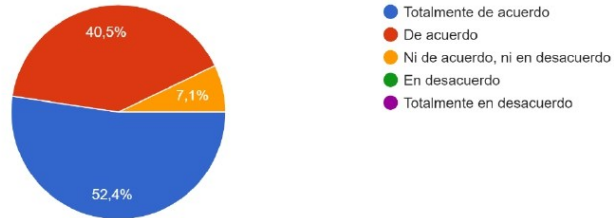


Figura 9. Pregunta AU2

¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida en el laboratorio de software, es una buena idea en general?
42 respuestas

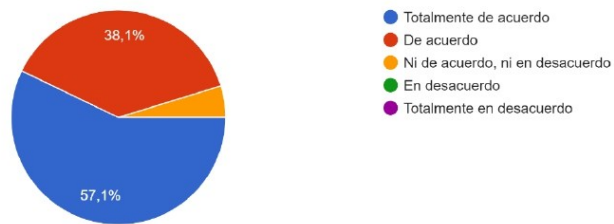


Figura 10. Pregunta AU3

¿Considera agradable el uso del sistema web que gestiona las asistencias y reportes en el laboratorio de software?

42 respuestas

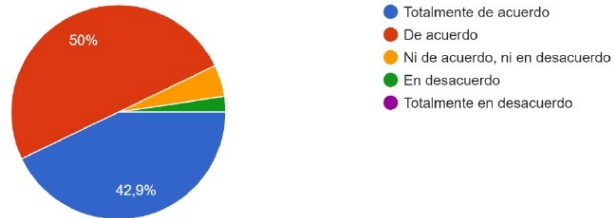


Figura 11. Pregunta AU4

¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es una opción viable y conveniente en comparación con el método que se lleva actualmente?

42 respuestas

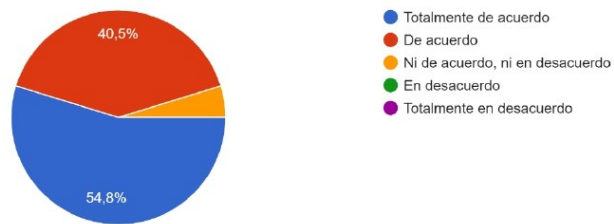


Figura 12. Pregunta AU5

2.4 Intención de uso (IU)

¿Tiene usted la intención de utilizar el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida dentro del laboratorio de software?

42 respuestas

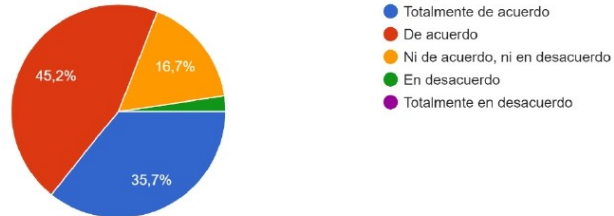


Figura 13. Pregunta IU1

¿Está de acuerdo que el uso del sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida es una op...e asistencias dentro del laboratorio de software?

42 respuestas

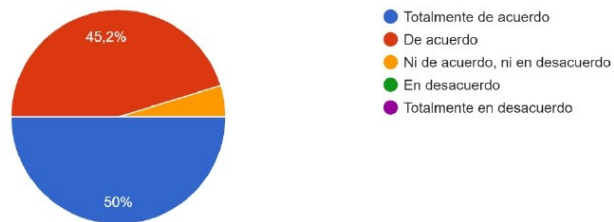


Figura 14. Pregunta IU2

¿Está de acuerdo que el sistema web para la gestión de asistencias mediante reconocimiento facial y detección de vida mejorará la experiencia de mant... de las asistencias en el laboratorio de software?
42 respuestas

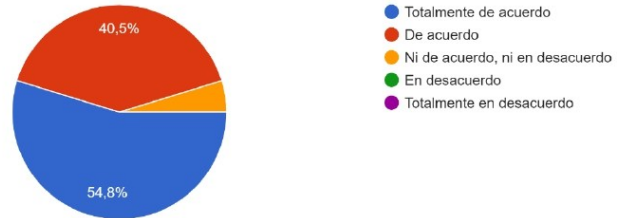


Figura 15. Pregunta IU3

ANEXO 6: Manual de usuario para administrador



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- Elaborado por:
 - Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).
- Revisado y aprobado por:
 - Ing. Edison Leonardo Coronel Romero, Mg.Sc., tutor del Trabajo de Integración Curricular.

Descripción: El presente anexo contiene el manual de usuario dirigido al administrador del sistema. Dicho manual detalla las funcionalidades a las que el administrador tiene acceso, organizadas por secciones. Además del registro de asistencia mediante reconocimiento facial, sin validación de horario, el administrador puede gestionar usuarios, laboratorios y equipos informáticos. También se describe cómo puede visualizar y administrar los registros de asistencia de todos los usuarios del sistema, incluidos otros administradores, proporcionando una guía clara para el uso efectivo de sus herramientas.

MANUAL DE USUARIO PARA ADMINISTRADOR

Es necesario mencionar que este manual de usuario se centra en todas las funcionalidades nuevas (registro de asistencia face ID, gestión de equipos informáticos, gestión de imagen de usuario, gestión de laboratorios con dirección IP y equipos informáticos), más no en las funcionalidades existentes (gestión de carreras, gestión de materias, gestión de cursos, etc.) del módulo de Control de Acceso dirigido al administrador.

1 Ingreso a Sistema

Se debe ingresar a la siguiente dirección web: [https://computacion.unl.edu.ec/web].

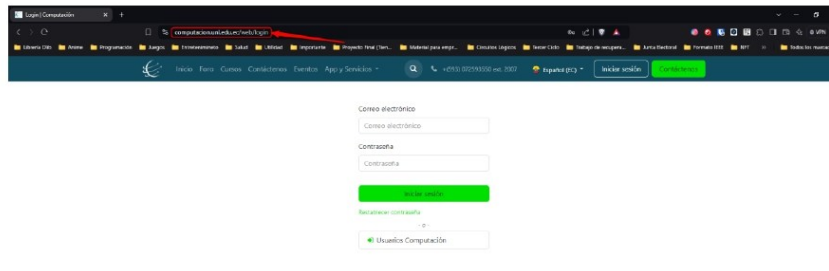


Figura 1. Accediendo al sistema mediante el navegador web

Se debe iniciar sesión con las respectivas credenciales del administrador.

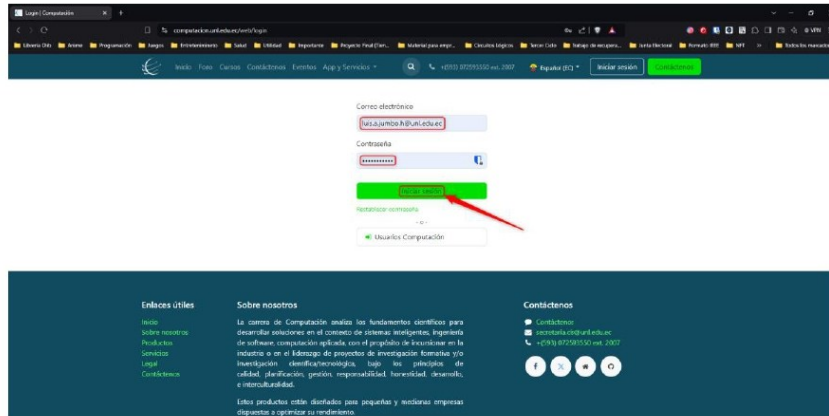


Figura 2. Inicio de sesión

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].



Figura 3. Accediendo al módulo de Control de acceso.

Una vez realizado estos pasos, ya se puede hacer uso del sistema de [Control de Acceso] como administrador.



Figura 4. Vista del módulo de Control de acceso

2 Gestión de Usuarios

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Ajustes].

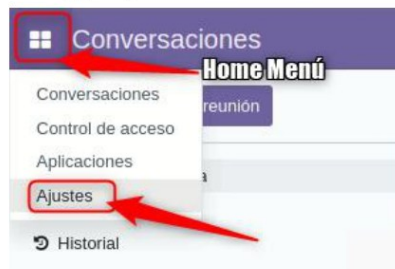


Figura 5. Accediendo al módulo de Ajustes.

Se selecciona [Administrar usuarios].

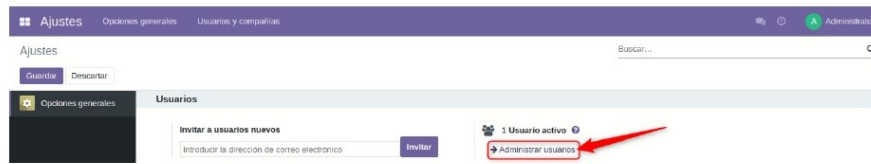


Figura 6. Accediendo a la sección de administrar usuarios.

Una vez realizado estos pasos, ya se puede gestionar los usuarios del sistema.



Figura 7. Vista de Usuarios.

2.1 Crear Usuario

Se selecciona [Crear].

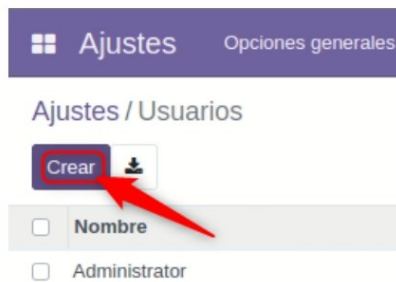


Figura 8. Accediendo a crear un nuevo usuario.

Se procede a llenar los datos informativos del usuario (nombre, correo electrónico e imagen). Tener en cuenta que todos los datos son obligatorios, especialmente la imagen de usuario (necesaria para poder realizar el reconocimiento facial posteriormente). Dicha imagen de usuario debe contener un rostro humano (en caso de que no se detecte un rostro humano, o exista más de un rostro, el sistema no permitirá el registro del usuario), y tanto el nombre como el correo electrónico deben de ser únicos para cada usuario. Además, en la opción [Control de acceso], se debe seleccionar el tipo de usuario (rol) que tendrá el usuario en el sistema [Control de acceso]. Se debe tomar en consideración que, al momento de crear el usuario, también se almacena la imagen de usuario en el servidor. Una vez completo todo

el formulario, se selecciona [Grabar]. Por otro lado, se puede cancelar el proceso de creación del usuario al seleccionar [Descartar].

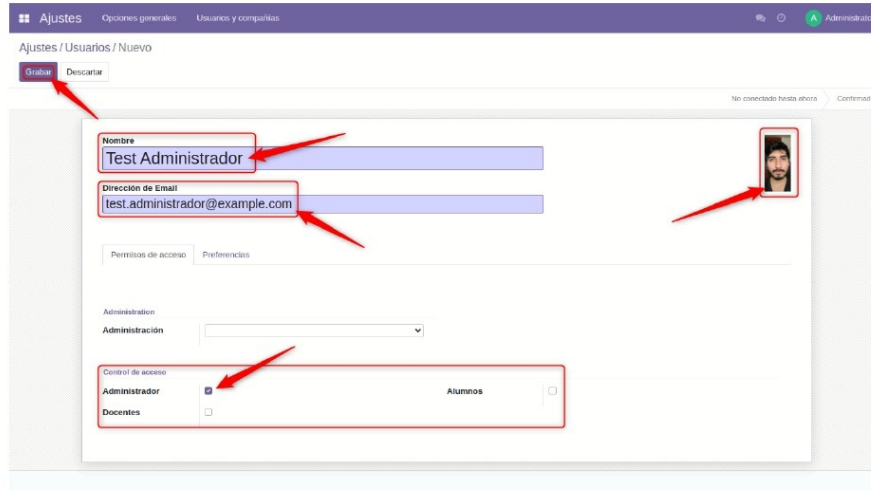


Figura 9. Creación de un nuevo usuario.

Una vez realizado estos pasos, ya se puede observar el nuevo usuario registrado en el sistema.



Nombre	Usuario	Idioma	Última autenticación
Administrador	administrador	Spanish (EC) / Español (EC)	17/10/2024 08:59:24
Test Administrador	test.administrador@example.com	Spanish (EC) / Español (EC)	

Figura 10. Lista de usuarios con el nuevo usuario registrado en el sistema.

2.2 Editar Usuario

Se selecciona un usuario.



Nombre	Usuario	Idioma	Última autenticación
Administrador	administrador	Spanish (EC) / Español (EC)	17/10/2024 08:59:24
Test A. Editor	test.a.editor@example.com	Spanish (EC) / Español (EC)	
Test Administrador	test.administrador@example.com	Spanish (EC) / Español (EC)	

Figura 11. Seleccionando un usuario existente.

Luego se selecciona [Editar].

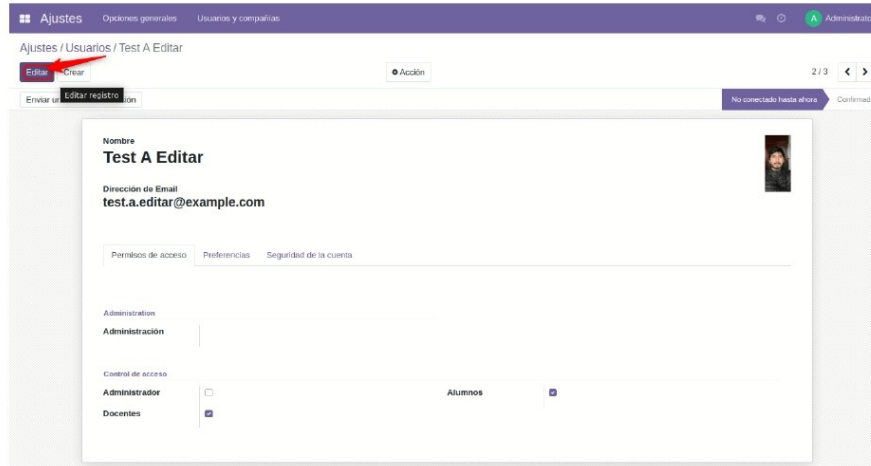
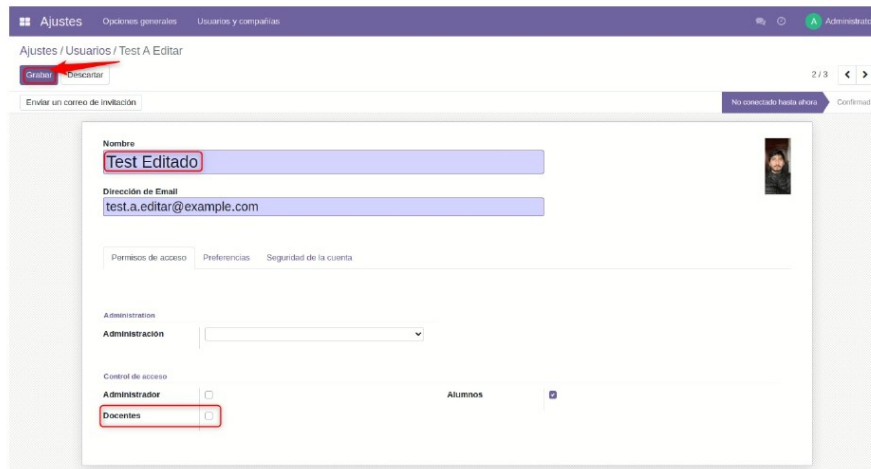


Figura 12. Accediendo a editar un usuario existente.

Se procede a llenar los datos informativos del usuario (nombre, correo electrónico e imagen). Tener en cuenta que todos los datos son obligatorios, especialmente la imagen de usuario (necesaria para poder realizar el reconocimiento facial posteriormente). Dicha imagen de usuario debe contener un rostro humano (en caso de que no se detecte un rostro humano, o exista más de un rostro, el sistema no permitirá la edición del usuario), y tanto el nombre como el correo electrónico deben de ser únicos para cada usuario. Además, en la opción [Control de acceso], se debe seleccionar el tipo de usuario (rol) que tendrá el usuario en el sistema [Control de acceso]. Se debe tomar en consideración que, al momento de editar el usuario, si se editó la imagen de usuario, esta también se sobrescribirá en el servidor. Una vez completo todo el formulario, se selecciona [Grabar]. Por otro lado, se puede cancelar el proceso de edición del usuario al seleccionar [Descartar].



Ajustes / Usuarios / Test A Editar

Crear Descartar

Enviar un correo de invitación No conectado hasta ahora Confirmado

Nombre: Test Editado

Dirección de Email: test.a.editar@example.com

Permisos de acceso: Preferencias Seguridad de la cuenta

Administración: Administración

Control de acceso: Administrador Alumnos Docentes

Figura 13. Edición de un usuario existente.

Una vez realizado estos pasos, ya se puede observar el usuario editado en el sistema.



Ajustes / Usuarios

Usuarios internos

Nombre	Usuario	Idioma	Última autenticación
<input type="checkbox"/> Administrador	administrador	Spanish (EC) / Español (EC)	17/10/2024 08:59:24
<input type="checkbox"/> Test Administrador	test.administrador@example.com	Spanish (EC) / Español (EC)	
<input checked="" type="checkbox"/> Test Editado	test.a.editar@example.com	Spanish (EC) / Español (EC)	

Figura 14. Lista de usuarios con el usuario editado en el sistema.

2.3 Dar de Baja Usuario

Se selecciona uno o más usuarios.



Ajustes / Usuarios

Usuarios internos

Crear 1 seleccionado Acción

Nombre	Usuario	Idioma	Última autenticación
<input type="checkbox"/> Administrador	administrador	Spanish (EC) / Español (EC)	17/10/2024 08:59:24
<input checked="" type="checkbox"/> Test A Eliminar	test.a.eliminar@example.com	Spanish (EC) / Español (EC)	
<input type="checkbox"/> Test Administrador	test.administrador@example.com	Spanish (EC) / Español (EC)	
<input type="checkbox"/> Test Editado	test.a.editar@example.com	Spanish (EC) / Español (EC)	

Figura 15. Seleccionando uno o más usuarios existentes.

Se selecciona [Acción] y luego [Suprimir].

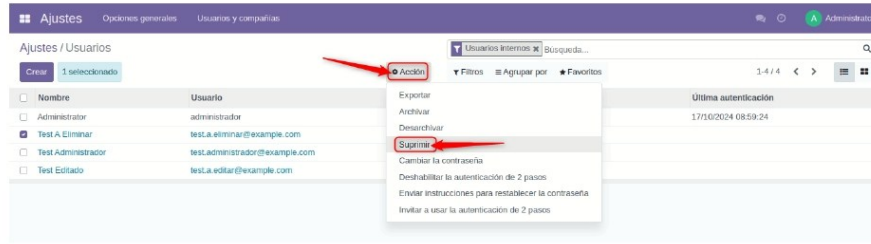


Figura 16. Accediendo a dar de baja uno o más usuarios existentes.

En la ventana emergente, se selecciona [Aceptar] para eliminar permanentemente el usuario, o [Cancelar] para abortar la eliminación del usuario. Se debe tomar en consideración que, al momento de dar de baja el usuario, la imagen de usuario también se eliminará permanentemente del servidor.

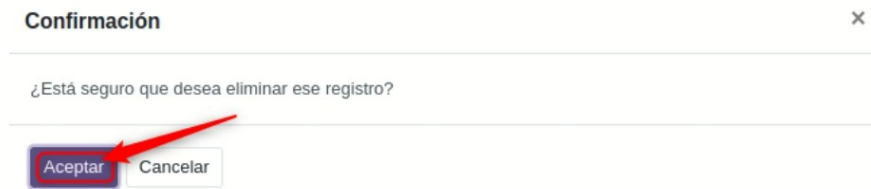


Figura 17. Eliminación de uno o más usuarios existentes.

Una vez realizado estos pasos, ya se puede observar los usuarios sin el/los usuarios dados de bajas del sistema.



Figura 18. Lista de laboratorios sin el/los usuarios dados de baja del sistema.

3 Gestión de Laboratorios

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].



Figura 19. Accediendo al módulo de Control de acceso.

Se selecciona [Gestionar Laboratorios].

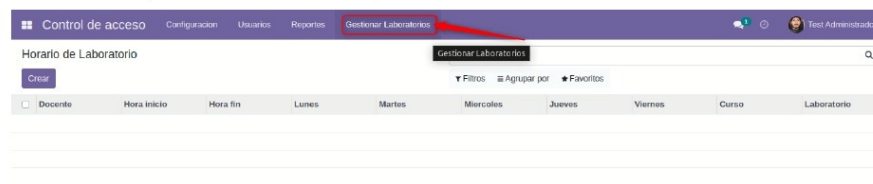


Figura 20. Accediendo a la sección de gestionar laboratorios.

Se selecciona [Laboratorios].

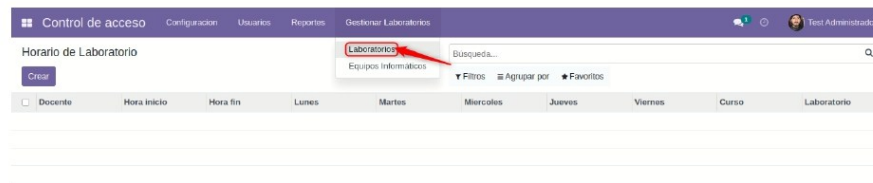


Figura 21. Accediendo a la subsección de laboratorios.

Una vez realizado estos pasos, ya se puede gestionar los laboratorios del sistema.

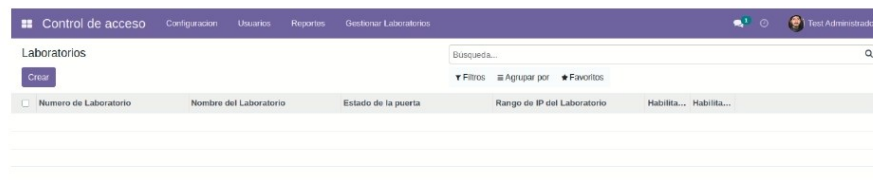


Figura 22. Vista de laboratorios.

3.1 Crear Laboratorio

Se selecciona [Crear].



Figura 23. Accediendo a crear un nuevo laboratorio.

Se procede a llenar los datos informativos del laboratorio (número, nombre, rango IP, habilitar rango IP y habilitar equipos informáticos). Tener en cuenta que todos los datos son obligatorios. El nombre del laboratorio debe ser único. El sistema validará el rango de IP ingresado junto con su máscara de red. En caso que se detecte datos ingresados no válidos, el sistema no permitirá el registro del laboratorio. Una vez completo todo el formulario, se selecciona [Grabar]. Por otro lado, se puede cancelar el proceso de creación del laboratorio al seleccionar [Descartar].

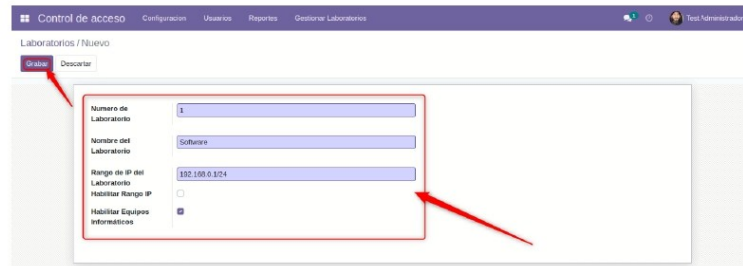


Figura 24. Creación de un nuevo laboratorio.

Una vez realizado estos pasos, ya se puede observar el nuevo laboratorio registrado en el sistema.



Figura 25. Lista de laboratorios con el nuevo laboratorio registrado en el sistema.

3.2 Editar Laboratorio

Se selecciona un laboratorio.



Figura 26. Seleccionando un laboratorio existente.

Luego se selecciona [Editar].

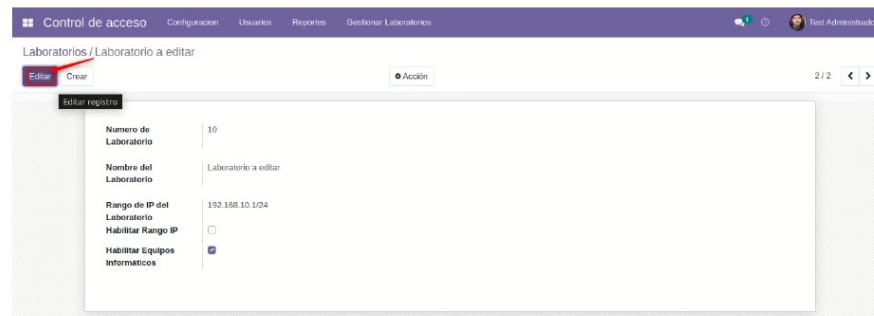


Figura 27. Accediendo a editar un laboratorio existente.

Se procede a llenar los datos informativos del laboratorio (número, nombre, rango IP, habilitar rango IP y habilitar equipos informáticos). Tener en cuenta que todos los datos son obligatorios. El nombre del laboratorio debe ser único. El sistema validará el rango de IP ingresado junto con su máscara de red. En caso que se detecte datos ingresados no válidos, el sistema no permitirá la edición del laboratorio. Una vez completo todo el formulario, se selecciona [Grabar]. Por otro lado, se puede cancelar el proceso de edición del laboratorio al seleccionar [Descartar].

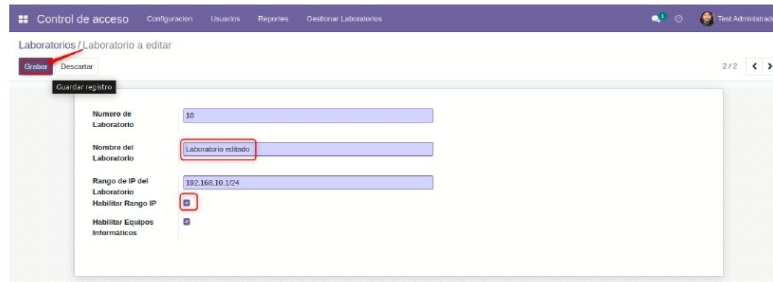


Figura 28. Edición de un laboratorio existente.

Una vez realizado estos pasos, ya se puede observar el laboratorio editado en el sistema.



Numero de Laboratorio	Nombre del Laboratorio	Estado de la puerta	Rango de IP del Laboratorio	Habilitar Rango IP	Habilitar Equipos Informaticos
1	Software	cerrado	192.168.0.1/24	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10	Laboratorio editado	cerrado	192.168.10.1/24	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 29. Lista de laboratorios con el laboratorio editado en el sistema.

3.3 Dar de Baja Laboratorio

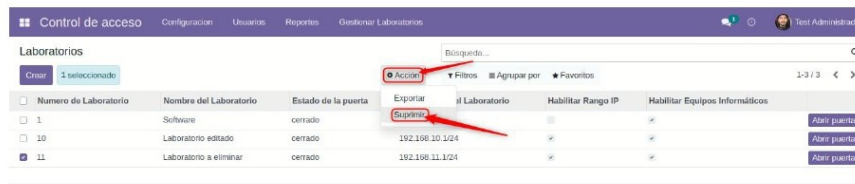
Se selecciona uno o más laboratorios.



Numero de Laboratorio	Nombre del Laboratorio	Estado de la puerta	Rango de IP del Laboratorio	Habilitar Rango IP	Habilitar Equipos Informaticos
1	Software	cerrado	192.168.0.1/24	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10	Laboratorio editado	cerrado	192.168.10.1/24	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Laboratorio a eliminar	cerrado	192.168.11.1/24	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 30. Seleccionando uno o más laboratorios existentes.

Se selecciona [Acción] y luego [Suprimir].



Numero de Laboratorio	Nombre del Laboratorio	Estado de la puerta	Acción	Habilitar Rango IP	Habilitar Equipos Informaticos
1	Software	cerrado	Exportar	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10	Laboratorio editado	cerrado	Exportar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Laboratorio a eliminar	cerrado	Suprimir	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 31. Accediendo a dar de baja uno o más laboratorios existentes.

En la ventana emergente, se selecciona [Aceptar] para eliminar permanentemente el laboratorio, o [Cancelar] para abortar la eliminación del laboratorio.

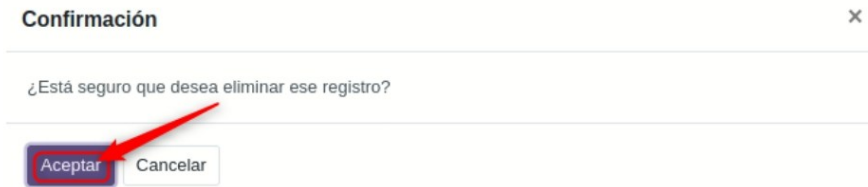


Figura 32. Eliminación de uno o más laboratorios existentes.

Una vez realizado estos pasos, ya se puede observar los laboratorios sin el/los laboratorios dados de baja del sistema.



Figura 33. Lista de laboratorios sin el/los laboratorios dados de baja del sistema.

4 Gestión de Equipos Informáticos

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].



Figura 34. Accediendo al módulo de Control de acceso.

Se selecciona [Gestionar Laboratorios].



Figura 35. Accediendo a la sección de gestionar laboratorios.

Se selecciona [Equipos Informáticos].

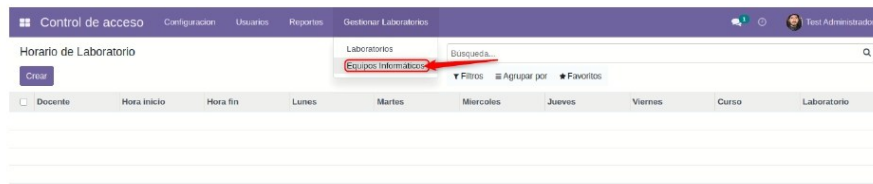


Figura 36. Accediendo a la subsección de equipos informáticos.

Una vez realizado estos pasos, ya se puede gestionar los equipos informáticos del sistema.

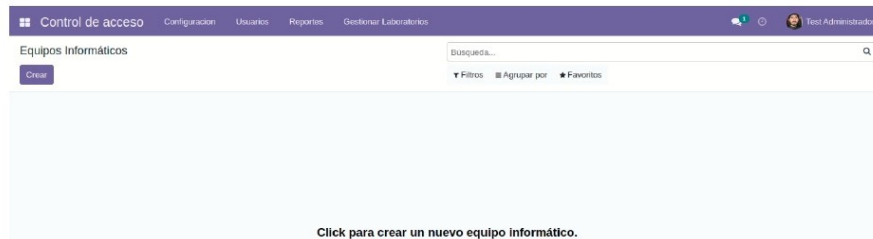


Figura 37. Vista de equipos informáticos.

4.1 Obtener UUID

NOTA: Como administrador, para poder registrar un equipo informático, antes deberá de generar un UUID en el dispositivo a registrar. Esto, lo tendrá que hacer ingresando en con su cuenta dentro de dicho dispositivo, y generando el UUID, para luego, copiarlo y pegarlo en el formulario de creación del equipo informático. Una vez aclarado el punto anterior, se procede a detallar el proceso para poder Obtener un UUID único siguiendo el formato de las secciones anteriores.

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].

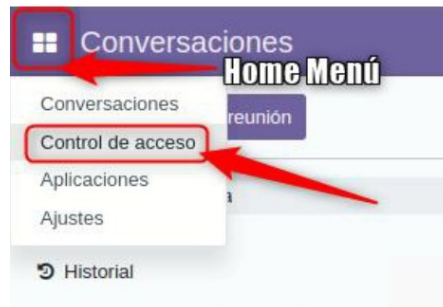


Figura 38. Accediendo al módulo de Control de acceso.

Se selecciona [Herramientas].

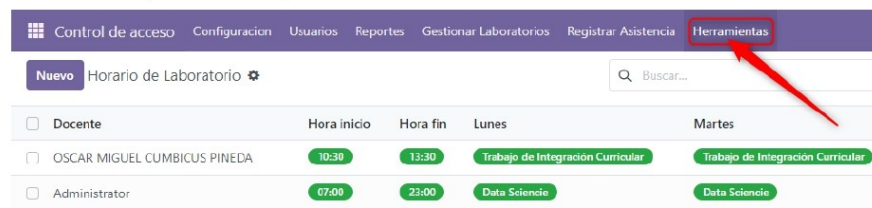


Figura 39. Accediendo a la sección de herramientas.

Se selecciona [Obtener UUID].



Figura 40. Accediendo a la subsección de Obtener UUID.

Se selecciona [Generar UUID].

Generar UUID

Información del Dispositivo

UUID:

Generar UUID

Copiar UUID

Figura 41. Vista para Generar un UUID.

Se selecciona [Copiar UUID].

Generar UUID

Información del Dispositivo

UUID:

7d553324-1315-4611-9e70-0347a8c0bde7

Generar UUID

Copiar UUID

Figura 42. Vista para Copiar el UUID generado.

Con estos pasos, ya se puede pegar el UUID generado en el formulario de creación del equipo informático.

4.2 Crear Equipo Informático

Se selecciona [Crear].

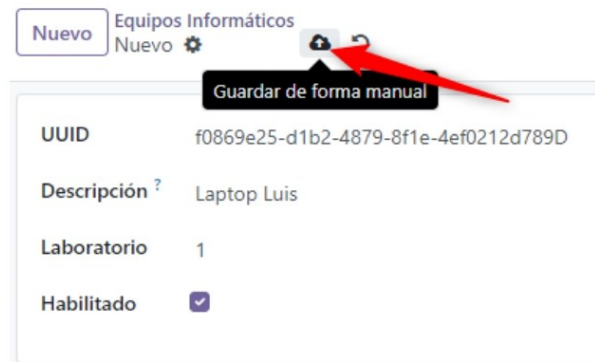
Control de acceso

Equipos Informáticos

Crear

Figura 43. Accediendo a crear un nuevo equipo informático.

Se procede a llenar los datos informativos del equipo informático (código identificador, UUID, laboratorio al que pertenece, habilitado). Tener en cuenta que todos los datos son obligatorios. El UUID debe ser único. En caso que se detecte datos ingresados no válidos, el sistema no permitirá el registro del equipo informático. Una vez completo todo el formulario, se selecciona [Guardar de forma manual]. Por otro lado, se puede cancelar el proceso de creación del equipo informático al seleccionar [Descartar cambios].



Nuevo Equipos Informáticos

Guardar de forma manual

UUID: f0869e25-d1b2-4879-8f1e-4ef0212d789D

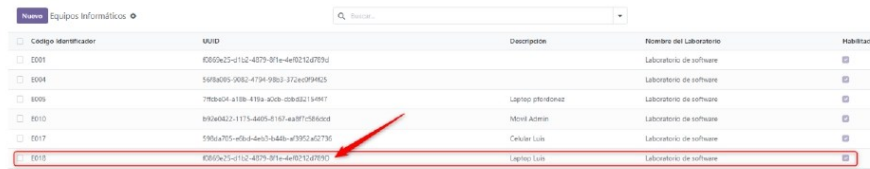
Descripción: Laptop Luis

Laboratorio: 1

Habilitado:

Figura 44. Creación de un nuevo equipo informático.

Una vez realizado estos pasos, ya se puede observar el nuevo equipo informático registrado en el sistema.



Codigo Identificador	UUID	Descripcion	Nombre del Laboratorio	Habilitado
E001	0305e25-d1b2-4879-8f1e-4ef0212d789D		Laboratorio de software	<input type="checkbox"/>
E004	54f8a095-2082-4794-9983-372e09994c55		Laboratorio de software	<input type="checkbox"/>
E005	78b8c04-418b-419a-a00b-08a852134947	Laptop pteronaz	Laboratorio de software	<input type="checkbox"/>
E010	910a0822-1175-4a05-8167-ea9f158660cd	Móvil Asteris	Laboratorio de software	<input type="checkbox"/>
E017	595a705-4b64-4e13-648b-w3952402736	Celular Luis	Laboratorio de software	<input type="checkbox"/>
E018	0305e25-d1b2-4879-8f1e-4ef0212d789D	Laptop Luis	Laboratorio de software	<input type="checkbox"/>

Figura 45. Lista de equipos informáticos con el nuevo equipo informático registrado en el sistema.

4.3 Editar Equipo Informático

Se selecciona un equipo informático.

Nuevo Equipos Informáticos

Código Identificador	UUID	Descripción	Nombre del Laboratorio	Habilitado
E001	0066e25-d1b2-4879-8f1e-4ef0212d789d		Laboratorio de software	<input type="checkbox"/>
E004	5498a05-9082-4794-9983-372e0994625		Laboratorio de software	<input type="checkbox"/>
E005	782a404-a109-419a-902b-c2852154917	Laptop pñedonez	Laboratorio de software	<input type="checkbox"/>
E010	8949622-1175-4485-8167-4a39f05860d	Móvil Adem	Laboratorio de software	<input type="checkbox"/>
E017	598a705-45b4-4a63-844b-4f392462736	Cellular Luis	Laboratorio de software	<input type="checkbox"/>
E018	0066e25-d1b2-4879-8f1e-4ef0212d789d	Laptop Luis	Laboratorio de software	<input type="checkbox"/>

Figura 46. Seleccionando un equipo informático existente.

Se procede a llenar los datos informativos del equipo informático (código identificador, UUID, laboratorio al que pertenece, habilitado). Tener en cuenta que todos los datos son obligatorios. El UUID debe ser único. En caso que se detecte datos ingresados no válidos, el sistema no permitirá el registro del equipo informático. Una vez completo todo el formulario, se selecciona [Guardar de forma manual]. Por otro lado, se puede cancelar el proceso de creación del equipo informático al seleccionar [Descartar cambios].

Nuevo Equipos Informáticos controlacceso.equipoinformatico,26

UUID: f0869e25-d1b2-4879-8f1e-4ef0212d789D
 Descripción:
 Laboratorio: 1
 Habilitado:

Figura 47. Edición de un equipo informático existente.

Una vez realizado estos pasos, ya se puede observar el equipo informático editado en el sistema.

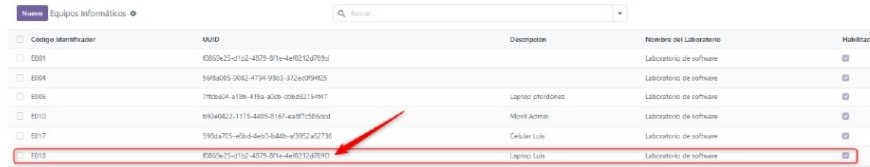
Nuevo Equipos Informáticos

Código Identificador	UUID	Descripción	Nombre del Laboratorio	Habilitado
E001	0066e25-d1b2-4879-8f1e-4ef0212d789d		Laboratorio de software	<input type="checkbox"/>
E004	5498a05-9082-4794-9983-372e0994625		Laboratorio de software	<input type="checkbox"/>
E005	782a404-a109-419a-902b-c2852154917	Laptop pñedonez	Laboratorio de software	<input type="checkbox"/>
E010	8949622-1175-4485-8167-4a39f05860d	Móvil Adem	Laboratorio de software	<input type="checkbox"/>
E017	598a705-45b4-4a63-844b-4f392462736	Cellular Luis	Laboratorio de software	<input type="checkbox"/>
E018	0066e25-d1b2-4879-8f1e-4ef0212d789d	Laptop Luis Editado	Laboratorio de software	<input type="checkbox"/>

Figura 48. Lista de equipos informáticos con el equipo informático editado en el sistema.

4.4 Dar de Baja Equipos Informáticos

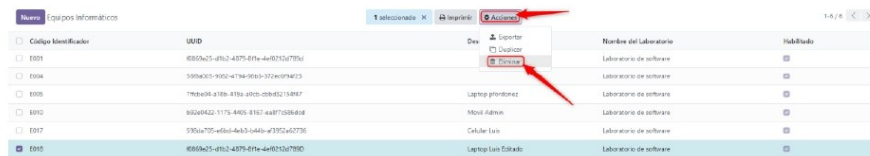
Se selecciona uno o más equipos informáticos.



Código Identificador	UUID	Descripción	Nombre del Laboratorio	Habilitado
<input type="checkbox"/>	E001	60662e25-d112-4879-9f1e-4e8212d4789d	Laboratorio de software	<input type="checkbox"/>
<input type="checkbox"/>	E004	54958095-9082-4794-9983-372e0994625	Laboratorio de software	<input type="checkbox"/>
<input type="checkbox"/>	E005	786b404-a18b-419a-800b-08a832134947	Laptop pterodact	<input type="checkbox"/>
<input type="checkbox"/>	E010	80248422-1175-4405-8167-aa9f1c5846cd	Móvil Acer	<input type="checkbox"/>
<input type="checkbox"/>	E017	595a705-a9d4-4a63-844b-w3952402736	Celular Luis	<input type="checkbox"/>
<input checked="" type="checkbox"/>	E018	60662e25-d112-4879-9f1e-4e8212d4789d	Laptop Luis	<input type="checkbox"/>

Figura 49. Seleccionando uno o más equipos informáticos existentes.

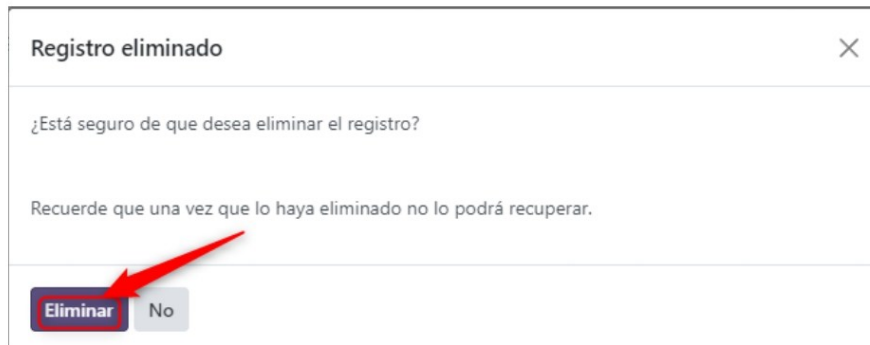
Se selecciona [Acción] y luego [Suprimir].



Código Identificador	UUID	Dispositivo	Nombre del Laboratorio	Habilitado
<input type="checkbox"/>	E001	60662e25-d112-4879-9f1e-4e8212d4789d	Laboratorio de software	<input type="checkbox"/>
<input type="checkbox"/>	E004	54958095-9082-4794-9983-372e0994625	Laboratorio de software	<input type="checkbox"/>
<input type="checkbox"/>	E005	786b404-a18b-419a-800b-08a832134947	Laptop pterodact	<input type="checkbox"/>
<input type="checkbox"/>	E010	80248422-1175-4405-8167-aa9f1c5846cd	Móvil Acer	<input type="checkbox"/>
<input type="checkbox"/>	E017	595a705-a9d4-4a63-844b-w3952402736	Celular Luis	<input type="checkbox"/>
<input checked="" type="checkbox"/>	E018	60662e25-d112-4879-9f1e-4e8212d4789d	Laptop Luis Editado	<input type="checkbox"/>

Figura 50. Accediendo a dar de baja uno o más equipos informáticos existentes.

En la ventana emergente, se selecciona [Aceptar] para eliminar permanentemente el equipo informático, o [Cancelar] para abortar la eliminación del equipo informático.



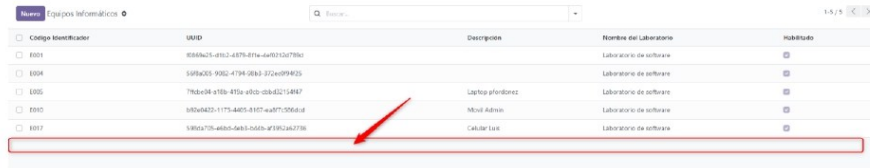
Registro eliminado ✕

¿Está seguro de que desea eliminar el registro?

Recuerde que una vez que lo haya eliminado no lo podrá recuperar.

Figura 51. Eliminación de uno o más equipos informáticos existentes.

Una vez realizado estos pasos, ya se puede observar los equipos informáticos sin el/los equipos informáticos dados de baja del sistema.

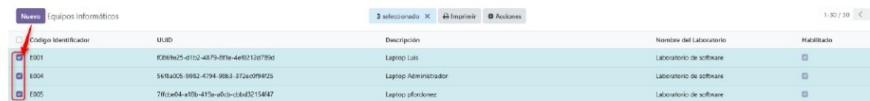


Código Identificador	UUID	Descripción	Nombre del Laboratorio	Habilitado
E001	80864c25-41b2-4879-8f1e-4a0212d789e	Laptop Lait	Laboratorio de software	<input type="checkbox"/>
E004	5496a005-9002-4794-981d-372e0094f25	Laptop Administrador	Laboratorio de software	<input type="checkbox"/>
E005	7f16e04-e18b-415a-a0eb-cba02154847	Laptop profesor	Laboratorio de software	<input type="checkbox"/>
E002	80d0a022-1175-4a05-8107-ea071506a1d	Movil Admin	Laboratorio de software	<input type="checkbox"/>
E003	51903705-4600-5683-565b-4f93c2a63796	Calculo Lait	Laboratorio de software	<input type="checkbox"/>

Figura 52. Lista de equipos informáticos sin el/los equipos informáticos dados de baja del sistema.

4.5 Generar Reporte de Equipos Informáticos

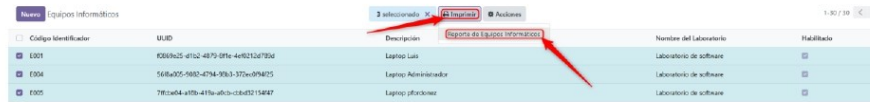
Se selecciona uno o más equipos informáticos.



Código Identificador	UUID	Descripción	Nombre del Laboratorio	Habilitado
<input checked="" type="checkbox"/>	E001	80864c25-41b2-4879-8f1e-4a0212d789e	Laptop Lait	Laboratorio de software
<input checked="" type="checkbox"/>	E004	5496a005-9002-4794-981d-372e0094f25	Laptop Administrador	Laboratorio de software
<input checked="" type="checkbox"/>	E005	7f16e04-e18b-415a-a0eb-cba02154847	Laptop profesor	Laboratorio de software

Figura 53. Seleccionando uno o más equipos informáticos existentes.

Se selecciona [Imprimir] y luego [Reporte de Equipos Informáticos].



Código Identificador	UUID	Descripción	Nombre del Laboratorio	Habilitado
<input checked="" type="checkbox"/>	E001	80864c25-41b2-4879-8f1e-4a0212d789e	Laptop Lait	Laboratorio de software
<input checked="" type="checkbox"/>	E004	5496a005-9002-4794-981d-372e0094f25	Laptop Administrador	Laboratorio de software
<input checked="" type="checkbox"/>	E005	7f16e04-e18b-415a-a0eb-cba02154847	Laptop profesor	Laboratorio de software

Figura 54. Accediendo a reporte de equipos informáticos.

Se previsualiza el reporte de equipos informáticos generado, y, si se encuentra todo en orden, se selecciona [Imprimir] para poder descargar el archivo en formato PDF.



Reporte de Equipos Informáticos

Loja, 15 de enero del 2005

Código Identificador	UUID	Habilitado
E001	80864c25-41b2-4879-8f1e-4a0212d789e	Si
E004	5496a005-9002-4794-981d-372e0094f25	Si
E005	7f16e04-e18b-415a-a0eb-cba02154847	Si

Figura 55. Descargando reporte de equipos informáticos en formato PDF.

Una vez realizado estos pasos, ya se puede observar el archivo PDF descargado con el reporte de los equipos informáticos, para su posterior impresión en físico.

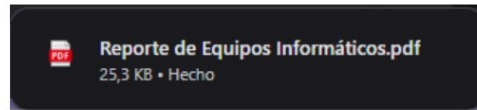


Figura 56. Descarga del reporte de equipos informáticos en formato PDF.

Your logo

Computación, Carrera de Computación
Ecuador

Reporte de Equipos Informáticos

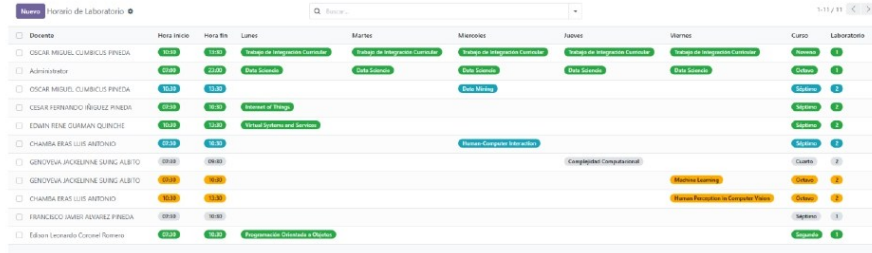
Loja, 16 de enero del 2025

Código Identificador	UUID	Habilitado
E001	f0869e25-d1b2-4879-8f1e-4ef0212d789d	Sí
E004	56f8a005-9082-4794-98b3-372ec0f94f25	Sí
E005	7ffcbe04-a18b-419a-a0cb-cbbd32154f47	Sí

Figura 57. Visualización del reporte de equipos informáticos en formato PDF.

5 Gestión de Registro Asistencia Face ID

NOTA: Como administrador, no se necesita registrar asistencia face ID dentro de un horario establecido previamente. Se puede marcar asistencia en cualquier momento del día, aunque sólo con los equipos informáticos configurados con antelación para dicho registro.



Docente	Hora inicio	Hora fin	Lunes	Martes	Miércoles	Jueves	Viernes	Curso	Laboratorio
<input type="checkbox"/> OSCAR MIGUEL CLIMBICUS PINEDA	08:00	08:30	Equipos de Manejo de Control	Equipos de Manejo de Control	Equipos de Manejo de Control	Equipos de Manejo de Control	Equipos de Manejo de Control	Control 1	1
<input type="checkbox"/> Administrador	08:00	08:30	Equipos de Manejo de Control	Equipos de Manejo de Control	Equipos de Manejo de Control	Equipos de Manejo de Control	Equipos de Manejo de Control	Control 1	1
<input type="checkbox"/> OSCAR MIGUEL CLIMBICUS PINEDA	08:30	09:00			Equipos de Manejo de Control			Control 1	1
<input type="checkbox"/> CESAR FERRANDO RIVERA PINEDA	08:30	09:00	Equipos de Manejo de Control					Control 1	1
<input type="checkbox"/> EDWIN FENE GUAMAN QUENDE	08:30	09:00	Equipos de Manejo de Control					Control 1	1
<input type="checkbox"/> CHAMISA BRAS LUIS ANTONIO	08:30	09:00						Control 1	1
<input type="checkbox"/> GENOVEVA JACQUELINE SAURIG ALBERTO	08:30	09:00						Control 1	1
<input type="checkbox"/> GENOVEVA JACQUELINE SAURIG ALBERTO	08:30	09:00						Control 1	1
<input type="checkbox"/> CHAMISA BRAS LUIS ANTONIO	08:30	09:00						Control 1	1
<input type="checkbox"/> FRANCISCO JAVIER ANDRÉS PINEDA	08:30	09:00						Control 1	1
<input type="checkbox"/> Edson Leonardo Coronel Romero	08:30	09:00	Equipos de Manejo de Control					Control 1	1

Figura 58. Vista de horario del módulo Control de Acceso.



Código Identificador	UUID	Descripción	Nombre del Laboratorio	Habilitado
<input type="checkbox"/> E001	0569a25-4f102-4676-9f9a-6a4521307894		Laboratorio de software	<input type="checkbox"/>
<input type="checkbox"/> E004	5498a025-9862-4794-9093-372a09594025		Laboratorio de software	<input type="checkbox"/>
<input type="checkbox"/> E005	785c464-a80b-4d3e-a65c-10a48714447	Laptop 64-bit	Laboratorio de software	<input type="checkbox"/>
<input type="checkbox"/> E005	9836a225-1175-4425-815f-a4a75155500d	Monitor Admin	Laboratorio de software	<input type="checkbox"/>
<input type="checkbox"/> E017	3992c795-6804-4a03-8449-4f3920a24739	Celular Luis	Laboratorio de software	<input type="checkbox"/>

Figura 59. Vista de equipos informáticos del módulo Control de Acceso.

Una vez aclarado el punto anterior, se procede a detallar el proceso para la Gestión de Registro de Asistencia Face ID siguiendo el formato de las secciones anteriores.

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].



Figura 60. Accediendo al módulo de Control de acceso.

Se selecciona [Reportes].

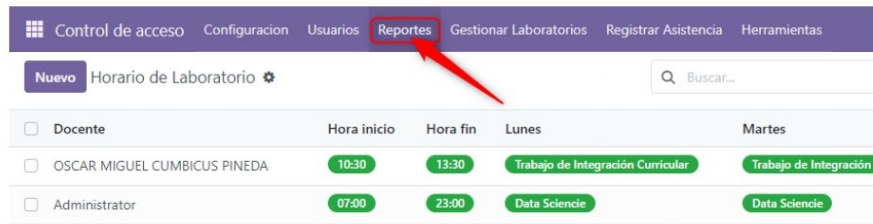


Figura 61. Accediendo a la sección de reportes.

Se selecciona [Asistencias Face ID].

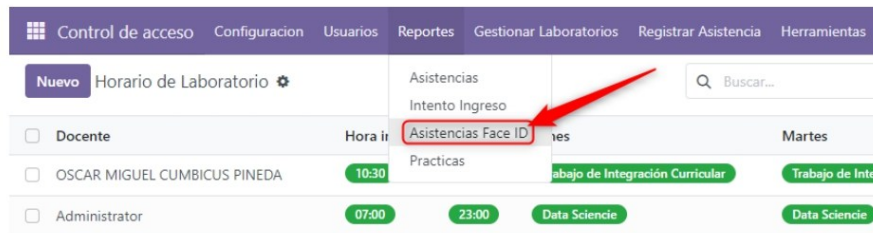


Figura 62. Accediendo a la subsección de asistencias face ID.

Una vez realizado estos pasos, ya se puede gestionar los registros de asistencia face ID del sistema.



Figura 63. Vista de Registro de Asistencia Face ID.

5.1 Crear Asistencia Face ID

Se selecciona [Registrar Asistencia], y luego [Face ID].

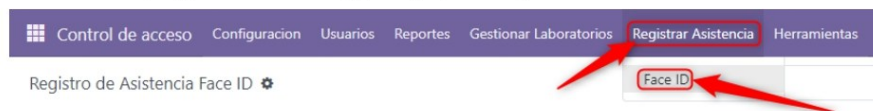


Figura 64. Accediendo a crear un nuevo registro de asistencia face ID.

Se procede a seleccionar [Registrar Asistencia].

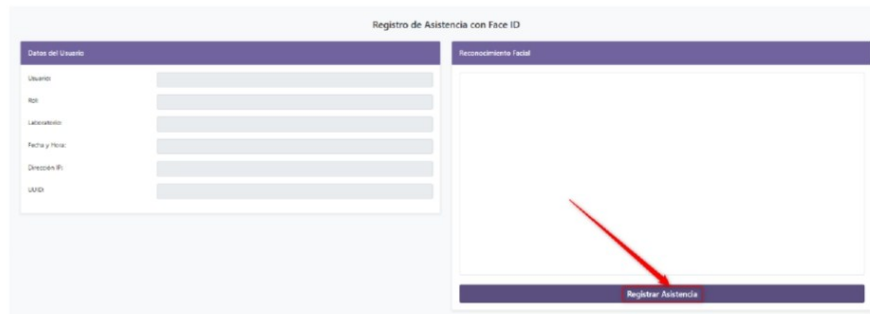


Figura 65. Creación de un nuevo registro de asistencia face ID.

El sistema empezará el reconocimiento facial y lo comparará con su imagen de usuario.

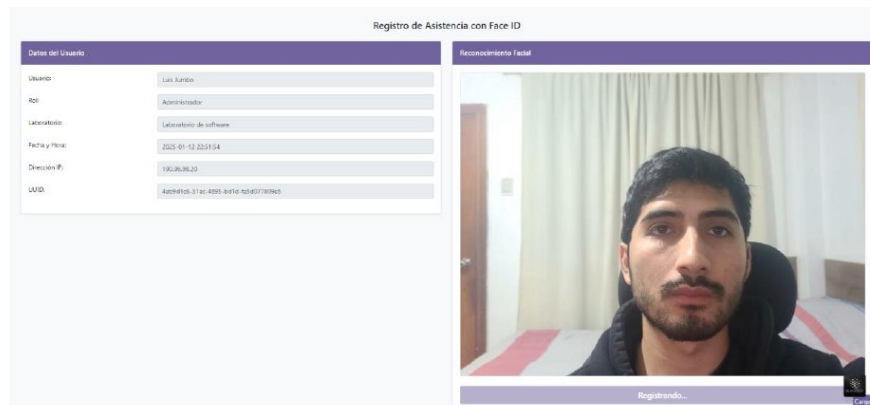


Figura 66. Reconociendo el rostro y comparándola con la imagen de usuario.

Si todo estuvo en orden, se crea un nuevo registro de asistencia face ID. En caso de que algún error ocurra, el sistema le informará mediante las notificaciones (no se documenta dichos errores por la extensión de los controles que se posee, por lo que resulta inviable dicha documentación)

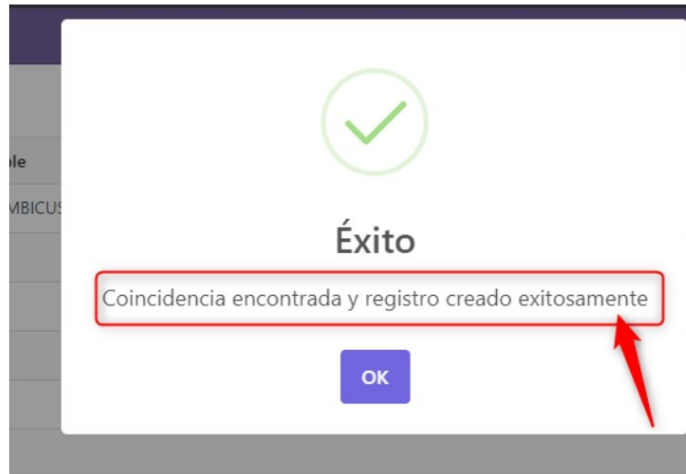


Figura 67. Creación de un nuevo registro de asistencia face ID exitoso.

Una vez realizado estos pasos, ya se puede observar el nuevo registro de asistencia face ID en el sistema.

Nombre del usuario	Rol Usuario	Laboratorio	Fecha y hora	Periodo	Centro	Curso	Materia	Evento	Responsable	Directorio IP	UUID
<input type="checkbox"/> Luis Jumbo	Administrador	Laboratorio de software	09/01/2025 15:12:27							192.168.58.251	0369a25-4152-4879-8f9a-448212a2789a
<input type="checkbox"/> Luis Jumbo	Administrador	Laboratorio de software	09/01/2025 15:15:00							10.20.206.143	568a205-9862-4794-9963-373ac6f4a25
<input type="checkbox"/> Luis Jumbo	Administrador	Laboratorio de software	09/01/2025 15:49:27							192.168.58.251	0369a25-4152-4879-8f9a-448212a2789a
<input type="checkbox"/> Administrador	Administrador	Laboratorio de software	10/01/2025 12:43:24							10.20.206.21	7f02a64-4158-415e-8508-c0a232152a67
<input type="checkbox"/> Administrador	Administrador	Laboratorio de software	10/01/2025 12:43:28							10.20.206.21	7f02a64-4158-415e-8508-c0a232152a67
<input type="checkbox"/> Luis Jumbo	Administrador	Laboratorio de software	10/01/2025 13:03:48							192.168.58.251	0369a25-4152-4879-8f9a-448212a2789a
<input type="checkbox"/> Luis Jumbo	Administrador	Laboratorio de software	10/01/2025 13:07:27							192.168.58.251	0369a25-4152-4879-8f9a-448212a2789a
<input type="checkbox"/> Luis Jumbo	Administrador	Laboratorio de software	10/01/2025 13:08:14							192.168.58.251	0369a25-4152-4879-8f9a-448212a2789a
<input type="checkbox"/> Administrador	Administrador	Laboratorio de software	10/01/2025 15:19:46							10.20.176.31	781ca64-4158-415e-8508-c0a232152a67
<input type="checkbox"/> Luis Jumbo	Administrador	Laboratorio de software	10/01/2025 22:11:52							192.168.58.20	4686118-51aa-4939-ca15-fa3077f07645
<input type="checkbox"/> Luis Jumbo	Administrador	Laboratorio de software	12/01/2025 22:51:54							192.168.58.20	4686118-51aa-4939-ca15-fa3077f07645

Figura 68. Lista de registros de asistencia face ID con el nuevo registro de asistencia face ID en el sistema.

5.2 Editar Asistencia Face ID

No se puede editar un registro de asistencia face ID bajo ningún motivo. El sistema presentará la información de dicho registro en campos deshabilitados.

Registro de Asistencia Face ID
controlacceso.registroasistenciafaceid,57 




Nombre del usuario	Luis Jumbo
Rol Usuario	Administrador
Laboratorio	Laboratorio de software
Fecha y Hora	2025-01-12T22:51:54.000-05:00
Periodo	
Carrera	
Curso	
Materia	
Docente Responsable	
Dirección IP	190.96.98.20
UUID	4eb9d1c6-31ac-4895-bd1d-fa5d077809e5

Figura 69. Seleccionando un registro de asistencia face ID existente.

5.3 Dar de Baja Asistencia Face ID

No se puede dar de baja un registro de asistencia face ID bajo ningún motivo. El sistema presentará la opción de [Acción], pero no presentará la opción de [Suprimir].

Registro de Asistencia Face ID

Seleccionado X   

<input type="checkbox"/>	Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
<input type="checkbox"/>	Luis Jumbo	Administrador	Laboratorio de software	09/01/2025 15:52:27						192.168.58.251	6259a25-4762-4879-8f9a-4a8212a17894
<input type="checkbox"/>	Luis Jumbo	Administrador	Laboratorio de software	09/01/2025 15:55:08						192.168.58.251	565ba205-f680-4764-9903-3776cd6425
<input type="checkbox"/>	Luis Jumbo	Administrador	Laboratorio de software	09/01/2025 16:09:27						192.168.58.251	6259a25-4762-4879-8f9a-4a8212a17894
<input type="checkbox"/>	Administrador	Administrador	Laboratorio de software	10/01/2025 12:43:24						192.168.58.251	7f02a04-4158-4158-4608-09a032154467
<input type="checkbox"/>	Administrador	Administrador	Laboratorio de software	10/01/2025 12:43:58						192.168.58.251	7f02a04-4158-4158-4608-09a032154467
<input type="checkbox"/>	Luis Jumbo	Administrador	Laboratorio de software	10/01/2025 13:03:08						192.168.58.251	6259a25-4762-4879-8f9a-4a8212a17894
<input type="checkbox"/>	Luis Jumbo	Administrador	Laboratorio de software	10/01/2025 13:07:37						192.168.58.251	6259a25-4762-4879-8f9a-4a8212a17894
<input type="checkbox"/>	Luis Jumbo	Administrador	Laboratorio de software	10/01/2025 13:08:54						192.168.58.251	6259a25-4762-4879-8f9a-4a8212a17894
<input checked="" type="checkbox"/>	Administrador	Administrador	Laboratorio de software	10/01/2025 15:59:46						192.168.58.251	7f02a04-4158-4158-4608-09a032154467
<input checked="" type="checkbox"/>	Luis Jumbo	Administrador	Laboratorio de software	12/01/2025 22:51:52						190.96.98.20	4eb9d1c6-31ac-4895-bd1d-fa5d077809e5
<input checked="" type="checkbox"/>	Luis Jumbo	Administrador	Laboratorio de software	12/01/2025 22:51:54						190.96.98.20	4eb9d1c6-31ac-4895-bd1d-fa5d077809e5

Figura 70. Seleccionando un registro de asistencia face ID existente.

5.4 Generar Reporte de Asistencia Face ID

Se selecciona uno o más registros de asistencia face ID.

Registro de Asistencia Face ID

3 seleccionado X [Imprimir] [Reporte de Asistencia Face ID] [Acciones]

1:25 / 28

<input type="checkbox"/>	Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
<input checked="" type="checkbox"/>	Administrador	Administrador	Laboratorio de software	15/01/2025 15:19:46						10.10.193.31	79c2e84-y19b-419e-a03c-c8d3215487
<input checked="" type="checkbox"/>	Test Docente	Docente	Laboratorio de software	15/01/2025 01:14:19	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular		190.96.96.20	a850e916-fc3a-4078-aed8-d5d5a72625
<input checked="" type="checkbox"/>	Test Estudiante	Estudiante	Laboratorio de software	15/01/2025 00:36:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.96.20	e88ed16-fc3a-4078-aed8-d5d5a72625

Figura 71. Seleccionando uno o más registros de asistencia face ID existentes.

Se selecciona [Imprimir] y luego [Reporte de Registros de Asistencia Face ID].

Registro de Asistencia Face ID

3 seleccionado X [Imprimir] [Reporte de Asistencia Face ID] [Acciones]

1:25 / 28

<input type="checkbox"/>	Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
<input checked="" type="checkbox"/>	Administrador	Administrador	Laboratorio de software	15/01/2025 15:19:46						10.10.193.31	79c2e84-y19b-419e-a03c-c8d3215487
<input checked="" type="checkbox"/>	Test Docente	Docente	Laboratorio de software	15/01/2025 01:14:19	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular		190.96.96.20	a850e916-fc3a-4078-aed8-d5d5a72625
<input checked="" type="checkbox"/>	Test Estudiante	Estudiante	Laboratorio de software	15/01/2025 00:36:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.96.20	e88ed16-fc3a-4078-aed8-d5d5a72625

Figura 72. Accediendo a reporte de registros de asistencia face ID.

Se previsualiza el reporte de registros de asistencia face ID generado, y, si se encuentra todo en orden, se selecciona [Imprimir] para poder descargar el archivo en formato PDF.

 Registro de Asistencia Face ID
 Registro de Asistencia Face ID
 Your logo
 Computación, Carrera de Computación Ecuador

Reporte de Asistencia Face ID

Loja, 15 de enero del 2025

Usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Administrador	Administrador	Laboratorio de software	2025-01-10 20:19:46						10.20.136.21	7f8b204-e18b-419e-a0db-08bd32154f47
Test Docente	Docente	Laboratorio de software	2025-01-13 06:14:19	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular		190.96.98.20	e888e916-fb3a-4079-ae88-0e0e4470e26
Test Estudiante	Estudiante	Laboratorio de software	2025-01-15 05:36:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.98.20	e888e916-fb3a-4079-ae88-0e0e4470e26

Figura 73. Descargando reporte de registros de asistencia face ID en formato PDF.

Una vez realizado estos pasos, ya se puede observar el archivo PDF descargado con el reporte de los registros de asistencia face ID, para su posterior impresión en físico.

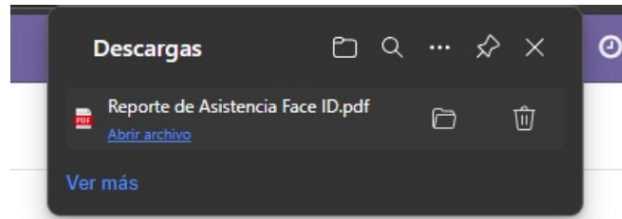


Figura 74. Descarga del reporte de registros de asistencia face ID en formato PDF.

 Your logo

Computación, Carrera de Computación
Ecuador

Reporte de Asistencia Face ID

Loja, 15 de enero del 2025

Usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Administrator	Administrador	Laboratorio de software	2025-01-10 20:19:46						10.20.136.31	7ffcbe04-a18b-419a-a0cb-cbbd32154f47
Test Docente	Docente	Laboratorio de software	2025-01-13 06:14:19	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular		190.96.98.20	e888e916-fb3a-4078-ae88-dd0e5a470e26
Test Estudiante	Estudiante	Laboratorio de software	2025-01-15 05:36:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.98.20	e888e916-fb3a-4078-ae88-dd0e5a470e26

Figura 75. Visualización del reporte de registros de asistencia face ID en formato PDF.

Para constancia y validación de la información expuesta, firma a continuación el tutor del Trabajo de Integración Curricular, en su calidad de docente guía en el desarrollo del proyecto:

Ing. Edison Leonardo Coronel Romero, Mg.Sc.	 Firma digitalizada por EDISON LEONARDO CORONEL ROMERO
---	--

ANEXO 7: Manual de usuario para docente



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

- **Revisado y aprobado por:**

- Ing. Edison Leonardo Coronel Romero, Mg.Sc., tutor del Trabajo de Integración Curricular.

Descripción: El presente anexo contiene el manual de usuario dirigido al docente del sistema. En él se describen las funcionalidades específicas que los docentes pueden realizar, organizadas en secciones. Los docentes pueden registrar su asistencia mediante reconocimiento facial, siempre que lo hagan dentro del horario establecido. Además, tienen acceso a la sección "Asistencias Face ID", donde pueden visualizar sus propias asistencias y las de los estudiantes a su cargo. También se incluye la funcionalidad para generar nuevos registros de asistencia. Este manual proporciona una guía detallada para que los docentes puedan gestionar de manera eficiente la asistencia de sus estudiantes y la suya propia.

MANUAL DE USUARIO PARA DOCENTE

Es necesario mencionar que este manual de usuario se centra en la nueva funcionalidad del módulo de Control de Acceso (Registro de Asistencia Face ID) dirigido al docente, y más no en las funcionalidades existentes del mismo (gestión de prácticas, gestión de perfil de usuario, etc.).

1 Ingreso a Sistema

Se debe ingresar a la siguiente dirección web: [https://computacion.unl.edu.ec/web].

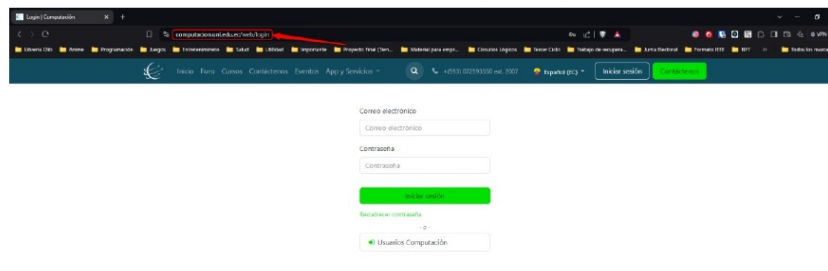


Figura 1. Accediendo al sistema mediante el navegador web

Se debe iniciar sesión con las respectivas credenciales del docente.

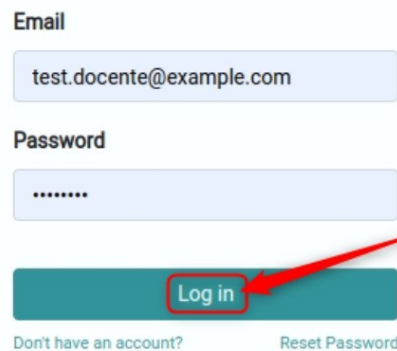


Figura 2. Inicio de sesión

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].



Figura 3. Accediendo al módulo de Control de acceso.

Una vez realizado estos pasos, ya se puede hacer uso del sistema de [Control de Acceso] como docente.

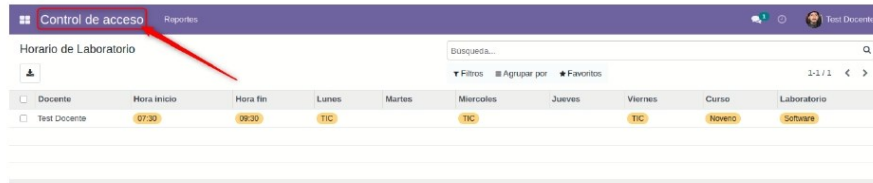


Figura 4. Vista del módulo de Control de acceso

2 Obtener UUID

NOTA: En caso de que el administrador solicite el UUID de su dispositivo, como docente deberá de generar un UUID en su equipo, y este pasarle por cualquier medio al administrador del laboratorio. Esto, lo tendrá que hacer ingresando en con su cuenta dentro del equipo y generando el UUID, para luego, copiarlo y enviarlo por el medio indicado por el administrador. Una vez aclarado el punto anterior, se procede a detallar el proceso para poder Obtener un UUID único siguiendo el formato de las secciones anteriores.

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].

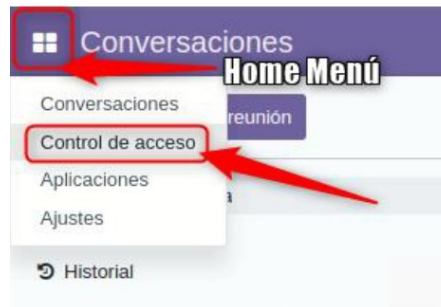


Figura 5. Accediendo al módulo de Control de acceso.

Se selecciona [Herramientas].



Figura 6. Accediendo a la sección de herramientas.

Se selecciona [Obtener UUID].



Figura 7. Accediendo a la subsección de Obtener UUID.

Se selecciona [Generar UUID].

Generar UUID

Información del Dispositivo

UUID:

Generar UUID

Copiar UUID

Figura 8. Vista para Generar un UUID.

Se selecciona [Copiar UUID].

Generar UUID

Información del Dispositivo

UUID:

7d553324-1315-4611-9e70-0347a8cbbde7

Generar UUID

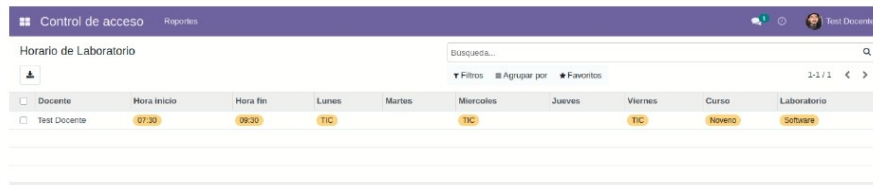
Copiar UUID

Figura 9. Vista para Copiar el UUID generado.

Con estos pasos, ya se puede pegar el UUID por el medio indicado por el administrador, con ello, se podrá registrar el equipo informático dentro del sistema.

3 Gestión de Registro Asistencia Face ID

NOTA: Como docente, únicamente se puede registrar asistencia face ID dentro de un horario establecido previamente, y sólo con los equipos informáticos configurados con antelación para dicho registro.



Docente	Hora inicio	Hora fin	Lunes	Martes	Miércoles	Jueves	Viernes	Curso	Laboratorio
<input type="checkbox"/> Test Docente	07:30	08:30	TIC		TIC		TIC	Novena	Software

Figura 10. Vista de horario del módulo Control de Acceso.

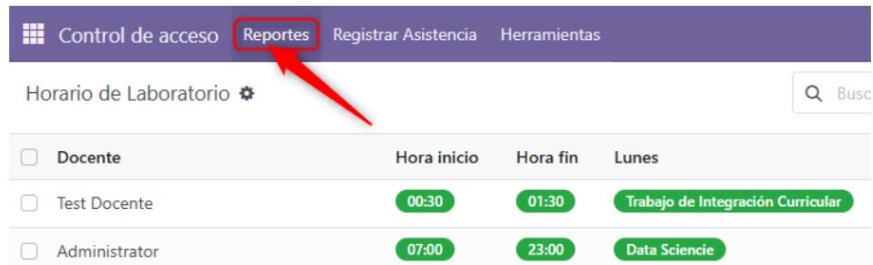
Una vez aclarado el punto anterior, se procede a detallar el proceso para la Gestión de Registro de Asistencia Face ID siguiendo el formato de las secciones anteriores.

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].



Figura 11. Accediendo al módulo de Control de acceso.

Se selecciona [Reportes].



Docente	Hora inicio	Hora fin	Lunes
<input type="checkbox"/> Test Docente	00:30	01:30	Trabajo de Integración Curricular
<input type="checkbox"/> Administrator	07:00	23:00	Data Science

Figura 12. Accediendo a la sección de reportes.

Se selecciona [Asistencias Face ID].

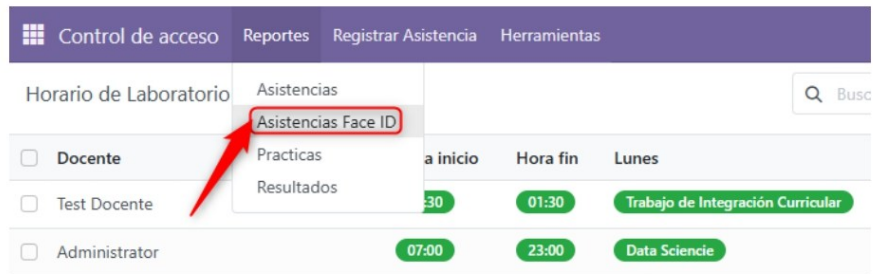


Figura 13. Accediendo a la subsección de asistencias face ID.

Una vez realizado estos pasos, ya se puede gestionar los registros de asistencia face ID del sistema como docente, lo cual compete la visualización tanto de las asistencias propias del usuario, como de todos los estudiantes a los cuales imparte clases.

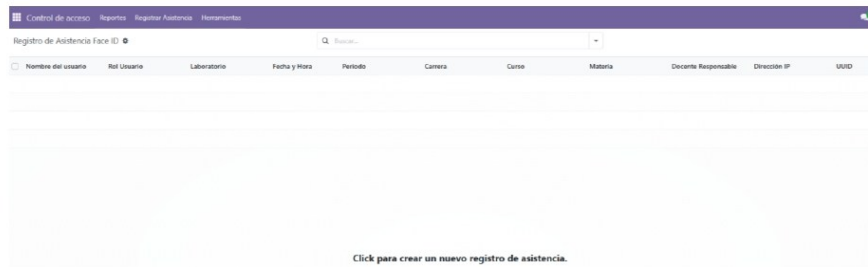


Figura 14. Vista de Registro de Asistencia Face ID.

3.1 Crear Asistencia Face ID

Se selecciona [Registrar Asistencia], y luego [Face ID].

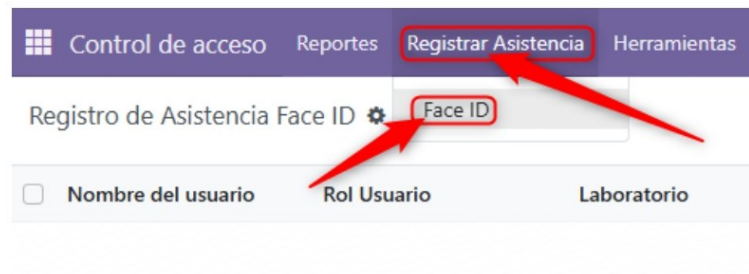


Figura 15. Accediendo a crear un nuevo registro de asistencia face ID.

Se procede a seleccionar [Registrar Asistencia].

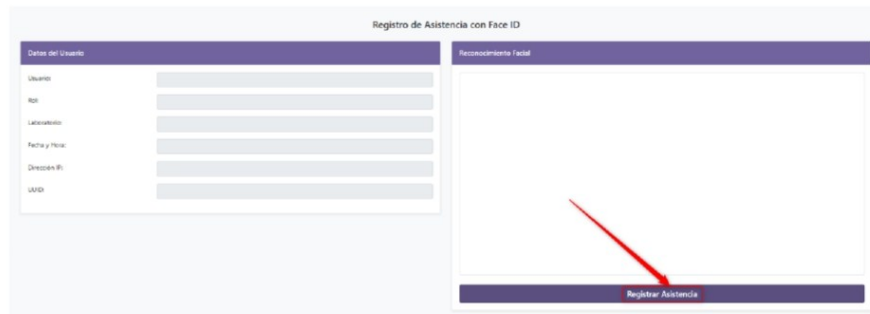


Figura 16. Creación de un nuevo registro de asistencia face ID.

El sistema empezará el reconocimiento facial y lo comparará con su imagen de usuario.

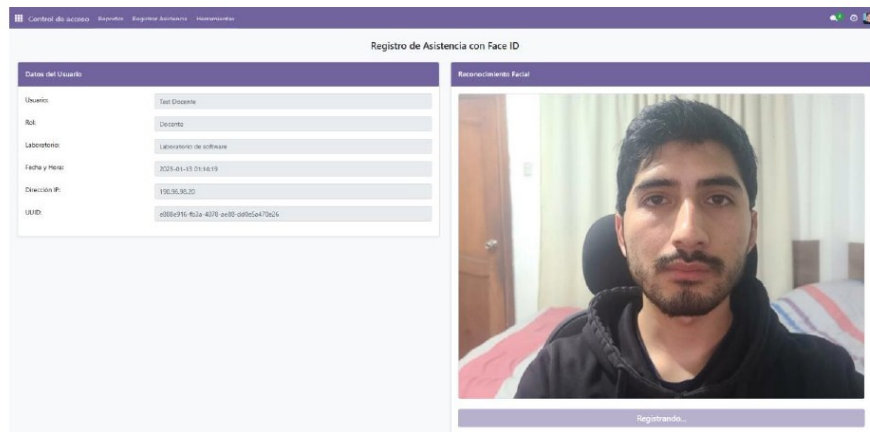


Figura 17. Reconociendo el rostro y comparándola con la imagen de usuario.

Si todo estuvo en orden, se crea un nuevo registro de asistencia face ID. En caso de que algún error ocurra, el sistema le informará mediante las notificaciones (no se documenta dichos errores por la extensión de los controles que se posee, por lo que resulta inviable dicha documentación). No obstante, es necesario hacer hincapié en que, como docente, sólo puede realizar un único registro de asistencia face ID por clase (materia, curso y hora específicas) dentro del horario establecido. Es decir, no se puede volver a realizar un registro de asistencia Face ID si ya se marcó asistencia previamente para esa materia y ese curso con una hora concretamente. Por tanto, para la siguiente clase (aunque sea la misma materia, al ser otro

curso y otra hora específica, se considera otra clase) se deberá realizar el registro de asistencia Face ID con normalidad.

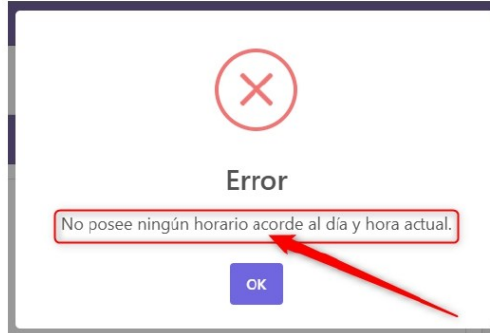


Figura 18. Creación de un nuevo registro de asistencia face ID fallido por intentar marcar asistencia fuera de horario.

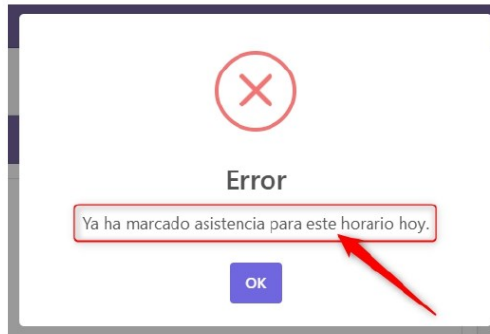


Figura 19. Creación de un nuevo registro de asistencia face ID fallido por intentar marcar asistencia cuando ya se ha marcado asistencia anteriormente.

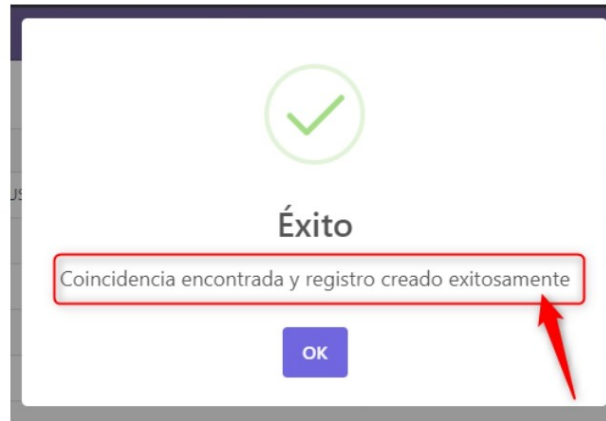
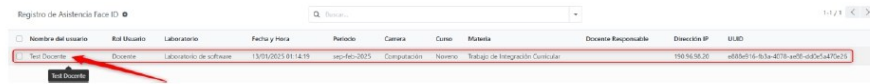


Figura 20. Creación de un nuevo registro de asistencia face ID exitoso.

Una vez realizado estos pasos, ya se puede observar el nuevo registro de asistencia face ID en el sistema. Tomar en consideración que de igual manera debe de verse reflejado los nuevos registros de asistencia Face ID de los estudiantes que asistieron a dicha clase.



Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Período	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UIID
Test Docente	Docente	Laboratorio de software	13/01/2025 01:14:19	sep-feb-2025	Computación	Noveno	Trabajo de Integración Computar		192.168.20	e853d16-91a-407b-ae08-05041447913

Figura 21. Lista de registros de asistencia face ID con el nuevo registro de asistencia face ID en el sistema.

3.2 Editar Asistencia Face ID

No se puede editar un registro de asistencia face ID bajo ningún motivo. El sistema presentará la información de dicho registro en campos deshabilitados.

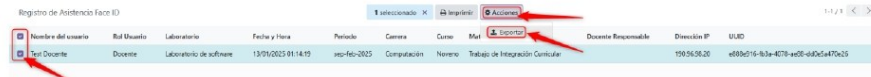
Registro de Asistencia Face ID
controlacceso.registroasistenciafaceid.58

Nombre del usuario	Test Docente
Rol Usuario	Docente
Laboratorio	Laboratorio de software
Fecha y Hora	2025-01-13T01:14:19.000-05:00
Periodo	sep-feb-2025
Carrera	Computación
Curso	Noveno
Materia	Trabajo de Integración Curricular
Docente Responsable	
Dirección IP	190.96.98.20
UUID	e888e916-fb3a-4078-ae88-dd0e5a470e26

Figura 22. Seleccionando un registro de asistencia face ID existente.

3.3 Dar de Baja Asistencia Face ID

No se puede dar de baja un registro de asistencia face ID bajo ningún motivo. El sistema presentará la opción de [Acción], pero no presentará la opción de [Suprimir].

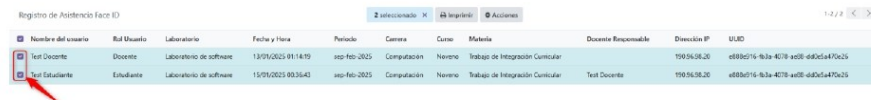


Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Mat	Docente Responsable	Dirección IP	UUID
Test Docente	Docente	Laboratorio de software	13/01/2025 01:14:19	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular		190.96.98.20	e888e916-fb3a-4078-ae88-dd0e5a470e26

Figura 23. Seleccionando un registro de asistencia face ID existente.

3.4 Generar Reporte de Asistencia Face ID

Se selecciona uno o más registros de asistencia face ID.



Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Test Docente	Docente	Laboratorio de software	13/01/2025 01:14:19	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.98.20	e888e916-fb3a-4078-ae88-dd0e5a470e26
Test Estudiante	Estudiante	Laboratorio de software	15/01/2025 09:30:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.98.20	e888e916-fb3a-4078-ae88-dd0e5a470e26

Figura 24. Seleccionando uno o más registros de asistencia face ID existentes.

Se selecciona [Imprimir] y luego [Reporte de Registros de Asistencia Face ID].

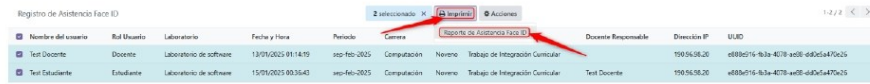


Figura 25. Accediendo a reporte de registros de asistencia face ID.

Se previsualiza el reporte de registros de asistencia face ID generado, y, si se encuentra todo en orden, se selecciona [Imprimir] para poder descargar el archivo en formato PDF.



Reporte de Asistencia Face ID

Loja, 15 de enero del 2025

Usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Test Docente	Docente	Laboratorio de software	2025-01-12 04:34:12	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular		195.94.98.20	e888e716-83a-4078-ae68-d6d5e470a26
Test Estudiante	Estudiante	Laboratorio de software	2025-01-15 05:36:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	195.94.98.20	e888e716-83a-4078-ae68-d6d5e470a26

Figura 26. Descargando reporte de registros de asistencia face ID en formato PDF.

Una vez realizado estos pasos, ya se puede observar el archivo PDF descargado con el reporte de los registros de asistencia face ID, para su posterior impresión en físico.

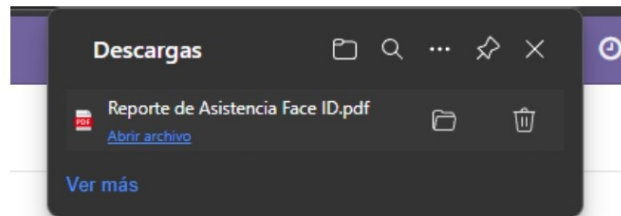


Figura 27. Descarga del reporte de registros de asistencia face ID en formato PDF.

 Your logo

Computación, Carrera de Computación
Ecuador

Reporte de Asistencia Face ID

Loja, 15 de enero del 2025

Usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Test Docente	Docente	Laboratorio de software	2025-01-13 06:14:19	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular		190.96.98.20	e888e916-fb3a-4078-a880-d40e5a470e26
Test Estudiante	Estudiante	Laboratorio de software	2025-01-15 09:30:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.98.20	e888e916-fb3a-4078-a880-d40e5a470e26

Figura 28. Visualización del reporte de registros de asistencia face ID en formato PDF.

Para constancia y validación de la información expuesta, firma a continuación el tutor del Trabajo de Integración Curricular, en su calidad de docente guía en el desarrollo del proyecto:

Ing. Edison Leonardo Coronel Romero, Mg.Sc.



ANEXO 8: Manual de usuario para estudiante



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

- **Revisado y aprobado por:**

- Ing. Edison Leonardo Coronel Romero, Mg.Sc., tutor del Trabajo de Integración Curricular.

Descripción: El presente anexo contiene el manual de usuario para el estudiante del sistema. Se detallan las funcionalidades a las que los estudiantes tienen acceso, organizadas por secciones. Los estudiantes pueden registrar su asistencia a los laboratorios utilizando reconocimiento facial, siempre y cuando lo hagan dentro del horario establecido. Tienen acceso a la sección "Asistencias Face ID", donde pueden consultar únicamente sus propios registros de asistencia y generar nuevos registros cuando sea necesario. Este manual ofrece una guía clara y concisa para que los estudiantes puedan registrar y revisar su asistencia de manera efectiva.

MANUAL DE USUARIO PARA ESTUDIANTE

Es necesario mencionar que este manual de usuario se centra en la nueva funcionalidad del módulo de Control de Acceso (Registro de Asistencia Face ID) dirigido al estudiante, y más no en las funcionalidades existentes del mismo (gestión de prácticas, gestión de perfil de usuario, etc.).

1 Ingreso a Sistema

Se debe ingresar a la siguiente dirección web: [https://computacion.unl.edu.ec/web].

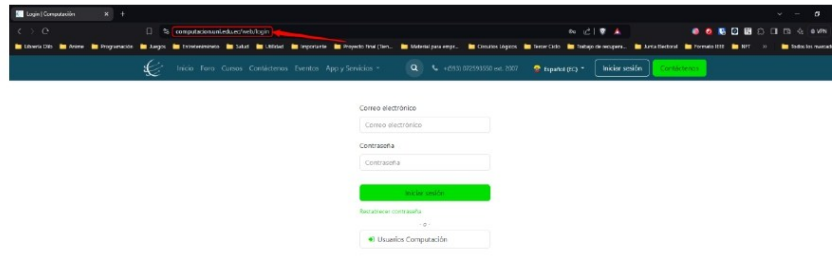


Figura 1. Accediendo al sistema mediante el navegador web

Se debe iniciar sesión con las respectivas credenciales del estudiante.

Email

Password

Log in

[Don't have an account?](#) [Reset Password](#)

Figura 2. Inicio de sesión

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].



Figura 3. Accediendo al módulo de Control de acceso.

Una vez realizado estos pasos, ya se puede hacer uso del sistema de [Control de Acceso] como estudiante.

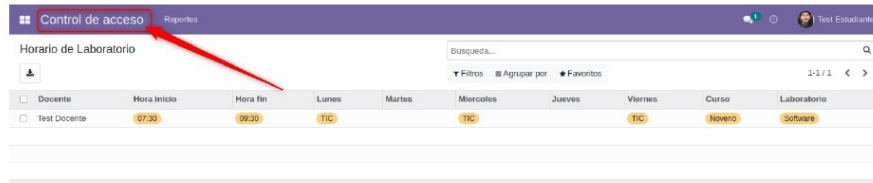


Figura 4. Vista del módulo de Control de acceso

2 Obtener UUID

NOTA: En caso de que el administrador solicite el UUID de su dispositivo, como estudiante deberá de generar un UUID en su equipo, y este pasarle por cualquier medio al administrador del laboratorio. Esto, lo tendrá que hacer ingresando en con su cuenta dentro del equipo y generando el UUID, para luego, copiarlo y enviarlo por el medio indicado por el administrador. Una vez aclarado el punto anterior, se procede a detallar el proceso para poder Obtener un UUID único siguiendo el formato de las secciones anteriores.

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].

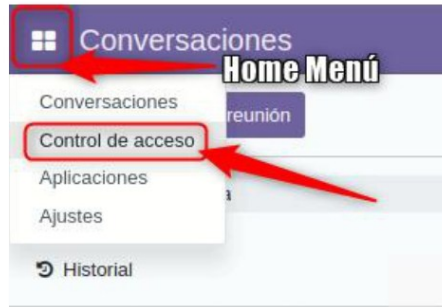


Figura 5. Accediendo al módulo de Control de acceso.

Se selecciona [Herramientas].



Figura 6. Accediendo a la sección de herramientas.

Se selecciona [Obtener UUID].



Figura 7. Accediendo a la subsección de Obtener UUID.

Se selecciona [Generar UUID].

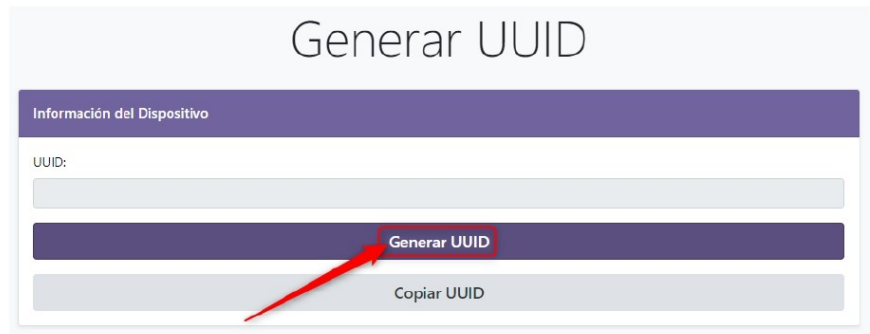


Figura 8. Vista para Generar un UUID.

Se selecciona [Copiar UUID].


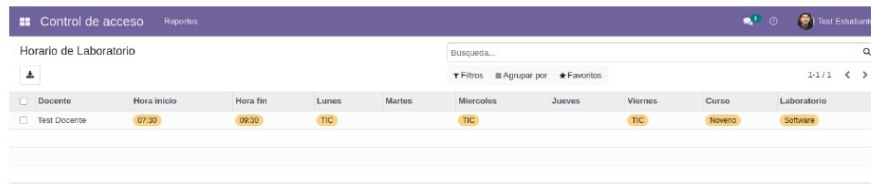


Figura 9. Vista para Copiar el UUID generado.

Con estos pasos, ya se puede pegar el UUID por el medio indicado por el administrador, con ello, se podrá registrar el equipo informático dentro del sistema.

3 Gestión de Registro Asistencia Face ID

NOTA: Como estudiante, únicamente se puede registrar asistencia face ID dentro de un horario establecido previamente, y sólo con los equipos informáticos configurados con antelación para dicho registro.



Docente	Hora inicio	Hora fin	Lunes	Martes	Miércoles	Jueves	Viernes	Curso	Laboratorio
<input type="checkbox"/> Test Docente	07:30	08:30	TIC		TIC		TIC	Novena	Software

Figura 10. Vista de horario del módulo Control de Acceso.

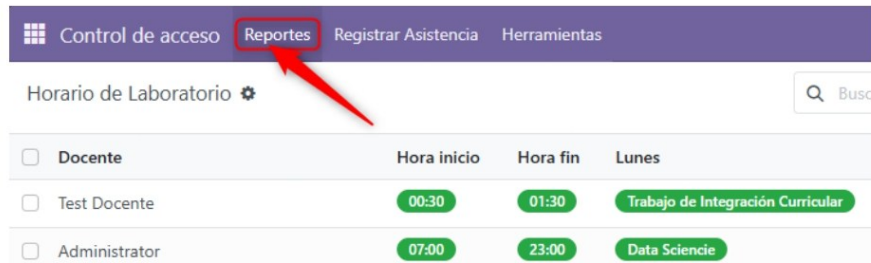
Una vez aclarado el punto anterior, se procede a detallar el proceso para la Gestión de Registro de Asistencia Face ID siguiendo el formato de las secciones anteriores.

Se selecciona [Home Menu] y se procede a seleccionar el módulo de [Control de Acceso].



Figura 11. Accediendo al módulo de Control de acceso.

Se selecciona [Reportes].



Docente	Hora inicio	Hora fin	Lunes
<input type="checkbox"/> Test Docente	00:30	01:30	Trabajo de Integración Curricular
<input type="checkbox"/> Administrator	07:00	23:00	Data Science

Figura 12. Accediendo a la sección de reportes.

Se selecciona [Asistencias Face ID].

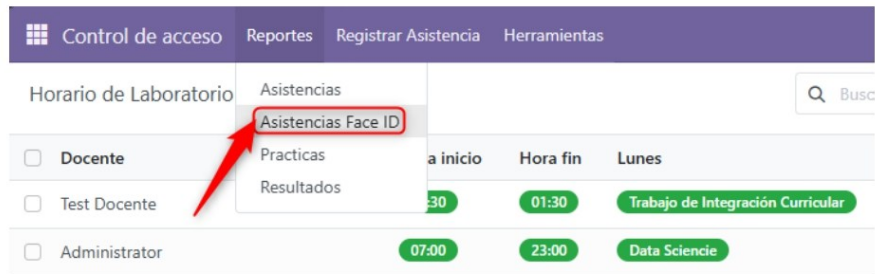


Figura 13. Accediendo a la subsección de asistencias face ID.

Una vez realizado estos pasos, ya se puede gestionar los registros de asistencia face ID del sistema como estudiante, lo cual compete la visualización de las asistencias propias del usuario.

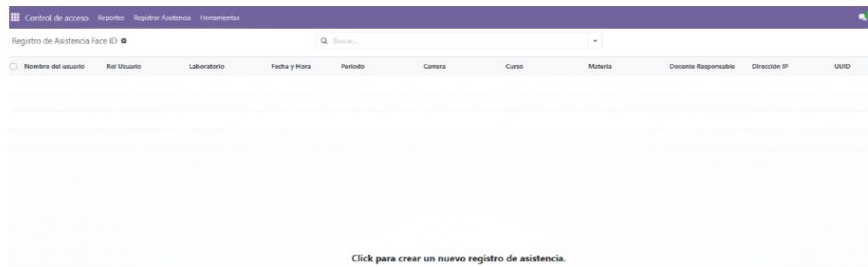


Figura 14. Vista de Registro de Asistencia Face ID.

3.1 Crear Asistencia Face ID

Se selecciona [Crear].

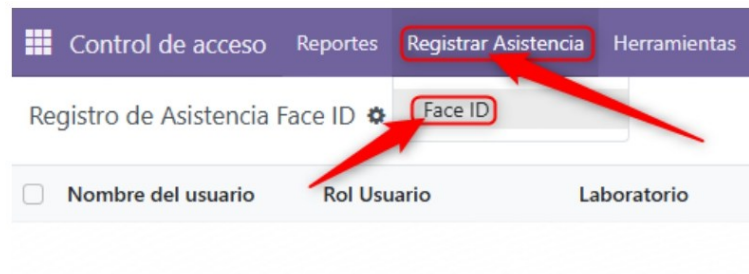


Figura 15. Accediendo a crear un nuevo registro de asistencia face ID.

Se procede a seleccionar [Registrar Asistencia].

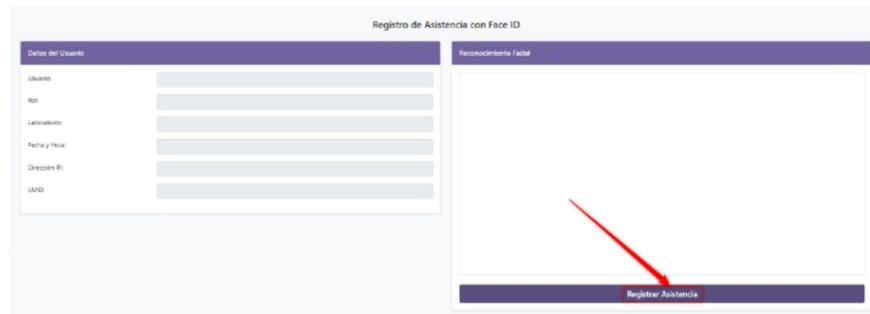


Figura 16. Creación de un nuevo registro de asistencia face ID.

El sistema empezará el reconocimiento facial y lo comparará con su imagen de usuario.

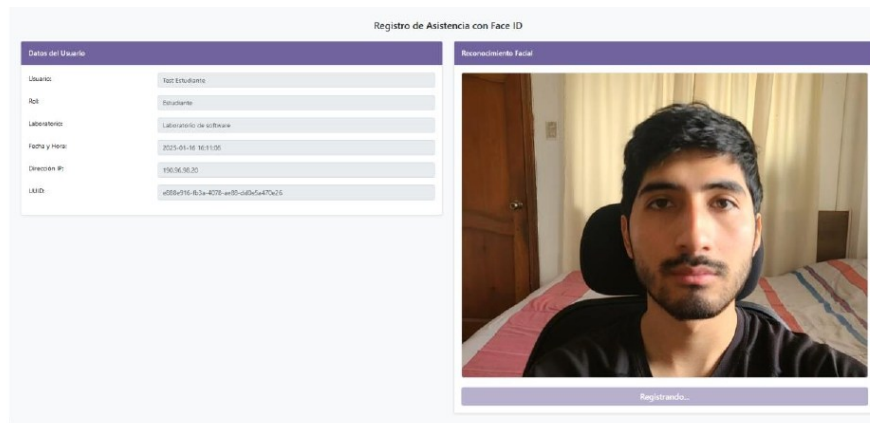


Figura 17. Reconociendo el rostro y comparándola con la imagen de usuario.

Si todo estuvo en orden, se crea un nuevo registro de asistencia face ID. En caso de que algún error ocurra, el sistema le informará mediante las notificaciones (no se documenta dichos errores por la extensión de los controles que se posee, por lo que resulta inviable dicha documentación). No obstante, es necesario hacer hincapié en que, como estudiante, sólo puede realizar un único registro de asistencia face ID por clase (materia, curso y hora específicas) dentro del horario establecido. Es decir, no se puede volver a realizar un registro de asistencia Face ID si ya se marcó asistencia previamente para esa materia y ese curso con una hora concretamente. Por tanto, para la siguiente clase (aunque sea el mismo curso, al ser

otra materia y otra hora específica, se considera otra clase) se deberá realizar el registro de asistencia Face ID con normalidad.

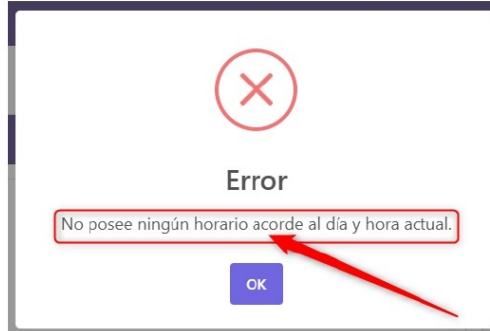


Figura 18. Creación de un nuevo registro de asistencia face ID fallido por intentar marcar asistencia fuera de horario.

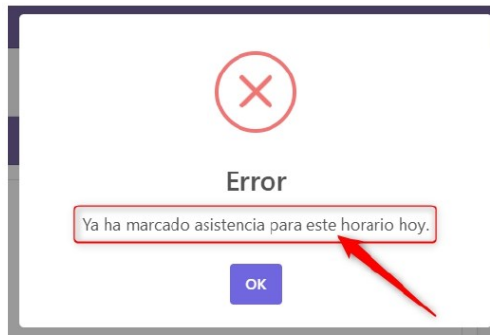


Figura 19. Creación de un nuevo registro de asistencia face ID fallido por intentar marcar asistencia cuando ya se ha marcado asistencia anteriormente.

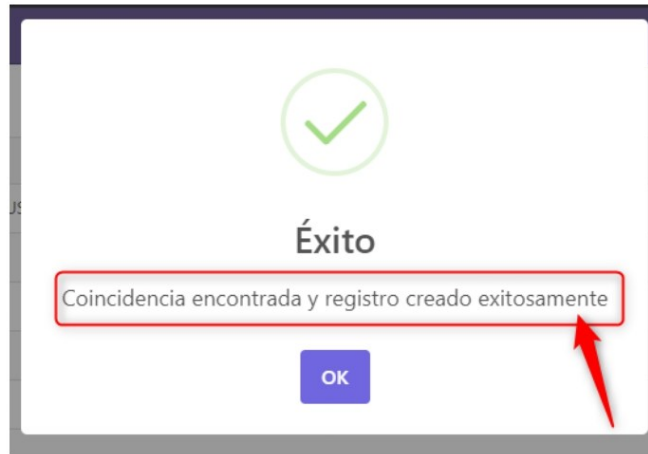


Figura 20. Creación de un nuevo registro de asistencia face ID exitoso.

Una vez realizado estos pasos, ya se puede observar el nuevo registro de asistencia face ID en el sistema.

Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Test Estudiante	Estudiante	Laboratorio de software	15/09/2023 09:36:42	sep-ago-2023	Computación	Noveno	Trabajo de integración Computar	Test Docente	150.96.90.10	e05d49f6-f03a-4076-ae6b-51b05a479a06

Figura 21. Lista de registros de asistencia face ID con el nuevo registro de asistencia face ID en el sistema.

3.2 Editar Asistencia Face ID

No se puede editar un registro de asistencia face ID bajo ningún motivo. El sistema presentará la información de dicho registro en campos deshabilitados.

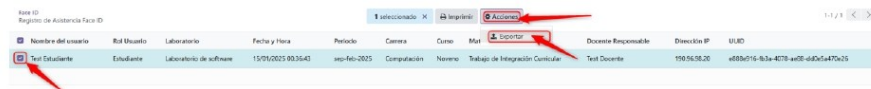
Face ID / Registro de Asistencia Face ID
controlacceso.registroasistenciafaceid,73

Nombre del usuario	Test Estudiante
Rol Usuario	Estudiante
Laboratorio	Laboratorio de software
Fecha y Hora	2025-01-15T00:36:43.000-05:00
Periodo	sep-feb-2025
Carrera	Computación
Curso	Noveno
Materia	Trabajo de Integración Curricular
Docente Responsable	Test Docente
Dirección IP	190.96.98.20
UUID	e888e916-fb3a-4078-ae88-dd0e5a470e26

Figura 22. Seleccionando un registro de asistencia face ID existente.

3.3 Dar de Baja Asistencia Face ID

No se puede dar de baja un registro de asistencia face ID bajo ningún motivo. El sistema presentará la opción de [Acción], pero no presentará la opción de [Suprimir].



Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Test Estudiante	Estudiante	Laboratorio de software	15/01/2025 00:36:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.98.20	e888e916-fb3a-4078-ae88-dd0e5a470e26

Figura 23. Seleccionando un registro de asistencia face ID existente.

3.4 Generar Reporte de Asistencia Face ID

Se selecciona uno o más registros de asistencia face ID.



Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Test Estudiante	Estudiante	Laboratorio de software	15/01/2025 00:36:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.98.20	e888e916-fb3a-4078-ae88-dd0e5a470e26

Figura 24. Seleccionando uno o más registros de asistencia face ID existentes.

Se selecciona [Imprimir] y luego [Reporte de Registros de Asistencia Face ID].



Figura 25. Accediendo a reporte de registros de asistencia face ID.

Se previsualiza el reporte de registros de asistencia face ID generado, y, si se encuentra todo en orden, se selecciona [Imprimir] para poder descargar el archivo en formato PDF.

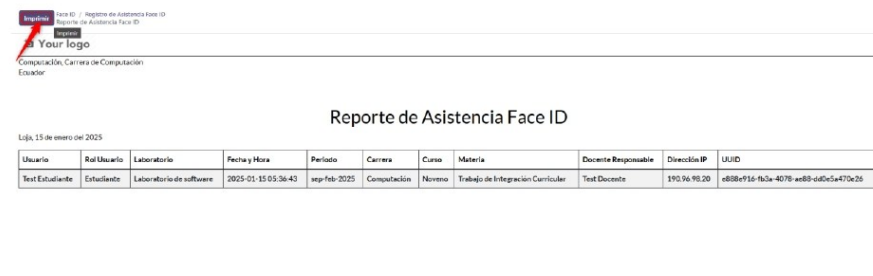


Figura 26. Descargando reporte de registros de asistencia face ID en formato PDF.

Una vez realizado estos pasos, ya se puede observar el archivo PDF descargado con el reporte de los registros de asistencia face ID, para su posterior impresión en físico.

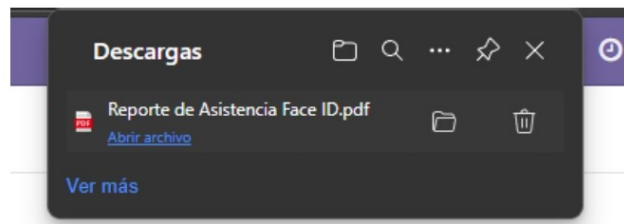


Figura 27. Descarga del reporte de registros de asistencia face ID en formato PDF.

 Your logo

Computación, Carrera de Computación
Ecuador

Reporte de Asistencia Face ID

Hoja 15 de enero del 2025

Usuario	Rol Usuario	Laboratorio	Fecha y Hora	Periodo	Carrera	Curso	Materia	Docente Responsable	Dirección IP	UUID
Test Estudiante	Estudiante	Laboratorio de software	2025-01-15 05:36:43	sep-feb-2025	Computación	Noveno	Trabajo de Integración Curricular	Test Docente	190.96.98.20	e808e916-fb3a-4078-ae88-dd0e5a470e26

Figura 28. Visualización del reporte de registros de asistencia face ID en formato PDF.

Para constancia y validación de la información expuesta, firma a continuación el tutor del Trabajo de Integración Curricular, en su calidad de docente guía en el desarrollo del proyecto:

Ing. Edison Leonardo Coronel Romero, Mg.Sc.



ANEXO 9: Pruebas de validación operativas del sistema en un entorno controlado



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

• **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

• **Revisado y aprobado por:**

- Ing. Edison Leonardo Coronel Romero, Mg.Sc., tutor del Trabajo de Integración Curricular.

Descripción: El presente anexo detalla las pruebas de validación operativas realizadas al sistema "Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja". Estas pruebas se llevaron a cabo en un entorno controlado que replica el escenario real de uso del sistema, con la participación de estudiantes, docentes y clientes directos del proyecto. Dichas pruebas tuvieron como objetivo verificar el correcto funcionamiento del sistema en condiciones reales simuladas, garantizando que se cumplan los requerimientos establecidos. El proceso se estructuró en tres fases principales: registro de usuarios, registro de equipos informáticos y una demostración en vivo de la funcionalidad del software.

PRUEBAS DE VALIDACIÓN OPERATIVAS DEL SISTEMA EN UN ENTORNO CONTROLADO

1 Fase de registro de usuarios

En esta fase, se diseñó un formulario que permitió obtener los datos personales de los participantes, para su posterior registro en el sistema. Los estudiantes y docente del Segundo Ciclo de la Carrera de Computación ingresaron sus nombres, apellidos, correos institucionales y una imagen actualizada de su rostro. En total, se obtuvieron 26 respuestas, lo que permitió crear los 26 perfiles individuales con la información proporcionada dentro del software.

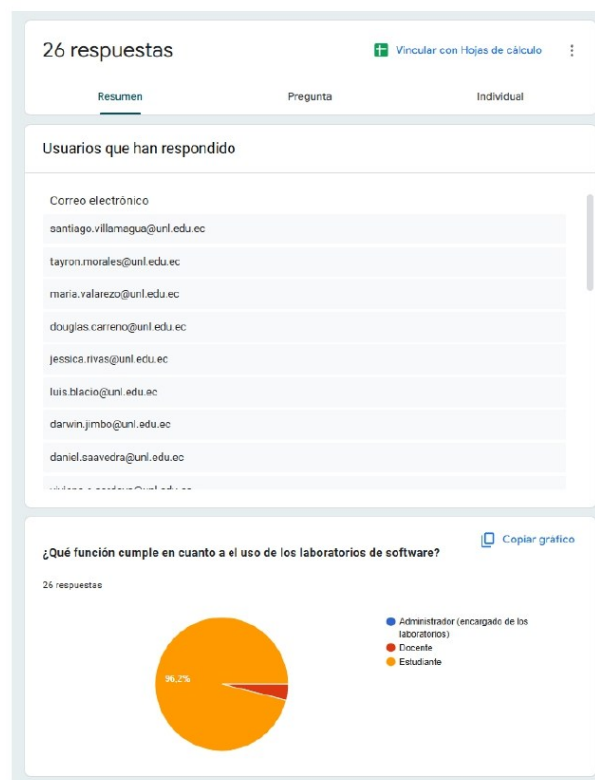


Figura 1. Fragmento de respuestas obtenidas para el registro de usuarios.

2 Fase de registro de equipos informáticos

Se implementó un formulario para registrar los dispositivos que serían utilizados durante las pruebas. Cada participante generó un UUID único desde una página del sistema, que posteriormente ingresaron en el formulario. Esto permitió identificar y asociar cada equipo al usuario correspondiente. En total, se obtuvieron 24 respuestas, registrando 23 equipos de estudiantes y 1 del docente.

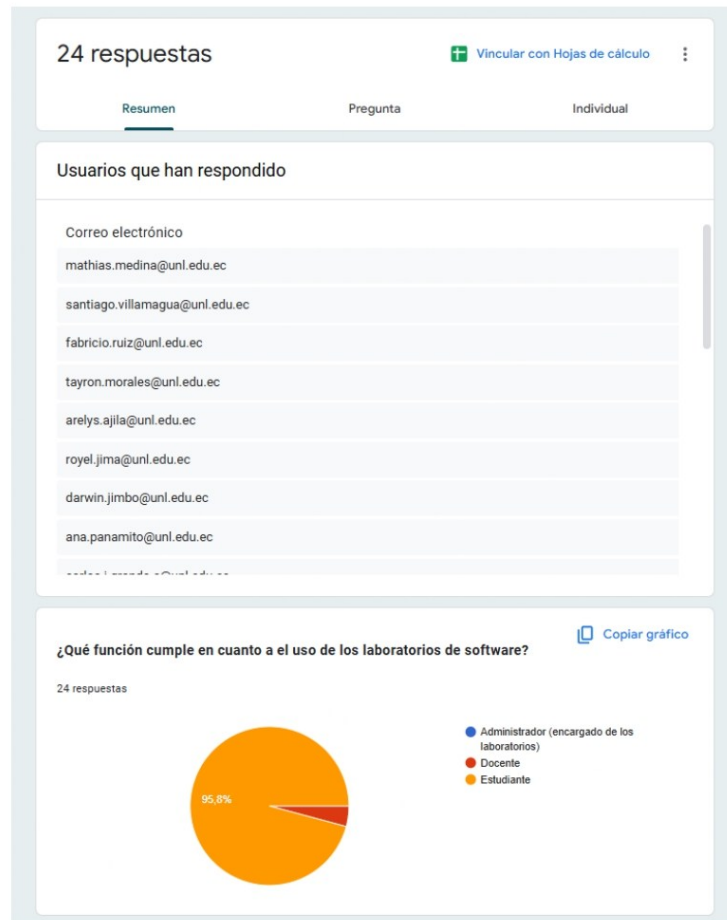


Figura 2. Fragmento de respuestas obtenidas para el registro de equipos informáticos.



Figura 3. Fragmento de respuestas obtenidas para el registro de equipos informáticos - UUID.

3 Fase de demostración en vivo

El día de la demostración, se reunieron todos los participantes en el Laboratorio de Software de la Carrera de Computación. La actividad contó con la presencia del docente a cargo de la materia "Programación Orientada a Objetos", Ing. Edison Coronel; el director de la carrera, Ing. Pablo Ordoñez; el encargado de los laboratorios, Ing. Luis Sinche; y el autor del Trabajo de Integración Curricular (tesista), Luis Jumbo.

Durante la sesión, tanto el docente como los estudiantes ingresaron al laboratorio de software en su respectivo horario, en donde los estudiantes hicieron uso del sistema para registrar sus asistencias. Los resultados obtenidos fueron los siguientes:

- **14 estudiantes** registraron su asistencia correctamente.
- **1 estudiante** no pudo registrar asistencia debido a que la imagen de perfil no coincidía con el rostro escaneado.

- **2 estudiantes** no pudieron marcar asistencia ya que el modelo de detección de vida los identificó erróneamente como “no vivos”.

Esta demostración permitió evaluar la eficacia del sistema en un ambiente controlado y documentar los casos donde el sistema detectó inconsistencias.

Registro de Asistencia Face ID

14 conectados | 14 registros | Acciones

Nombre del usuario	Rol Usuario	Laboratorio	Fecha y Hora	Postura	Cámara	Córeo	Móvil	Docente Responsable	Dirección IP	UUID
Jessica Catalina Klotz Jimenez	Estudiante	Laboratorio de software	14/01/2025 11:27:01	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.200.40	7479466-7854-4036-a048-6991389688
Luiz Alkjanio Elias Torres	Estudiante	Laboratorio de software	14/01/2025 11:23:02	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.204.155	64181956-5343-4518-8c76-527089462226
Cristhian Daughen Valiente Vilaslegua	Estudiante	Laboratorio de software	14/01/2025 11:13:06	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.200.42	36ac3568-8e18-4218-8818-6568866688
Viviana Elizabeth Córdoba Celi	Estudiante	Laboratorio de software	14/01/2025 11:33:08	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.204.119	5f623258-c528-4285-9a21-074176972176
Martín Fabián Medina Aparicio	Estudiante	Laboratorio de software	14/01/2025 11:33:16	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.204.156	471515ac-7955-4564-8a58-16802928107
Miguel Angel Vianemilla Espanza	Estudiante	Laboratorio de software	14/01/2025 11:33:27	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.200.86	94958468-8858-4718-8a10-690206438a8
Yohán Edison Geraza Geraza	Estudiante	Laboratorio de software	14/01/2025 11:33:32	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.204.154	2a65809-ef36-46d2-9444-e716d5215f6c
Fabrizio Isaac Ruiz Aguilar	Estudiante	Laboratorio de software	14/01/2025 11:33:36	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.200.70	482d6688-9e45-4312-9956-8a6689208e
Ana Cristina Panavito Flores	Estudiante	Laboratorio de software	14/01/2025 11:33:36	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.204.157	ff321ce8-6217-4806-9456-c0c992078de
Juan Alberto Calpielo Martínez	Estudiante	Laboratorio de software	14/01/2025 11:34:45	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.200.55	44848519-2758-4897-816c-167f95832b
Sören Daniel Parde Lima	Estudiante	Laboratorio de software	14/01/2025 11:34:47	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.200.75	4100716c-315c-4854-8e47-56628524915
Gerardo David Novia Sanguro	Estudiante	Laboratorio de software	14/01/2025 11:34:48	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.204.156	c81ac737-41c5-4817-acc1-4a804896d395
Daniel Alejandro Sandoza Jaramillo	Estudiante	Laboratorio de software	14/01/2025 11:37:17	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.200.53	5183731f-4828-4423-8a68-05473047a684
Israel Sebastián Rodríguez Torres	Estudiante	Laboratorio de software	14/01/2025 11:38:37	sep-feb-2025	Computación	Sigendo	Programación Orientada a Objetos	Edison Leonardo Cornejo Romero	10.20.204.145	d0ca8176-1a2d-4523-ae82-ee2c771732dc

• **Figura 4. Reporte de asistencia en la demostración en vivos.**



Figura 5. Demostración en vivo.

Para constancia y validación de la información expuesta, firma a continuación el tutor del Trabajo de Integración Curricular, en su calidad de docente guía en el desarrollo del proyecto:

Ing. Edison Leonardo Coronel Romero, Mg.Sc.	 Firma electrónica por: EDISON LEONARDO CORONEL ROMERO
---	--

ANEXO 10: Pruebas de aceptación realizadas al director de la carrera de computación, Ing. Pablo Ordoñez



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

- **Revisado y aprobado por:**

- Ing. Pablo Fernando Ordoñez Ordoñez, Mg.Sc., director de la carrera de Computación y encargado principal del laboratorio de software de la Universidad Nacional de Loja.


Descripción: El presente anexo documenta las pruebas de aceptación finales del sistema, las cuales validan el cumplimiento de todas las historias de usuario planteadas. Cabe destacar que cada historia de usuario corresponde a un requisito funcional del sistema, detallando de manera específica cómo se desarrollará cada uno de ellos. Estas pruebas de aceptación son cruciales para asegurar que el sistema entregue todas las funcionalidades esperadas por el cliente, permitiendo la verificación y aprobación de cada historia de usuario antes de la entrega final. El éxito en estas pruebas garantiza que el sistema cumple completamente con los requerimientos iniciales planteados y validados por el cliente.

PRUEBAS DE ACEPTACIÓN FINALES DEL SISTEMA

Tabla 1. Pruebas de aceptación finales en base a las Historias de Usuario

HU #	Historia de usuario	Estado
HU01	Refactorizar el sistema de control de acceso	Aprobado
HU02	Actualizar gestión de laboratorio	Aprobado
HU03	Obtener UUID	Aprobado
HU04	Registrar equipo informático	Aprobado
HU05	Editar equipo informático	Aprobado
HU06	Dar de baja equipo informático	Aprobado
HU07	Listar equipos informáticos	Aprobado
HU08	Generar reporte de equipos informáticos	Aprobado
HU09	Imprimir reporte de equipos informáticos	Aprobado
HU10	Registrar usuario e imagen de usuario	Aprobado
HU11	Editar usuario e imagen de usuario	Aprobado
HU12	Dar de baja usuario e imagen de usuario	Aprobado
HU13	Listar usuarios	Aprobado
HU14	Registrar asistencia Face ID	Aprobado
HU15	Listar asistencias Face ID	Aprobado
HU16	Generar reporte de asistencias Face ID	Aprobado
HU17	Imprimir reporte de asistencias Face ID	Aprobado

Para constancia y validación de la información expuesta, firma a continuación el director de la carrera de Computación y encargado principal del laboratorio de software de la Universidad Nacional de Loja, en su calidad de cliente principal del proyecto:

<p>Ing. Pablo Fernando Ordoñez Ordoñez, Mg.Sc.</p>	
--	--

ANEXO 11: Pruebas de aceptación realizadas al encargo de los laboratorios de software de la carrera de computación, Ing. Luis Sinche



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

- **Revisado y aprobado por:**

- Ing. Luis Darío Sinche Cueva, Mg.Sc., encargado del laboratorio de software de la carrera de Computación de la Universidad Nacional de Loja.


Descripción: El presente anexo documenta las pruebas de aceptación finales del sistema, las cuales validan el cumplimiento de todas las historias de usuario planteadas. Cabe destacar que cada historia de usuario corresponde a un requisito funcional del sistema, detallando de manera específica cómo se desarrollará cada uno de ellos. Estas pruebas de aceptación son cruciales para asegurar que el sistema entregue todas las funcionalidades esperadas por el cliente, permitiendo la verificación y aprobación de cada historia de usuario antes de la entrega final. El éxito en estas pruebas garantiza que el sistema cumple completamente con los requerimientos iniciales planteados y validados por el cliente.

PRUEBAS DE ACEPTACIÓN FINALES DEL SISTEMA

Tabla 1. Pruebas de aceptación finales en base a las Historias de Usuario

HU #	Historia de usuario	Estado
HU01	Refactorizar el sistema de control de acceso	Aprobado
HU02	Actualizar gestión de laboratorio	Aprobado
HU03	Obtener UUID	Aprobado
HU04	Registrar equipo informático	Aprobado
HU05	Editar equipo informático	Aprobado
HU06	Dar de baja equipo informático	Aprobado
HU07	Listar equipos informáticos	Aprobado
HU08	Generar reporte de equipos informáticos	Aprobado
HU09	Imprimir reporte de equipos informáticos	Aprobado
HU10	Registrar usuario e imagen de usuario	Aprobado
HU11	Editar usuario e imagen de usuario	Aprobado
HU12	Dar de baja usuario e imagen de usuario	Aprobado
HU13	Listar usuarios	Aprobado
HU14	Registrar asistencia Face ID	Aprobado
HU15	Listar asistencias Face ID	Aprobado
HU16	Generar reporte de asistencias Face ID	Aprobado
HU17	Imprimir reporte de asistencias Face ID	Aprobado

Para constancia y validación de la información expuesta, firma a continuación el encargado del laboratorio de software de la carrera de Computación de la Universidad Nacional de Loja, en su calidad de cliente principal del proyecto:

<p>Ing. Luis Darío Sinche Cueva, Mg.Sc.</p>	 <p>Firma digitalizada por: LUIS DARIO SINCHE CUEVA</p>
---	---

ANEXO 12: Reporte de plagio en la herramienta Copyleaks



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

- **Revisado y aprobado por:**

- Ing. Edison Leonardo Coronel Romero, Mg.Sc., tutor del Trabajo de Integración Curricular.

Descripción: El presente anexo contiene el reporte de plagio generado a través de la herramienta CopyLeaks, realizado por el tutor del Trabajo de Integración Curricular (TIC). Este reporte tiene como objetivo evidenciar el porcentaje de similitud detectado en el documento, así como verificar el nivel de inteligencia artificial utilizado. La inclusión de este reporte es fundamental para garantizar la originalidad del trabajo y cumplir con los estándares éticos y académicos establecidos. A través de esta verificación, se asegura que el contenido del TIC es auténtico y cumple con los criterios de calidad exigidos.

REPORTE DE PLAGIO EN LA HERRAMIENTA COPYLEAKS



Analysis Report

Scan details
 Scan Time: December 11th, 2024 at 20:53 UTC
 Total Pages: 95
 Total Words: 23577

Plagiarism Detection Report
 TIC - Luis Jumbo Herrera V14 - Para Reporte Plagio.pdf

Plagiarism Detection



Plagiarism Types	Text Coverage	Words
Identical	1.3%	290
Minor Changes	0.3%	54
Paraphrased	1.1%	228
Excluded		
Omitted Words		2,089

AI Detection



Text Coverage
AI Text
Human Text
Excluded
Omitted Words

Plagiarism



Results (20)

*Results may not appear because the feature has been disabled.



Plagiarism Types	Text Coverage	Words
Identical	13%	290
Minor Changes	0.3%	54
Paraphrased	11%	228
Excluded		
Omitted Words		2,089

About Plagiarism Detection

Our AI-powered plagiarism scans offer three layers of text similarity detection: Identical, Minor Changes, and Paraphrased. Based on your scan settings we also provide insight on how much of the text you are not scanning for plagiarism (Omitted words).

Identical

One to one exact word matches. [Learn more](#)

Paraphrased

Different words that hold the same meaning that replace the original content (e.g. "large" becomes "big"). [Learn more](#)

Minor Changes

Words that hold nearly the same meaning but have a change to their form (e.g. "large" becomes "largely"). [Learn more](#)

Omitted Words

The portion of text that is not being scanned for plagiarism based on the scan settings (e.g. the "ignore quotations" setting is enabled and the document is 20% quotations making the omitted words percentage 20%). [Learn more](#)

Copyrights Internal Database

Our Internal Database is a collection of millions of user-submitted documents that you can utilize as a scan resource and choose whether or not you would like to submit the file you are scanning into the Internal Database. [Learn more](#)









Filtered and Excluded Results









The report will generate a complete list of results. There is always the option to exclude specific results that are not relevant. Note, by unchecking certain results, the similarity percentage may change. [Learn more](#)





Current Batch Results

These are the results displayed from the collection, or batch, of files uploaded for a scan at the same time. [Learn more](#)

🔍 Plagiarism Detection Results: (20)

 <p>BryanJavier_OrdonezJaramillo.pdf https://dspace.unl.edu.ec/jspui/bitstream/123456789/30461/1/bryanjavier_ordonezjaramillo.pdf Bryan Portada Universidad Nacional de Loja facultad de la Energía, las Industrias y los Recursos Naturales No Renovables Carrera de Computació...</p>	0.9%
 <p>Repositorio Digital - Universidad Nacional de Loja: Aplicación web para estim... https://dspace.unl.edu.ec/jspui/handle/123456789/30461 Skip navigation...</p>	0.7%
 <p>IsraelAlexander_CampoverdePenaherrera_EduardoAlexander_LeonCastillo.pdf https://dspace.unl.edu.ec/jspui/bitstream/123456789/27713/1/IsraelAlexander_campoverdepenaherrera_eduar... Israel Campoverde;Eduardo Leon; Israel Campoverde; Eduardo Leon Universidad Nacional de Loja Facultad de la Energía Las Industrias y los Recursos Naturales no Renovables Carrera de Ingeniería en Sistem...</p>	0.6%
 <p>Repositorio Digital - Universidad Nacional de Loja: Aplicación web para estim... https://dspace.unl.edu.ec/jspui/handle/123456789/30461?mode=full Skip navigation...</p>	0.6%
 <p>DayannaMagdalena_AlvaradoCastillo.pdf https://dspace.unl.edu.ec/jspui/bitstream/123456789/27009/1/dayannamagdalla_alvaradocastillo.pdf Dayanna Alvarado tada Universidad Nacional de Loja Facultad de Energía, las Industrias y los Recursos Naturales no Renovables Carrera de Ingeniería en S...</p>	0.3%
 <p>Repositorio Digital - Universidad Nacional de Loja: Prototipo de Reconocimien... https://dspace.unl.edu.ec/jspui/handle/123456789/27009 Skip navigation...</p>	0.3%
 <p>Repositorio Digital Uniandes: Prototipo de reconocimiento facial para mejorar... https://dspace.uniandes.edu.ec/handle/123456789/13197?mode=full Skip navigation...</p>	0.1%
 <p>Repositorio Digital Uniandes: Prototipo de reconocimiento facial para mejorar... https://dspace.uniandes.edu.ec/handle/123456789/13197 Skip navigation...</p>	0.1%

	<p>Repositorio Digital - Universidad Nacional de Loja: Desarrollo del Prototipo ... https://dspace.unl.edu.ec/jspui/handle/123456789/27713 Skip navigation...</p>	0.1%
	<p>Repositorio Digital - Universidad Nacional de Loja: Sistema de reconocimiento... https://dspace.unl.edu.ec/jspui/handle/123456789/14237 Skip navigation...</p>	0.1%
	<p>Plantilla Elaboración de Manuales, Procedimientos e Instrucciones Particulare... https://www.puertovalparaiso.cl/epw/site/docs/20230421/2023042115031/manual_de_usuario__citasiones_v1... Lucero Salinas, Julio Javier Manual de Usuario Citaciones Implementación, Puesta en Marcha y Soporte de los Servicios Asociados al Sistema Logístico de la Comunidad P...</p>	0.1%
	<p>GESTIÓN FINANCIERA Y TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN EN LAS MIPY... https://ri.ujat.mx/bitstream/200.500.12107/4113/1/tesis%20de%20abril%20narez%20sanchez.pdf WinuE UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO DIVISIÓN ACADÉMICA DE CIENCIAS ECONÓMICAS ADMINISTRATIVAS "GESTIÓN FINANCIERA Y TECNOLOGÍAS DE LA ...</p>	0.1%
	<p>Resolución del régimen impositivo para microempresas: Impacto económico en el... https://revistapublicando.org/revista/index.php/crv/article/view/2394 Resolución del régimen impositivo para microempresas: Impacto económico en el sector comercial Revista Publicando ...</p>	0.1%
	<p>Repositorio Digital - Universidad Nacional de Loja: Buscando en DSpace https://dspace.unl.edu.ec/jspui/handle/123456789/9271/browse?type=author&order=asc&ipp=20&value=ort%... Skip navigation...</p>	0.1%
	<p>BomboMaster/unl_tesis_linea_investigacion · Datasets at Hugging Face https://huggingface.co/datasets/bombomaster/unl_tesis_linea_investigacion/viewer Datasets: Bombo Master / unl_tesis_linea_investigacion like 0 ...</p>	0.1%
	<p>La ULPGC celebra el Día Internacional de la Contraseña compartiendo consejos... https://www.eldiario.es/canariasahora/sociedad/ulpgc-celebra-dia-internacional-contrasena-compartiendo-con... Canarias Ahora</p>	0.1%

 A quiénes impactamos - Centro de Pensamiento de Transformación digital CEIPA	0.1%
https://centrodepensamiento digital.org/a-quiénes-impactamos/ Inicio Nosotros ¿Qué hacemos? CEIPA Powered by ASU ¿Qué es un Centro de Pensamiento? ¿A quiénes impactamos? Equipo líder Nuestras form...	
 La ULPGC celebra el Día Internacional de la Contraseña recordando la importan...	0.1%
https://www.ulpgc.es/noticia/2022/05/04/ulpgc-c-celebra-dia-internacional-contrasena-recordando-importan-ci... Ir al contenido principal Ir al menú principal...	
 Cómo ponérselo difícil a los piratas informáticos para que no adivinen nuestr...	0.1%
https://planetacanario.com/como-ponerse-lo-dificil-a-los-piratas-informaticas-para-que-no-adivinen-nuestras-c... Planeta Canario	
 Impulsan proyecto de movilidad integral en Campus Mederos	0.1%
https://www.laredpunto.com.mx/index.php/local/item/31049-impulsan-proyecto-de-movilidad-integral-en-ca... Monterrey *C...	

Para constancia y validación de la información expuesta, firma a continuación el tutor del Trabajo de Integración Curricular, en su calidad de docente guía en el desarrollo del proyecto:

Ing. Edison Leonardo Coronel Romero, Mg.Sc.



ANEXO 13: Documentos firmados electrónicamente



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

- **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

Descripción: El presente anexo proporciona los documentos firmados electrónicamente correspondientes a los anexos presentados en el Trabajo de Integración Curricular. Puesto que en el cuerpo del trabajo se encuentran en formato de imagen, se ha creado una carpeta compartida en Google Drive, donde se almacenan todos los archivos originales en formato PDF con sus respectivas firmas electrónicas. Esta medida permite validar la autenticidad de cada documento, garantizando la veracidad de los mismos al poder verificar electrónicamente las firmas de cada uno de los anexos incluidos.

DOCUMENTOS FIRMADOS ELECTRÓNICAMENTE

Dado que los anexos que poseen firmas electrónicas están presentados en formato de imagen en el presente Trabajo de Integración Curricular, se adjunta la carpeta compartida de Google Drive donde se encuentran cada uno de dichos anexos en formato PDF. Esto permite que se pueda comprobar su veracidad, ya que cada archivo cuenta con su respectiva firma electrónica.

Link a la carpeta compartida de Google Drive:

https://bit.ly/documentos_firmados_electronicamente_TIC

ANEXO 14: Certificado de traducción del resumen



UNIVERSIDAD NACIONAL DE LOJA



Carrera de Ingeniería en
Sistemas / Computación

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

CARRERA DE COMPUTACIÓN

Título del Trabajo de Integración Curricular (TIC): Integración de reconocimiento facial y sistema de Control de Acceso para el registro de asistencia en los laboratorios de software de la Carrera de Computación de la Universidad Nacional de Loja.

• **Elaborado por:**

- Luis Alfredo Jumbo Herrera, autor del Trabajo de Integración Curricular (tesista).

Descripción: El presente anexo contiene el certificado de traducción del resumen del Trabajo de Integración Curricular (TIC), emitido por un profesional en idiomas. Este documento garantiza la fidelidad y precisión de la traducción realizada, asegurando que el contenido refleje correctamente el significado del texto original en el idioma requerido. El certificado acompaña la traducción como respaldo oficial, validando la calidad del trabajo lingüístico y su conformidad con los estándares profesionales.

CERTIFICADO DE TRADUCCIÓN DEL RESUMEN

Loja, 11 de diciembre de 2024

A quien corresponda. -

De mi consideración,

La presente traducción de español a inglés del resumen de la tesis denominada "Integración de reconocimiento facial y sistema de control de acceso para el registro de asistencia en los laboratorios de software de la carrera de Computación de la Universidad Nacional de Loja" de autoría de Luis Alfredo Jumbo Herrera, C.I.: 1105946014, fue realizada y revisada por María Gabriela Jiménez con título de Licenciada en Ciencias de la Educación, mención Idioma Inglés, con número de registro en Senescyt 1008-2016-1754550. En consecuencia, esta traducción se considera válida para ser utilizada con fines académicos.

Particular que informo para los fines pertinentes.

Atentamente,

MARIA
GABRIELA
JIMENEZ
CARRION



Digitally signed by
MARIA GABRIELA
JIMENEZ CARRION

María Gabriela Jiménez

Licenciada en Ciencias de la Educación, mención Idioma Inglés.

Registro en Senescyt: 1008-2016-1754550