



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera de Computación

Front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja"

Front-end version 2.0 for in-situ inventory as a tool for the research project "Dynamics of growth and ecosystem services of urban trees in the city of Loja."

Trabajo de Integración
Curricular previo a la
obtención del título de
Ingeniera en Ciencias de la

AUTOR:

Carmen Azucena Toledo Iñiguez

DIRECTOR:

Ing. Wilman Patricio Chamba Zaragocín, Mg.Sc.

LOJA – ECUADOR

2024

Certificación

Loja, 06 de mayo de 2024

Ing. Wilman Patricio Chamba Zaragocín, Mg.Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo el proceso de la elaboración del Trabajo de Integración Curricular denominado: **Front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”**, previo a la obtención del título de **Ingeniera en Ciencias de la Computación** de autoría de la estudiante: **Carmen Azucena Toledo Iñiguez** con cédula de identidad Nro. **1106088469**, 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.

Ing. Wilman Patricio Chamba Zaragocín, Mg.Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Yo, **Carmen Azucena Toledo Iñiguez**, declaro ser autora 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.

Firma:

Cédula de Identidad: 1106088469

Fecha: 06 de mayo de 2024

Correo electrónico: carmen.toledo@unl.edu.ec

Teléfono: (+593) 995209013

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

Yo, **Carmen Azucena Toledo Iñiguez**, declaro ser autora del Trabajo de Integración Curricular denominado: **Front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”**; como requisito para optar el título de **Ingeniera en Ciencias de la Computación**; 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 de la siguiente manera 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 los 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 seis días del mes de mayo del dos mil veinticuatro.

Firma:

Autor: Carmen Azucena Toledo Iñiguez

Cédula: 1106088469

Dirección: Bo. San Cayetano Alto, Loja - Ecuador

Correo electrónico: carmen.toledo@unl.edu.ec

Celular: (+593) 995209013

DATOS COMPLEMENTARIOS:

Director del Trabajo de Integración Curricular: Ing. Wilman Patricio Chamba Zaragocín, Mg.Sc.

Dedicatoria

A mi familia por su paciencia y amor al guiarme, por sus palabras de aliento y por transmitirme su fortaleza que me ha motivado a seguir esforzándome a pesar de las dificultades.

Carmen Azucena Toledo Iñiguez

Agradecimiento

Agradezco a mi familia, amigos cercanos y a mi pareja por ser mi apoyo y mi guía en los días difíciles.

Un especial agradecimiento al Ing. Wilman Chamba por su valiosa orientación durante el desarrollo del presente Trabajo de Integración Curricular, y a los miembros del proyecto “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” por permitirme ser parte de su equipo y por brindarme la oportunidad de contribuir con mis habilidades y experiencia.

Carmen Azucena Toledo Iñiguez

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
Abstract.....	3
3. Introducción	4
4. Marco Teórico	6
4.1. Antecedentes.....	6
4.2. Revisión comparativa de metodologías ágiles en el desarrollo de software para dispositivos móviles.....	7
4.2.1. Metodología Mobile-D	7
4.2.2. Scrum.....	8
4.2.3. Programación Extrema (XP)	8
4.2.4. Evaluación para la selección de la Metodología de Desarrollo.....	9
4.3. Tecnologías	10
4.3.1. Flutter.....	10
4.3.2. Dart	11
4.4. Arquitectura	11
4.4.1. Arquitectura Limpia.....	11
4.4.2. Patrones de diseño.....	13

4.5.	Pruebas de software.....	15
4.5.1.	Pruebas unitarias.....	15
4.5.2.	Pruebas de aceptación.....	16
4.6.	Modelo de aceptación de tecnología (TAM)	17
4.6.1.	Utilidad estimada	17
4.7.	Trabajos Relacionados.....	18
5.	Metodología	19
5.1.	Área de Estudio	19
5.2.	Procedimiento.....	19
5.2.1.	Fase 1. Desarrollar el front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” en base a los requerimientos actuales del proyecto.	19
5.2.2.	Fase 2. Evaluar la aplicación móvil mediante la utilidad estimada del modelo de aceptación de tecnología (TAM) para determinar el grado de soporte que brinda en el inventario in-situ.	20
5.3.	Recursos.....	21
5.3.1.	Recursos Científicos.....	21
5.3.2.	Recursos de Hardware y Software	21
5.3.3.	Recursos Técnicos	22
5.4.	Participantes	22
6.	Resultados	23
6.1.	Objetivo 1: Desarrollar el front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” en base a los requerimientos actuales del proyecto.....	23
6.1.1.	Fase 1: Exploración.....	23
6.1.2.	Fase 2: Inicialización	27
6.1.3.	Fase 3: Producción.....	33
6.1.4.	Fase 4: Estabilización.....	42

6.2. Objetivo 2: Evaluar la aplicación móvil mediante la utilidad estimada del modelo de aceptación de tecnología (TAM) para determinar el grado de soporte que brinda en el inventario in-situ.....	47
6.2.1. Fase 1: Elaboración de cuestionario de evaluación en base a la utilidad estimada del modelo TAM.....	47
6.2.2. Fase 2: Tabulación de resultados considerando la escala Likert.....	48
6.2.3. Fase 3: Análisis de resultados	51
7. Discusión	55
7.1. Objetivo 1: Desarrollar el front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” en base a los requerimientos actuales del proyecto.....	55
7.2. Objetivo 2: Evaluar la aplicación móvil mediante la utilidad estimada del modelo de aceptación de tecnología (TAM) para determinar el grado de soporte que brinda en el inventario in-situ.....	56
8. Conclusiones.....	58
9. Recomendaciones	59
10. Bibliografía.....	60
11. Anexos.....	64

Índice de Tablas:

Tabla 1. Tabla comparativa de metodologías ágiles	9
Tabla 2. Criterios de selección de metodología	10
Tabla 3. Formato para pruebas unitarias	16
Tabla 4. Escala Likert para la utilidad estimada	17
Tabla 5. Ejemplos resultados de la evaluación TAM.....	18
Tabla 6. Trabajos relacionados	18
Tabla 7. Requerimientos funcionales del sistema	24
Tabla 8. Requerimientos no funcionales del sistema	25
Tabla 9. Ejemplo de historia de usuario	26
Tabla 10. Planificación de fases a través de historias de usuario.....	33
Tabla 11. Validación pruebas unitarias	41
Tabla 12. Pruebas de aceptación en base a las historias de usuario	42
Tabla 13. Características del sistema	42
Tabla 14. Cuestionario para la utilidad estimada del modelo TAM.....	47
Tabla 15. Resultados de la evaluación TAM.....	48
Tabla 16. Resultados de la evaluación TAM por administradores	49
Tabla 17. Resultados de la evaluación TAM por técnicos.....	50
Tabla 18. Resultados de la media ponderada	51

Índice de Figuras:

Figura 1. Arquitectura Limpia.....	12
Figura 2. Estructura del patrón Facade.....	14
Figura 3. Media entre el dominio y las capas de mapeo de datos utilizando repository.....	15
Figura 4. Diagrama de casos de uso del sistema.....	27
Figura 5. Diagrama de componentes.....	28
Figura 6. Diagrama de despliegue.....	29
Figura 7. Interfaces generales.....	30
Figura 8. Interfaces de administración.....	30
Figura 9. Interfaces para agregar árboles.....	31
Figura 10. Interfaces para código QR.....	31
Figura 11. Interfaces para seguridad.....	32
Figura 12. Interfaces para visualizar datos.....	32
Figura 13. Estructura de directorios.....	34
Figura 14. Estructuración del módulo User.....	35
Figura 15. Entidad user.....	36
Figura 16. Repositorio de user.....	36
Figura 17. Caso de uso de user.....	37
Figura 18. Test del caso de uso user.....	37
Figura 19. Modelo de user.....	38
Figura 20. Implementación de repositorio de user.....	38
Figura 21. Test del modelo de user.....	39
Figura 22. Datasource de user.....	39
Figura 23. Test de datasource de user.....	40
Figura 24. Pruebas unitarias.....	40
Figura 25. Aplicación móvil disponible en Google Play.....	43
Figura 26. Interfaces para visualizar información.....	44
Figura 27. Interfaces para administrar y visualizar registros.....	44
Figura 28. Interfaces de inicio de sesión y registro.....	45
Figura 29. Diagrama de entregables de la metodología Mobile-D.....	46
Figura 30. Cuestionario para evaluar la utilidad estimada en Google Forms.....	48
Figura 30. Resultados de la evaluación TAM por administradores.....	49
Figura 32. Resultados de la evaluación TAM por técnicos.....	50
Figura 33. Pregunta UE01.....	52
Figura 34. Pregunta UE02.....	52

Figura 35. Pregunta UE03.....	53
Figura 36. Pregunta UE04.....	54
Figura 37. Pregunta UE05.....	54
Figura 38. Resultados de la evaluación TAM.....	57

Índice de Anexos:

Anexo 1. Entrevista para la elicitación de requerimientos.....	64
Anexo 2. Requerimientos del sistema y prototipado de interfaces de usuario	64
Anexo 3. Pruebas unitarias	64
Anexo 4. Pruebas de aceptación	64
Anexo 5. Resultados de la evaluación de la utilidad estimada del modelo TAM.....	64
Anexo 6. Manual de usuario	64
Anexo 7. Manual de Desarrollador	64
Anexo 8. Acta de entrega del sistema	64
Anexo 9. Informe de pertinencia del proyecto de Trabajo de Integración Curricular	64
Anexo 10. Certificado de traducción del resumen.....	64

1. Título

Front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”

2. Resumen

La realización de un inventario in-situ de árboles en entornos urbanos se presenta como una herramienta valiosa para los proyectos dendrométricos, que se enfocan en la medición de propiedades físicas y biológicas de los árboles. En el contexto local de la ciudad de Loja, se llevó a cabo un estudio titulado “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”; en donde se desarrolló una versión prototipo de la aplicación que quedó en desuso debido a diversos problemas identificados. Considerando este antecedente, surgió la necesidad del desarrollo de una nueva aplicación móvil con el propósito de abordar los requerimientos actuales del proyecto. Por lo que, el presente Trabajo de Integración Curricular (TIC) tiene como objetivo principal determinar el grado de soporte que brinda el front-end versión 2.0 para el inventario in situ como herramienta del proyecto mencionado. Para guiar la ejecución del TIC se utilizó la metodología Mobile-D, dividida en cuatro fases: exploración, inicialización, producción y estabilización. De las cuales se obtuvo los requerimientos del sistema, el diseño de la aplicación, el desarrollo dividido en fases y guiado por pruebas, con la posterior entrega del sistema a los interesados del proyecto. Posteriormente, para resolver el cuestionamiento planteado, se utilizó la utilidad estimada del modelo de aceptación de tecnología (TAM); con el cual se pudo determinar que la aplicación móvil brinda un grado alto de soporte para el inventario in-situ como herramienta del proyecto de investigación.

Palabras claves: Flutter, Mobile-D, arquitectura limpia, TDD, TAM, árboles qr.

Abstract

The conduction of an in-situ inventory of trees in urban environments is presented as a valuable tool for dendrometric projects, which focus on measuring the physical and biological properties of trees. In the local context of the city of Loja, a study entitled "Dynamics of growth and ecosystem services of urban trees in the city of Loja" was carried out, where a prototype version of the application was developed, which fell into disuse due to various problems identified. Considering this background, the need arose for the development of a new mobile application in order to address current requirements of the project. Therefore, the main objective of the present Curricular Integration Work (TIC) is to determine the degree of support provided by the front-end version 2.0 for the in-situ inventory as a tool for the aforementioned project. To guide the execution of the TIC, the Mobile-D methodology was used, divided into four phases: exploration, initialization, production, and stabilization. From which the system requirements were obtained, the application design, the development divided into phases and guided by tests, with the subsequent delivery of the system to the project stakeholders. Subsequently, to solve the question posed, the estimated usefulness of the technology acceptance model (TAM) was used; with which it was determined that the mobile application provides a high degree of support for the in-situ inventory as a tool for the research project.

Keywords: Flutter, Mobile-D, clean architecture, TDD, TAM, qr trees.

2. Introducción

El inventario in-situ es una técnica utilizada para identificar y contar los bienes o recursos en su ubicación actual, sin tener que moverlos [1]. Un inventario in-situ de árboles urbanos es una herramienta valiosa para los proyectos dendrométricos, que son aquellos que miden las propiedades físicas y biológicas de los árboles. Este tipo de inventario involucra la identificación, medición y registro de los árboles en su lugar de crecimiento en la ciudad [2]. Por lo tanto, implica la pérdida de recursos humanos, materiales y económicos que derivan de la complejidad del levantamiento de estos datos.

En Ecuador se han desarrollado varios proyectos en torno a esta situación, uno de ellos es el proyecto denominado “Sistema de inventario forestal de parques urbanos de la ciudad de Cuenca”; dicha ciudad, según la información más actualizada al momento de realizar la investigación, contaba con un área verde de 1.82 metros cuadrados por habitante, con un inventario de 44 parques dentro del área urbana. La dificultad a la que se enfrentó la investigación radicó en este último punto, en la complejidad de inventariar in-situ los datos y mantenerlos actualizados, por la extensión del área del arbolado urbano [3].

A nivel local, en la ciudad de Loja, se ha llevado a cabo una investigación denominada “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” (vea el **Anexo 1**). Según el Plan de Desarrollo General de la ciudad de Loja, actualizado en el año 2021, la ciudad cuenta con una superficie total de 273 hectáreas de áreas verdes, lo que corresponde a 15.11 metros cuadrados de áreas verdes por habitante [4]. En la investigación local mencionada, realizan una toma manual de datos en los que identifican la geoubicación del árbol en cuestión y propiedades relacionadas con su morfología, fitopatologías, entre otros. Conforme la investigación avanza, el volumen de datos crece a su par y el levantamiento de datos in-situ se vuelve más complejo, generando el consumo de recursos económicos, de tiempo y personal.

Esta versión prototipo de la aplicación quedó en desuso debido a varios problemas identificados. Uno de los principales inconvenientes radicó en la incapacidad de la aplicación para gestionar eficientemente los datos creados tras la reestructuración de la Base de Datos. Además, la interfaz de usuario presentaba una navegación complicada y múltiples detalles que dificultaban su utilización. También se encontraron problemas de compatibilidad con distintos modelos de teléfonos móviles, lo que afectaba el acceso a la aplicación. Asimismo, el tamaño de descarga de la aplicación, aproximadamente 250 megabytes, la hacía pesada y poco práctica. Por último, la aplicación no estaba en producción y requería una instalación

manual. Por tanto, se requiere el desarrollo de una nueva aplicación que sea funcional, amigable para el usuario y compatible con diversos dispositivos. Si no se cuenta con una herramienta informática o con el apoyo de tecnología para realizar el levantamiento in-situ, el proceso de recolección de datos podría ser propenso a errores humanos y omisiones involuntarias.

Considerando lo expuesto con anterioridad, el presente trabajo de integración curricular (TIC) plantea el desarrollo de una aplicación informática con el propósito de abordar la interrogante de investigación: **¿En qué grado el front-end versión 2.0 brinda soporte para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”?** Con el fin de responder la pregunta de investigación, se propuso el siguiente objetivo general: Determinar el grado de soporte que brinda el front-end versión 2.0 para el inventario in situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”. Y para ejecutarlo, se desarrollaron dos objetivos específicos “Desarrollar el front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” en base a los requerimientos actuales del proyecto”; y “Evaluar la aplicación móvil mediante la utilidad estimada del modelo de aceptación de tecnología (TAM) para determinar el grado de soporte que brinda en el inventario in-situ”.

Para la ejecución del primer objetivo específico se empleó la metodología Mobile-D, dividida en cuatro fases: exploración, inicialización, producción y estabilización. En cuanto al segundo objetivo, se utilizó la utilidad estimada del modelo de aceptación de tecnología (TAM), que se dividió en tres partes: elaboración de cuestionario, tabulación y análisis de resultados. De esta manera, el presente TIC se compone por las siguientes secciones: en el **Marco Teórico** se encuentran los antecedentes, una revisión comparativa de metodologías para el desarrollo móvil, las tecnologías utilizadas, y conceptos generales para sustentar la investigación; en la **Metodología** está presente el área de estudio, el procedimiento que guió el desarrollo del TIC, los recursos necesarios para su ejecución y los participantes; la sección de **Resultados** se divide en dos fases para exponer lo logrado en cada uno de los objetivos específicos; la **Discusión** expone el análisis de los resultados obtenidos desde la perspectiva de la autora del TIC; la sección de **Conclusiones** aborda los hitos destacables del proyecto; finalmente, las **Recomendaciones** detallan las sugerencias para trabajos futuros.

3. Marco Teórico

En esta sección se exponen los fundamentos teóricos que respaldan el presente TIC, los cuales se basan en la literatura especializada. Para ello, en la sección **3.1** se mencionan los antecedentes sobre los que se sustenta la problemática del TIC. En la sección **3.2** se detallan las metodologías mediante una revisión comparativa de metodologías ágiles en el desarrollo de software para dispositivos móviles. La sección **3.3** aborda las tecnologías empleadas en el desarrollo del TIC como Flutter y Dart. Por su parte, la sección **4.5** detalla las pruebas de software de caja negra y de aceptación, que serán aplicadas al término del desarrollo del sistema. De igual manera, la sección **4.6** se expone el modelo de aceptación de tecnología, conocido como TAM, que se aplicará en el desarrollo del segundo objetivo del TIC. Por último, se abordan trabajos relacionados con la temática de estudio.

3.1. Antecedentes

En el caso del desarrollo de una aplicación para el inventario in-situ de árboles urbanos, se han encontrado varias investigaciones que abordan la problemática del levantamiento de datos in-situ, que es una técnica que identifica y cuenta los bienes o recursos en su ubicación actual sin tener que moverlos. Se ha comprobado que el levantamiento de datos in-situ de árboles urbanos es una tarea compleja que consume muchos recursos humanos, materiales y económicos.

Se han realizado varios proyectos de inventario forestal en áreas urbanas en Ecuador, incluyendo el “Sistema de inventario forestal de parques urbanos de la ciudad de Cuenca” y la investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”. En ambas investigaciones se ha encontrado que la toma de datos in-situ es compleja y requiere muchos recursos, lo que ha llevado a la búsqueda de herramientas tecnológicas que faciliten la recopilación y el almacenamiento de datos [1] (vea el **Anexo 1**).

El proyecto “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” busca desarrollar una aplicación móvil que permita almacenar y consultar información sobre los árboles de la ciudad, lo que permitiría crear un catastro urbano de los árboles de la ciudad y mejorar la gestión y conservación del arbolado urbano. Se ha desarrollado una versión prototipo de la aplicación móvil, pero se requiere seguir trabajando en una nueva versión, para mejorar la funcionalidad y la calidad de la información recolectada (vea el **Anexo 1**).

3.2. Revisión comparativa de metodologías ágiles en el desarrollo de software para dispositivos móviles

La metodología ágil es un enfoque práctico para el desarrollo de software, tanto en términos de modelado como de documentación. Los principios y prácticas de esta metodología son más flexibles en comparación con las metodologías tradicionales, lo que facilita su aplicación en proyectos de desarrollo de software. Además, la metodología ágil ofrece diversas formas de estructurar, comunicar y organizar los equipos de trabajo [5][6]

Esta metodología divide un proyecto en subproyectos de manera organizada, asegurando que sean independientes entre sí y tengan una duración de dos a seis semanas. La flexibilidad es un aspecto fundamental en esta metodología, ya que los equipos de trabajo se adaptan a los cambios que surgen en el proyecto. Además, se destaca la importancia de una comunicación continua con el cliente a lo largo de todo el proceso de desarrollo del software [7][8].

3.2.1. Metodología Mobile-D

La metodología Mobile D es un enfoque específico del desarrollo de aplicaciones móviles. Se centra en la entrega temprana y frecuente de versiones funcionales de la aplicación móvil y en la información continua a los usuarios [9].

Basada en principios ágiles, la metodología móvil utiliza técnicas como el desarrollo iterativo e incremental y la entrega continua para garantizar que la funcionalidad de una aplicación móvil se entrega de forma rápida y eficiente. Los equipos de desarrollo móvil también trabajan en estrecha colaboración con los usuarios finales para garantizar que la aplicación satisfice sus necesidades y expectativas [9] [10].

El proyecto de desarrollo se divide en cinco iteraciones siguiendo el enfoque Mobile-D. Estas fases son: lanzamiento, núcleo, núcleo2, estabilización y cierre. Cada fase consta de tres tipos diferentes de días de desarrollo: día de planificación, día de trabajo y día de lanzamiento. La práctica de las distintas fases incluye nueve elementos clave. Estos elementos son: fases y tiempos, línea arquitectónica, desarrollo móvil basado en pruebas, integración continua, programación por parejas, métricas, mejora ágil del proceso de software, cliente externo, enfoque centrado en el usuario. La mayoría de estos elementos son prácticas de la metodología ágil bien conocidas y especializadas para el desarrollo de software móvil [9].

Fases de Mobile-D

Según [11] las fases de Mobile-D son:

- **Exploración:** En la primera etapa se definen los objetivos de la aplicación definiendo los requisitos iniciales, el plan del proyecto, la descripción de los procesos y el plan de capacitación.
- **Inicialización:** En esta etapa se genera la arquitectura del software, se modifican los requisitos iniciales, se crean las interfaces de usuario y los diagramas de casos de uso.
- **Producción:** En esta etapa, se divide en tres partes: los días de planificación se centran en mejorar y analizar los objetivos, planificar la iteración y definir las pruebas de aceptación; los días laborales se dedican al desarrollo guiado por pruebas; y, por último, los días de lanzamiento, en los que se presenta al cliente una versión funcional para llevar a cabo las pruebas de aceptación.
- **Estabilización:** Esta etapa consiste en la integración del sistema completo de forma que se genere el software completo funcionando y la documentación terminada.

3.2.2. Scrum

Scrum es un marco de trabajo para la gestión de proyectos y productos de software, diseñado especialmente para proyectos que experimentan cambios constantes en sus requisitos. Se basa en iteraciones llamadas "sprints", que tienen una duración de una a cuatro semanas. Cada sprint se ejecuta de forma consecutiva y se presenta el resultado al cliente. Scrum se caracteriza por la realización de diversas reuniones con el objetivo de asegurar el cumplimiento de los objetivos. Estas reuniones incluyen la planificación de entrega, la planificación del sprint, las reuniones diarias, de revisión y de retrospectiva [12][13].

3.2.3. Programación Extrema (XP)

La Programación Extrema, también conocida como XP y desarrollada por Kent Beck, Ward Cunningham y Ron Jeffries, es una metodología ágil que se enfoca en la comunicación continua entre el cliente y el equipo de trabajo. Al igual que Scrum, es adecuada para proyectos con cambios constantes o requisitos poco precisos. Esta metodología requiere la participación activa del cliente en el proyecto y se recomienda que sean expertos en el área del producto a desarrollar, además de ser usuarios reales del software [13].

El ciclo de vida de XP según [14] es el siguiente:

- **Exploración:** En esta fase el cliente presenta el producto deseado, el equipo de desarrollo define las tecnologías, herramientas y patrones a usar.
- **Planificación de entrega:** Durante esta fase el cliente determina la prioridad de las historias de usuario mientras el equipo de desarrollo realiza la estimación de esfuerzo para cada una de ellas con una puntuación de 1 a 3 que representan el número de semanas para el desarrollo.
- **Iteraciones:** Una vez definidas las historias de usuario las cuales pasan a ser iteraciones. Primeramente, se seleccionan historias de usuario relacionadas con la arquitectura y al finalizar con la última interacción el sistema está listo para producción.
- **Producción:** En esta fase se llevan a cabo pruebas adicionales tales como pruebas de rendimiento o de carga, antes de ser entregada al cliente.
- **Mantenimiento:** Mientras se pasa a producción la primera versión del sistema y se crean nuevas iteraciones para funcionalidades adicionales, se debe asegurar que las funcionalidades existentes se mantengan para lo que se realizan tareas de soporte.
- **Muerte del proyecto:** Esta última fase del proyecto ocurre cuando el cliente no presenta nuevas historias de usuario, en esta fase se genera la documentación final. La finalización del proyecto también se puede dar si el sistema no cumple con lo esperado por el cliente o por falta de presupuesto.

3.2.4. Evaluación para la selección de la Metodología de Desarrollo

En la **Tabla 1** se realiza una comparación de las metodologías ágiles mencionadas anteriormente, teniendo en cuenta su tipo, nivel de documentación, grado de conocimiento por parte del equipo de trabajo y por último el entorno en el que se aplica la metodología ya sea móvil o web.

Tabla 1. Tabla comparativa de metodologías ágiles

Metodología	Tipo	Documentación	Grado de conocimiento	Ambiente
Mobile-D	Ágil	Destaca la motivación, los principios básicos, y el enfoque pragmático [15].	Alto	Móvil
Scrum	Ágil	Destaca el sprint, los roles, las reuniones, los principios y herramientas [12].	Bajo	Web, Móvil
Programación Extrema XP	Ágil	Destacan los principios, prácticas, ciclo de desarrollo, roles y la mejora continua [16].	Medio	Web, Móvil

En base a [17] se llevó a cabo una evaluación de metodologías scrum, programación extrema y Mobile-D como se muestra en la **Tabla 2**, en la que se asigna una calificación a cada metodología en función de ocho criterios específicos, los cuales fueron ponderados con un porcentaje máximo asignado a cada criterio, tal como se muestra en la segunda columna.

Tabla 2. Criterios de selección de metodología

Criterio	%	Scrum	XP	Mobile-D
Grado de conocimiento	20	10	10	15
Soporte orientado a objetos	10	5	10	10
Adaptable a cambios	15	10	15	15
Basado en casos de uso	10	5	5	10
Posee documentación adecuada	15	15	15	15
Facilita la integración entre las etapas de desarrollo	10	10	10	10
Relación con UML	10	5	8	8
Permite desarrollo software sobre cualquier tecnología	10	10	10	8
Total	100	70	83	91

Según los resultados presentados en la **Tabla 2** Mobile-D es la metodología más adecuada para utilizar principalmente debido al grado de conocimiento del equipo de trabajo, su adaptabilidad a cambios y su relación con los casos de uso.

3.3. Tecnologías

3.3.1. Flutter

Flutter se define como un conjunto de herramientas de interfaz de usuario portátil para crear aplicaciones compiladas de forma nativa para móviles, web y escritorio a partir de una única base de código. Una de las principales características de Flutter es la capacidad de crear interfaces de usuario atractivas y personalizadas utilizando widgets, que son elementos visuales reutilizables que se combinan para crear la interfaz de usuario. Flutter proporciona una amplia biblioteca de widgets que pueden ser utilizados para crear interfaces de usuario de alta calidad [18].

Otra característica importante de Flutter es que utiliza el motor de renderizado de Skia, que es un motor de renderizado de alta calidad utilizado por Google en sus productos. Esto permite que las aplicaciones desarrolladas con Flutter tengan un alto rendimiento y sean altamente responsivas [18]. El SDK de Flutter es la única herramienta indispensable para trabajar con Flutter, esta incluye el compilador Flutter, creador de proyectos, administrador de dispositivos, ejecutor de pruebas y herramientas que diagnostican, e incluso corrigen, problemas con la configuración de Flutter [19].

3.3.2. Dart

El lenguaje de programación Dart fue creado inicialmente por Google para su uso interno en algunos de sus productos más grandes, y se ha hecho disponible públicamente desde 2011. Dart se utiliza ampliamente para el desarrollo de aplicaciones móviles, web y de servidor. Se caracteriza por ser rápido, portátil, accesible y especialmente reactivo, lo que significa que es capaz de responder rápidamente a las interacciones de los usuarios [20].

Dart es un lenguaje de programación orientado a objetos con una sintaxis similar al estilo C diseñado de forma que sea familiar para la mayoría de programadores, utiliza un enfoque basado en clases. Una de las características clave de Dart es su capacidad para compilar el código de manera previa, lo que permite que las aplicaciones Flutter que utilizan Dart sean altamente eficientes y rápidas, ya que no requieren interpretaciones ni puentes para traducir el código a otro lenguaje [21].

3.4. Arquitectura

3.4.1. Arquitectura Limpia

Entre las diferentes arquitecturas de software tales como la hexagonal o también llamada Puertos y adaptadores, DCI de James Coplien o BCE de Ivar Jacobson, estas arquitecturas comparten similitudes en su objetivo que es la separación de responsabilidades dividiendo el software en capas, separando las reglas del negocio, la interfaz de usuario y el sistema. Estas arquitecturas siguen características como: independencia de marcos de trabajos es decir la arquitectura no depende de bibliotecas tomándolas como herramientas de trabajo, otra característica es la testeabilidad es decir las reglas del negocio pueden ser probadas independientemente de la existencia de interfaz de usuario, servidor web u otros elementos externos. Con el fin de integrar estas arquitecturas [22] el autor de la arquitectura limpia como se muestra en la **Figura 1** en la que los círculos internos representan las reglas del negocio mientras los externos los mecanismos. Esta arquitectura tiene como regla principal la dependencia como por ejemplo la dependencia del código fuente que debe apuntar únicamente a los círculos internos además no deben depender de los círculos externos esto incluyendo funciones, clases, variables o cualquier entidad del software, esto con el fin de que una implementación o componente del círculo externo no afecta a un círculo interno.

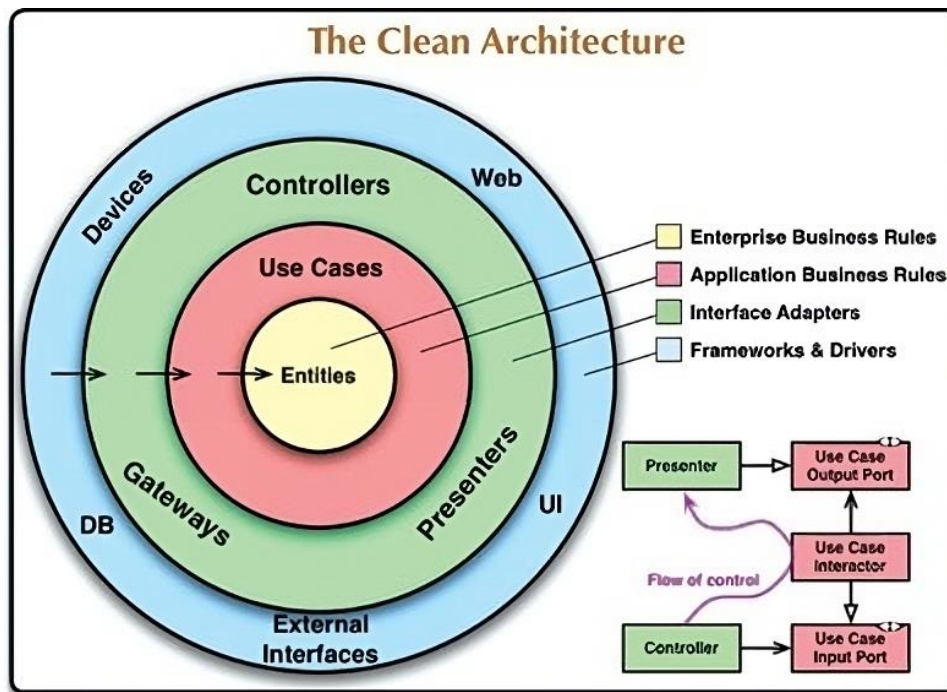


Figura 1. Arquitectura Limpia [22].

- **Entidades:** Como núcleo de la arquitectura contienen las reglas críticas del negocio, una entidad puede ser un objeto que encapsula reglas generales y de alto nivel las que no deben verse afectadas cuando algo de los círculos externos cambien.
- **Casos de uso:** Esta capa contiene reglas específicas de la aplicación, esta capa encapsula y aplican los casos de uso del sistema administrando el flujo de datos desde y hasta las entidades con el fin de cumplir el objetivo de los casos de uso.
- **Adaptadores de interfaz:** Esta capa se encarga de convertir los datos en una forma conveniente para las capas de casos de uso o entidades, así como las bases de datos o servicios web. Esta capa incluye la arquitectura Modelo-Vista-Controlador (MVC) tomando a los modelos simplemente como estructuras de datos que se envían desde los controladores a los casos de uso y vuelven hacia los presentadores y vistas.
- **Frameworks y Drivers:** En la última capa exterior contiene elementos como la base de datos o la web, esta capa contiene únicamente código de enlace para la comunicación con los círculos internos [23].

3.4.2. Patrones de diseño

- **Arrange Act Assert**

Este patrón más conocido como AAA se basa en tres principios los cuales son: organizar, actuar y afirmar, además presenta una estructura uniforme y simple para las pruebas de software. Su estructura va desde organizar que consiste en colocar el sistema bajo prueba (SUT) definiendo las dependencias en un estado deseado, en la sección de actuar se llama a los métodos SUT y por último la sección de afirmar verifica el resultado ya sea por el valor de retorno o el estado del SUT. Este patrón es similar al patrón Given-When-Then es decir dado un estado inicial cuando ocurre un evento, entonces se describe el resultado esperado. Generalmente se recomienda iniciar siempre por la fase de organizar, aunque no siempre es aplicable. Por ejemplo, en situaciones cuando no se conoce lo que se va a probar, como en el Desarrollo Guiado por Pruebas (Test-Driven Development, TDD) en la que se crea una prueba que fallara antes de iniciar el desarrollo. De manera similar con el patrón AAA se puede iniciar por la fase de afirmación definiendo como se espera que funcione el sistema y luego continuar con el desarrollo [24].

- **Facade**

Como patrón estructural, el patrón facade o fachada se caracteriza por su capacidad de enmascarar su complejidad de la interacción con un grupo de subsistemas mediante el uso de una interfaz de alto nivel encargada de la comunicación entre los subsistemas. Este patrón es útil en momentos que se requiera interactuar con distintos subsistemas para llevar a cabo una operación reduciendo la complejidad de los sistemas [25].

La estructura del patrón Facade según [25] en la **Figura 2** se presenta con los siguientes elementos:

- **IFacade**: Es la interfaz que enmascara la complejidad al interactuar con distintos sistemas.
- **Client**: Sistema que interactúa con IFacade.
- **DefaultFacadeImpl**: Es la implementación de IFacade realizando la comunicación con los sistemas.
- **Subsystems**: Son los subsistemas o componentes que ofrecen las interfaces facilitando la comunicación.

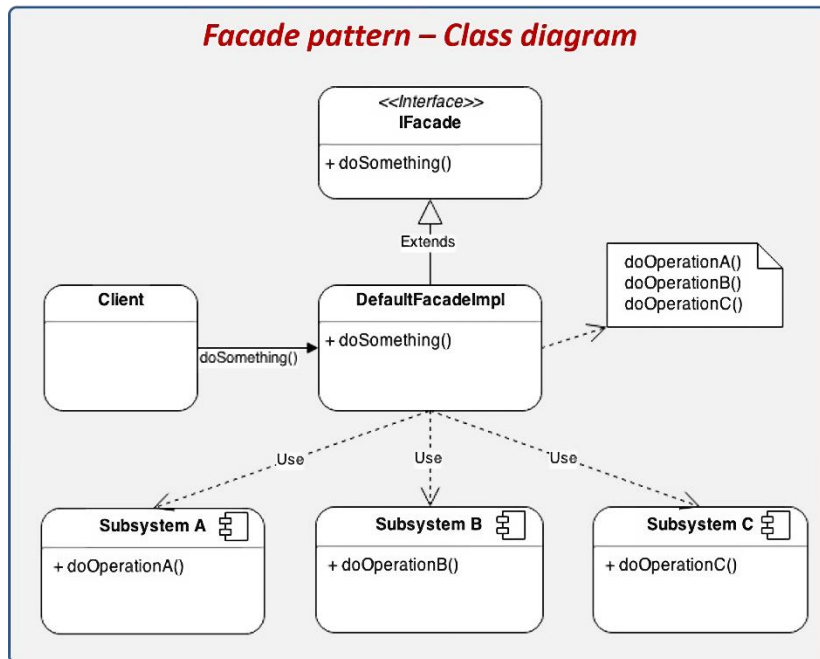


Figura 2. Estructura del patrón Facade [25].

- **Repository**

En un sistema complejo a menudo se lo resuelve agregando una capa de mapeo de datos como se muestra en la **Figura 3** de forma que se aíslan los objetos de dominio con el código de acceso con la base de datos, para reducir aún más la complejidad de estos sistemas agregando una capa de abstracción entre la capa de dominio y de mapeo, esta capa se la conoce como repository que actúa como intermediario entre el código o dominio de la aplicación y el código encargado del acceso a la base de datos o mapeo de datos. Por ejemplo, cuando el cliente necesita realizar una consulta a la base de datos en lugar de usar un código complejo realiza una consulta describiendo la información que requiere y enviándola al repositorio el cual lleva a cabo la consulta con la base de datos, este patrón oculta la complejidad del acceso a la base de datos mediante una interfaz más sencilla [26].

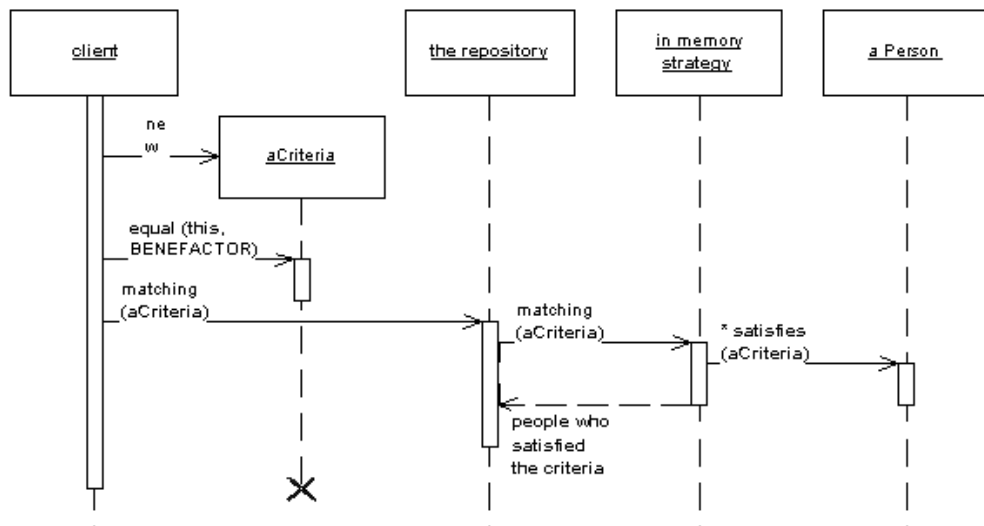


Figura 3. Media entre el dominio y las capas de mapeo de datos utilizando repository [26].

4.5. Pruebas de software

Las pruebas de software son un conjunto de actividades y técnicas utilizadas para evaluar y verificar el rendimiento y la calidad del software antes de comercializarlo o implantarlo en una empresa u organización. El objetivo principal es detectar y notificar fallos o errores en el software, y garantizar que éste cumple los requisitos y expectativas de los usuarios y las especificaciones del proyecto. Se cree erróneamente que probar un software es simplemente ejecutar el software y comprobar su comportamiento, la pruebas incluye una planificación, análisis, diseño, implementación y evaluación de calidad. Hay distintos tipos de pruebas de software, como las pruebas de caja negra, las pruebas de aceptación, etc. Cada tipo de prueba tiene una finalidad y un enfoque específicos a la hora de evaluar el software [27].

4.5.1. Pruebas unitarias

Las pruebas unitarias son una parte esencial en el proceso de desarrollo de software en el cual se somete a pruebas individuales a cada módulo por parte del programador. Estas pruebas se basan en pruebas de caja blanca de forma que se prueban los bloques más pequeños del software detectando errores de forma temprana, se enfoca en la lógica interna de cada componente. Para realizar este tipo de prueba se crea un módulo conductor para realizar la prueba el cual llama al módulo a evaluar de forma que las pruebas se realizan de forma aislada utilizando parámetros reales [28][29].

En base a [30] se definió la **Tabla 3** para la elaboración de las pruebas unitarias.

Tabla 3. Formato para pruebas unitarias

Prueba Unitaria			
N°:	PU-001	Versión de ejecución:	-
Componente:			-
Caso de prueba:			-
Descripción del caso:			-
Observaciones.			-
Método:			-
Descripción	Datos de entrada	Salida esperada	Resultado
-	-	-	-

Prueba Unitaria (PU): Técnica para validar un fragmento de código.

4.5.2. Pruebas de aceptación

Como última etapa antes de realizar una entrega formal del software se realiza para determinar si el software cumple con los requisitos y expectativas del usuario o del cliente. Durante las pruebas de aceptación, se evalúa el software desde la perspectiva del usuario final. Esto implica comprobar que el software cumple con las expectativas del usuario en cuanto a los requerimientos funcionales y no funcionales. Una vez se completan las pruebas de aceptación satisfactoriamente se realiza la entrega del sistema o aplicación [28].

La forma más común de realizar estas pruebas es:

- **Prueba de aceptación de usuario:** en este caso las pruebas las realizan los usuarios validando las funcionalidades del sistema con base en sus necesidades y procesos de negocio.
- **Prueba de aceptación operativa:** esta prueba se realiza por parte del personal administrativo del sistema realizando pruebas de restauración, instalación migración, actualización, mantenimiento o gestión de usuarios.
- **Prueba de aceptación contractual:** se realiza con base en los criterios de aceptación del contrato, este tipo de pruebas generalmente son realizadas por los usuarios o probadores independientes [27].

4.6. Modelo de aceptación de tecnología (TAM)

El modelo de aceptación TAM es un modelo de investigación desarrollado para explicar y predecir el comportamiento del usuario en relación con la adopción y uso de tecnologías. Ayuda a determinar si los usuarios aceptan o rechazan un software proporcionando una base a la utilidad estimada y la facilidad de uso percibida. TAM establece que las convicciones, actitudes, interacciones y acciones de una persona están interrelacionadas y son fundamentales al momento de aceptar o rechazar un software. Según el modelo, la intención de un usuario de utilizar una tecnología de la información está determinada por su percepción de que la tecnología es útil para su trabajo y que es fácil de usar [31], [32].

4.6.1. Utilidad estimada

La utilidad estimada se refiere a la percepción de que una tecnología de la información mejora la productividad y ayuda en la realización de tareas [33]. Se refiere al grado en el que una persona percibe que el uso de un sistema mejora el desempeño de sus actividades [32]. Una forma para medir la utilidad estimada es mediante el uso de la escala Likert como se presenta en la **Tabla 4** se basa en el nivel de acuerdo o desacuerdo de una persona con relación a la utilidad de un software [31].

Tabla 4. Escala Likert para la utilidad estimada

Respuesta	Valor asignado	Escala
Totalmente en desacuerdo	1	1.00-1.80
En desacuerdo	2	1.81-2.60
Ni de acuerdo, ni en desacuerdo	3	2.61-3.40
De acuerdo	4	3.41-4.20
Totalmente de acuerdo	5	4.21-5.00

Para el cálculo de la utilidad estimada con base en la escala Likert se define un cuestionario de evaluación basándose en el tema tratado, una vez obtenidos los resultados de la encuesta como se presenta en la **Tabla 5** se procede a la tabulación y análisis de los mismos tomando el valor asignado de la escala Likert como referencia para medir el nivel de satisfacción de los encuestados [31].

Tabla 5. Ejemplos resultados de la evaluación TAM

Utilidad estimada	Totalmente en desacuerdo (%)	En desacuerdo (%)	Ni de acuerdo, ni en desacuerdo (%)	De acuerdo (%)	Totalmente de acuerdo (%)
UE01	-	-	-	-	-
UE02	-	-	-	-	-
UE03	-	-	-	-	-
UE04	-	-	-	-	-

Utilidad Estimada (UE): Percepción de mejora en actividades.

4.7. Trabajos Relacionados

En la **Tabla 6** se presenta una lista de los trabajos relacionados con el TIC.

Tabla 6. Trabajos relacionados

Número	Título	Descripción
1	Sistema de inventario forestal de parques urbanos de la ciudad de Cuenca	La investigación abarca el inventario forestal de la parte urbana de la ciudad de Cuenca, e incluye el desarrollo de una aplicación móvil para el soporte del levantamiento de datos [1].
2	In-situ measurements from mobile platforms: An emerging approach to address the old challenges associated with forest inventories	La investigación se relaciona con las mediciones de recursos forestales in situ apoyadas con plataformas móviles como una técnica para resolver desafíos relacionados con los inventarios forestales [34].
3	Extensión al modelo de aceptación de tecnología TAM, para ser aplicado a sistemas colaborativos, en el contexto de pequeñas y medianas empresas.	Esta investigación propone un modelo de aceptación de tecnologías colaborativas, basado en TAM, para ser aplicado dentro de la empresa y evaluar la tecnología para estimar su utilidad [33].
4	Application Development using Flutter	La investigación abarca el desarrollado de una aplicación móvil basada en Flutter para un sistema de facturación [35].
5	Kusa-Yupaq: Mobile Application to Improve the Inventory of Heritage Assets	La investigación realiza el inventario de bienes, mediante el desarrollo de una aplicación móvil, que utiliza las fases de la metodología Mobile-D [36].
6	Aplicación móvil usando el framework Flutter para el control de las tutorías académicas del sistema integrado de la Universidad Técnica de Ambato	En este trabajo se desarrolla una aplicación móvil para la gestión de tutorías utilizando el framework Flutter, la metodología Mobile-D y el lenguaje Dart [37].

5. Metodología

En esta sección, se discute la metodología empleada para llevar a cabo el desarrollo y la ejecución del presente TIC. Para lo cual, en la sección **5.1** se describe el área de estudio en donde se realizó el TIC. La sección **5.2** aborda el procedimiento para ejecutar los objetivos establecidos. De igual manera, la sección **5.3** muestra los recursos necesarios para el desarrollo del TIC. Finalmente, la sección **5.4** detalla los participantes en el TIC.

5.1. Área de Estudio

El presente TIC fue realizado en el área urbana de la ciudad de Loja-Ecuador como apoyo al proyecto de investigación "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja" para la carrera de Ingeniería Forestal de la Universidad Nacional de Loja en un entorno académico.

5.2. Procedimiento

En el siguiente apartado se presenta el procedimiento establecido para ejecutar los objetivos planteados en el TIC.

5.2.1. Fase 1. Desarrollar el front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja" en base a los requerimientos actuales del proyecto.

Para la ejecución de la fase 1 se tomó como base el ciclo de desarrollo de la metodología Mobile-D.

- Exploración:
 - Se realizó el establecimiento de los grupos de interés, el levantamiento de requerimientos y la limitación del alcance del TIC mediante entrevistas al Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc., como docente investigador encargado de la parte tecnológica del proyecto "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja". Además, se realizaron reuniones periódicas con otros miembros del equipo, incluyendo al Dr. Darwin Pucha, director del proyecto y docente investigador de la carrera de Ingeniería Forestal.
 - Se especificaron los requerimientos por medio del estándar IEEE-830. En este documento constan los requerimientos del sistema y las historias de usuario.

- Inicialización:
 - Diseño de la aplicación a través de los siguientes diagramas:
 - Diagrama de casos de uso.
 - Diagrama de componentes.
 - Diagrama de despliegue.
 - Diseño de interfaces.
- Producción: Se realizó la planificación de fases, la estructura de los directorios del sistema, la implementación de la aplicación siguiendo la arquitectura definida en etapas anteriores; y, por último, se evaluó el sistema.
 - Planificación de fases a través de historias de usuario.
 - Pruebas del sistema: Se desarrollaron las siguientes pruebas:
 - Pruebas unitarias.
 - Pruebas de aceptación.
- Estabilización: El sistema finalizado se presenta y se documenta oficialmente mediante un acta de entrega.

5.2.2. Fase 2. Evaluar la aplicación móvil mediante la utilidad estimada del modelo de aceptación de tecnología (TAM) para determinar el grado de soporte que brinda en el inventario in-situ.

Se aplicó el modelo de aceptación TAM para evaluar la utilidad estimada del sistema. La cual consta de los siguientes pasos:

- Elaboración de cuestionario de evaluación en base a la utilidad estimada del modelo TAM.
- Tabulación de resultados considerando la escala Likert.
- Análisis de resultados.

5.3. Recursos

Con la finalidad de ejecutar las fases previamente mencionadas y cumplir los objetivos del presente TIC, se emplearon los siguientes recursos.

5.3.1. Recursos Científicos

5.3.1.1. Experimento

Para aplicar esta metodología, se llevó a cabo un proceso de experimentación que incluye una descripción detallada de las tareas a ejecutar. Este proceso se empleó específicamente en el segundo objetivo, que abarcaba la evaluación de la aplicación móvil mediante la utilidad estimada del modelo de aceptación de tecnología (TAM).

5.3.1.2. Investigación bibliográfica

Para respaldar y enriquecer el estudio, se llevó a cabo una investigación bibliográfica. Se recurrió a diversas fuentes científicas y académicas, como libros especializados, revistas científicas y artículos de investigación relevantes en el campo de estudio. Esta investigación bibliográfica permitió obtener información actualizada, fundamentos teóricos y enfoques metodológicos aplicables al contexto del presente TIC.

5.3.2. Recursos de Hardware y Software

5.3.2.1. Hardware

- **Laptop ASUS:** Equipo utilizado para el desarrollo del TIC.

5.3.2.2. Software

- **Visual Studio Code:** IDE empleado para el desarrollo del sistema.
- **Android Studio:** IDE para la plataforma de Android utilizado para la emulación de dispositivos móviles.
- **GitHub:** Portal web para llevar el control de versiones del sistema.
- **Enterprise Architecture:** Software para el diseño de diagramas.
- **Figma:** Programa de edición gráfica utilizado para el prototipado de interfaces gráficas de la aplicación.
- **Google Play Store:** Plataforma de distribución digital de aplicaciones móviles, usada para desplegar el sistema.

5.3.3. Recursos Técnicos

5.3.3.1. Entrevista

Como parte de los recursos técnicos utilizados, se llevó a cabo una entrevista al Ing. Oscar Miguel Cumbicus Pineda Mg.Sc., quien es parte del proyecto "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja" como Docente Investigador encargado de la parte tecnológica del proyecto. La entrevista brindó valiosos datos y conocimientos adicionales para respaldar el desarrollo del presente TIC.

5.4. Participantes

Los siguientes participantes estuvieron involucrados en el desarrollo del TIC:

- Carmen Azucena Toledo Iñiguez como autora del presente TIC.
- Ing. Wilman Patricio Chamba Zaragocín, Mg. Sc., como director del TIC e Investigador del proyecto "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja".
- Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc., como Docente Investigador encargado de la parte tecnológica del proyecto "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja", como facilitador de información para la realización del TIC.
- PhD. Darwin Alexander Pucha Cofrep como director del proyecto de investigación "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja", como facilitador de información para la realización del TIC.

6. Resultados

En este apartado se presenta detalladamente los resultados obtenidos por cada objetivo específico, así como las tareas que conllevan al cumplimiento total de dichos objetivos; en la sección **6.1** se realizó el primer objetivo guiado por las fases de la metodología Mobile-D; la sección **6.2** aborda el segundo objetivo, que se desarrolló siguiendo el modelo de aceptación TAM.

6.1. Objetivo 1: Desarrollar el front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” en base a los requerimientos actuales del proyecto.

Para cumplir con el primer objetivo, se aplicó la metodología Mobile-D, que consta de la fase de exploración, inicialización, producción y estabilización; que se detallan desde la sección **6.1.1** hasta la sección **6.1.4**.

6.1.1. Fase 1: Exploración

En esta fase se establecieron los requerimientos del sistema.

6.1.1.1. Establecimiento de los grupos de interés

En el **Anexo 1** se evidencia la entrevista realizada al Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc., como Docente Investigador encargado de la parte tecnológica del proyecto "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja" para determinar los grupos de interés y requerimientos del sistema. Además, se llevaron a cabo reuniones periódicas con el PhD. Darwin Alexander Pucha Cofrep, director del proyecto, y con otros miembros del equipo que forman parte de la carrera de Ingeniería Forestal.

6.1.1.2. Requerimientos del sistema

En la **Tabla 7** se muestran los requerimientos funcionales del sistema. Los cuales pueden consultarse en el **Anexo 2**.

Tabla 7. Requerimientos funcionales del sistema

Número	Nombre del requerimiento	Descripción del requerimiento
RF001	Registrar Cuenta	<p>El investigador o administrador debe registrar su cuenta para poder acceder al sistema.</p> <ul style="list-style-type: none">• Los datos requeridos de carácter obligatorio son: nombre, apellido, correo electrónico y contraseña; como opcional el grupo de medición al que pertenece.• El investigador al crear su cuenta estará deshabilitada hasta que el administrador la verifique.
RF002	Autenticar Usuario	<p>El investigador o administrador debe autenticarse (iniciar sesión) para poder acceder al sistema a través de su nombre de usuario (correo electrónico) y contraseña.</p>
RF003	Recuperar Contraseña	<p>El sistema debe permitir recuperar la contraseña a los usuarios a través del nombre de usuario (correo electrónico).</p>
RF004	Actualizar Cuenta	<p>El investigador y administrador pueden modificar datos de su cuenta. Los datos que pueden modificar son: nombre, apellido, correo y contraseña.</p>
RF005	Administrar Usuarios	<p>El administrador puede gestionar las cuentas de usuario y verificar la cuenta de investigador.</p> <p>La gestión de cuentas de usuario corresponde a:</p> <ul style="list-style-type: none">• Actualizar datos: nombre, apellido, correo electrónico, grupo de medición al que pertenece la cuenta del investigador y estado (activado/desactivado).
RF006	Administrar Familias	<p>El administrador puede gestionar los registros de familias.</p> <p>La gestión de familias corresponde a:</p> <ul style="list-style-type: none">• Crear familias. Los datos requeridos son: nombre y estado (activado/desactivado).• Actualizar datos: nombre y estado (activado/desactivado).
RF007	Administrar Especies	<p>El administrador puede gestionar los registros de especies.</p> <p>La gestión de especies corresponde a:</p> <ul style="list-style-type: none">• Crear especies. Los datos requeridos son: nombre, familia a la que pertenece y estado (activado/desactivado).• Actualizar datos: nombre de la especie, familia a la que pertenece y estado (activado/desactivado).
RF008	Administrar Árboles	<p>El investigador y administrador pueden gestionar los árboles.</p> <p>La gestión de árboles corresponde a:</p> <ul style="list-style-type: none">• Crear árboles, los datos requeridos son:<ul style="list-style-type: none">- Identidad del árbol: Familia, especie, nombre común, parroquia, sitio, tipo de sitio, acrónimo de ubicación, sección, estado actual, estado de registro, nombre del colector, comentario, fecha de registro.- Estado natural: Altitud, longitud y latitud (obtenidas automáticamente por el sistema), circunferencia de fuste, diámetro, altura total, altura comercial,

Número	Nombre del requerimiento	Descripción del requerimiento
		tamaño de copa media de norte a sur, tamaño de copa media de este a oeste, follaje o porcentaje de hojas, estado de madurez, floración, fructificación, rectitud de fuste, espacio de crecimiento. - Condición general: Estado fitosanitario, presencia de enfermedades, ubicación de las enfermedades, presencia de plagas, ubicación de las plagas, problemas físicos, riesgos potenciales, acciones recomendadas.
		<ul style="list-style-type: none"> • Actualizar datos: todos los datos asociados al árbol.
RF009	Eliminar árboles	El administrador puede eliminar árboles.
RF0010	Buscar Árboles	El invitado, investigador y administrador pueden buscar árboles por especie o por sitio.
RF0011	Visualizar árboles en el mapa	El invitado, investigador y administrador pueden visualizar árboles ubicados en el mapa.
		El invitado, investigador y administrador pueden visualizar los datos de los árboles. Los datos incluyen dos formatos:
RF0012	Visualizar datos de árboles	<ul style="list-style-type: none"> • Información resumida: Nombre científico, nombre común, longitud, latitud, código, circunferencia de fuste, diámetro, altura total, altura comercial, follaje o porcentaje de hojas, familia y sitio. • Información extendida: todos los datos asociados al árbol.
RF0013	Descargar código QR	El invitado, investigador y administrador pueden descargar código QR que contenga información asociada al árbol, en formato PNG, JPEG, WebP.
RF0014	Escanear código QR	El invitado, investigador y administrador pueden escanear código QR, y el sistema mostrará la información asociada.

Requerimiento Funcional (RF): Define una función del sistema.

En la **Tabla 8** se muestran los requerimientos no funcionales del sistema. Los cuales pueden consultarse en el **Anexo 2**.

Tabla 8. Requerimientos no funcionales del sistema

Número	Nombre del requerimiento	Descripción del requerimiento
RNF001	Usabilidad	El sistema móvil debe poseer interfaces amigables.
RNF002	Seguridad	La aplicación móvil debe conceder acceso a los usuarios según sus respectivas credenciales y permisos.
RNF003	Operacional	El sistema será implementado en un entorno móvil con el sistema operativo Android, con la versión mínima de Android 4.4 KitKat (API nivel 20) o superior.
RNF004	Desarrollo	El sistema será desarrollado con el SDK Flutter, el lenguaje de programación Dart y Visual Studio Code como IDE de desarrollo.
RNF005	Éticos	El sistema se enmarca en un ámbito de respeto y no discriminación a ningún tipo de persona, sociedad u organización.

Requerimiento No Funcional (RNF): Define las restricciones o requisitos del sistema.

A continuación, en la **Tabla 9**, se indica un ejemplo de historia de usuario para el registro de cuentas; el listado completo de historias de usuario puede verse en el **Anexo 2**.

Tabla 9. Ejemplo de historia de usuario

Historia de Usuario	
Número: 1	Nombre: Registrar cuenta
Usuario: Investigador, Administrador	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos Estimados: 2
Riesgo en desarrollo: Alta	Referencia requerimiento: RF001
Descripción: COMO Investigador y Administrador QUIERO registrar mi cuenta PARA usar las funcionalidades de investigador y administrador, respectivamente.	
Criterios de aceptación: <ul style="list-style-type: none"> • Validar los campos requeridos como: nombre, apellido, correo electrónico y contraseña. • En el campo grupo de medición debe presentar un listado para seleccionar el grupo. • Cuando registre su cuenta estará deshabilitada hasta que sea aprobada por el administrador. 	
Observaciones	
Requerimiento Funcional (RF): Define una función del sistema.	

6.1.2. Fase 2: Inicialización

En esta sección se realizó el diseño del sistema.

6.1.2.1. Diseño de la aplicación

a. Casos de uso

En la **Figura 4**¹ se indican en forma gráfica, a través de un diagrama de casos de uso, los escenarios en que interviene la aplicación móvil con sus respectivos actores.

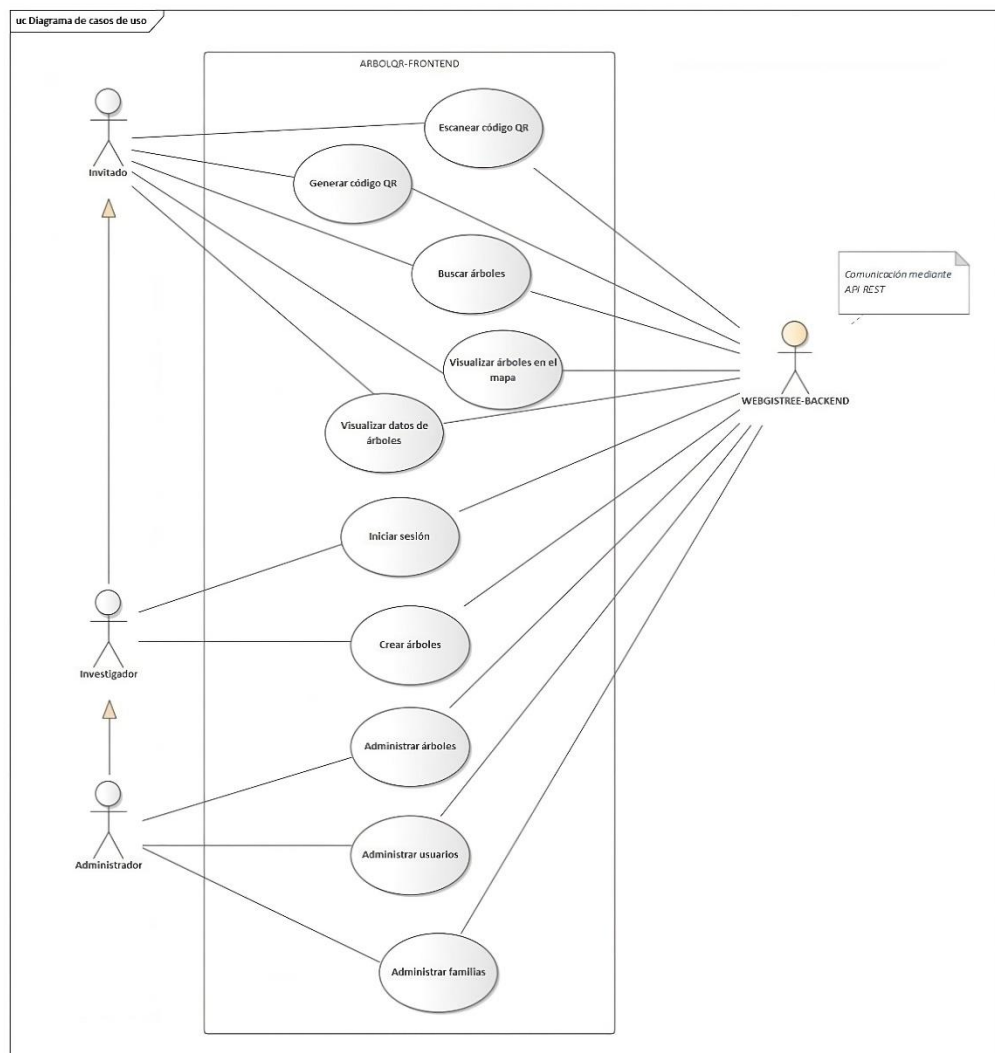


Figura 4. Diagrama de casos de uso del sistema

¹ <https://n9.cl/m5z36>

a. Diagrama de componentes

En la **Figura 5**² se muestra el diagrama de componentes del sistema, el cual representa la arquitectura n-capas empleada.

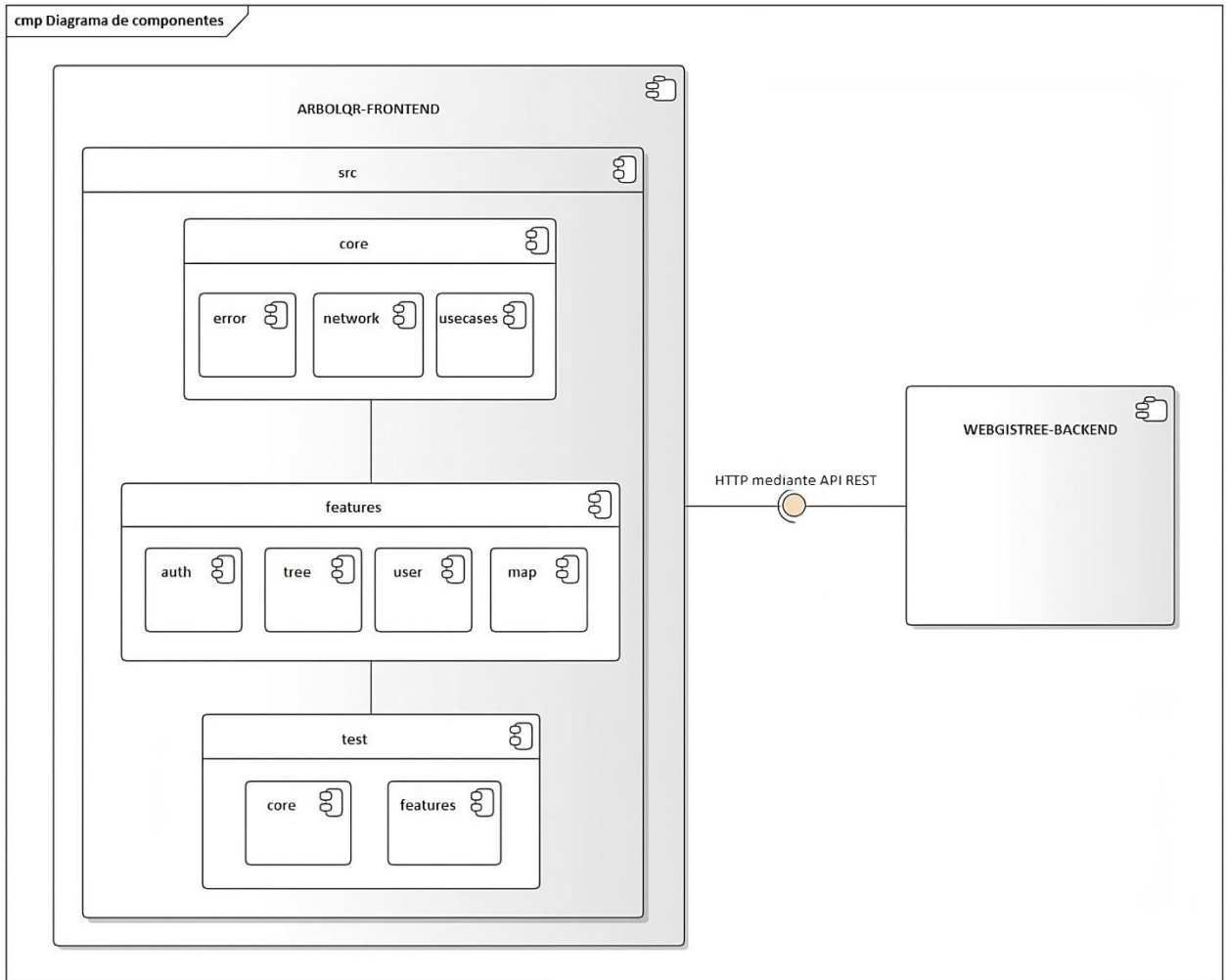


Figura 5. Diagrama de componentes

² <https://n9.cl/m5z36>

b. Diagrama de despliegue

En la **Figura 6³** se muestra el diseño físico de la aplicación representado por un diagrama de despliegue.

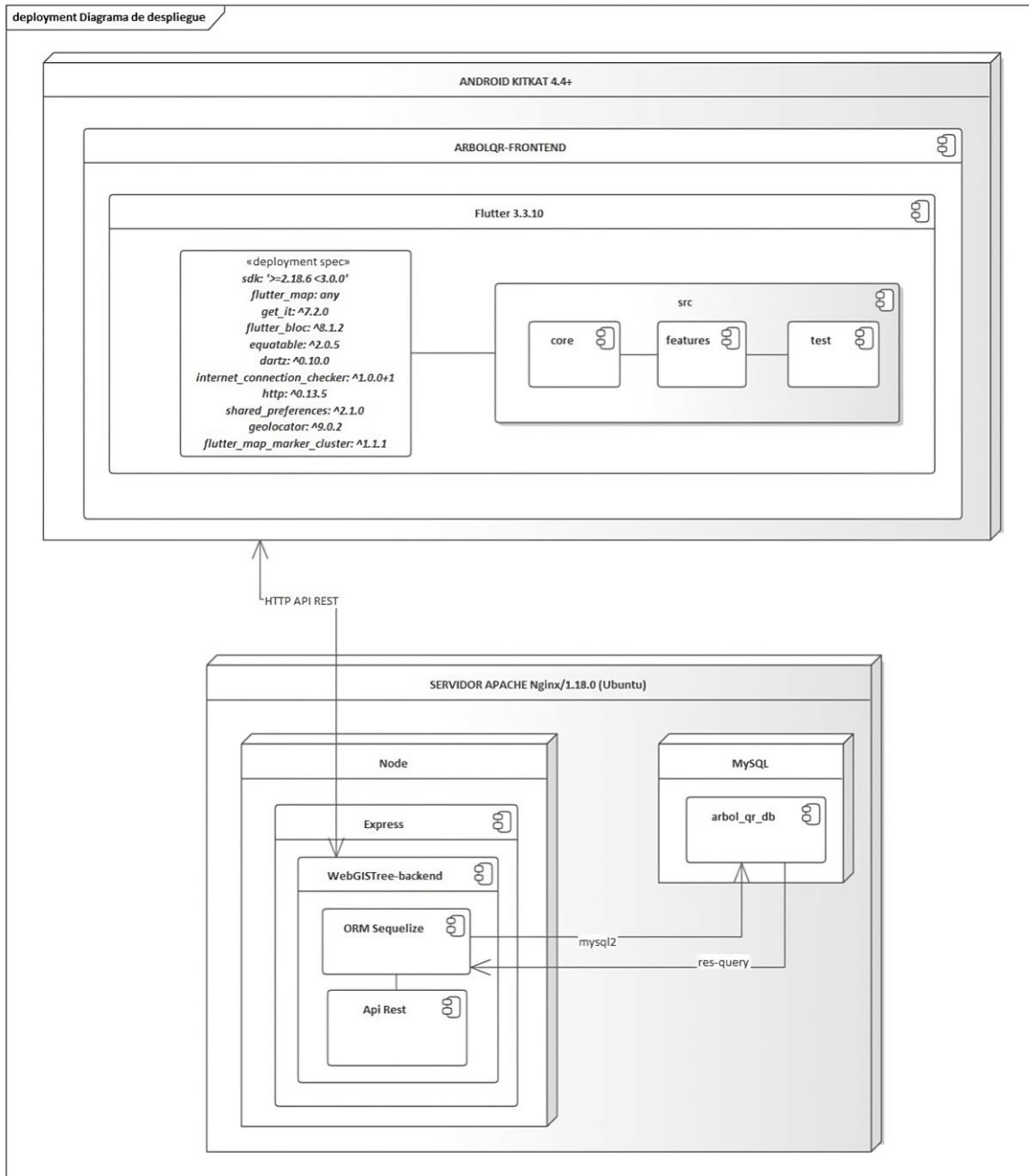


Figura 6. Diagrama de despliegue

³ <https://n9.cl/m5z36>

6.1.2.2. Diseño de interfaces

En la **Figura 7** a la **Figura 12** se presenta el diseño de interfaces⁴ que se evidencian en el **Anexo 2**.

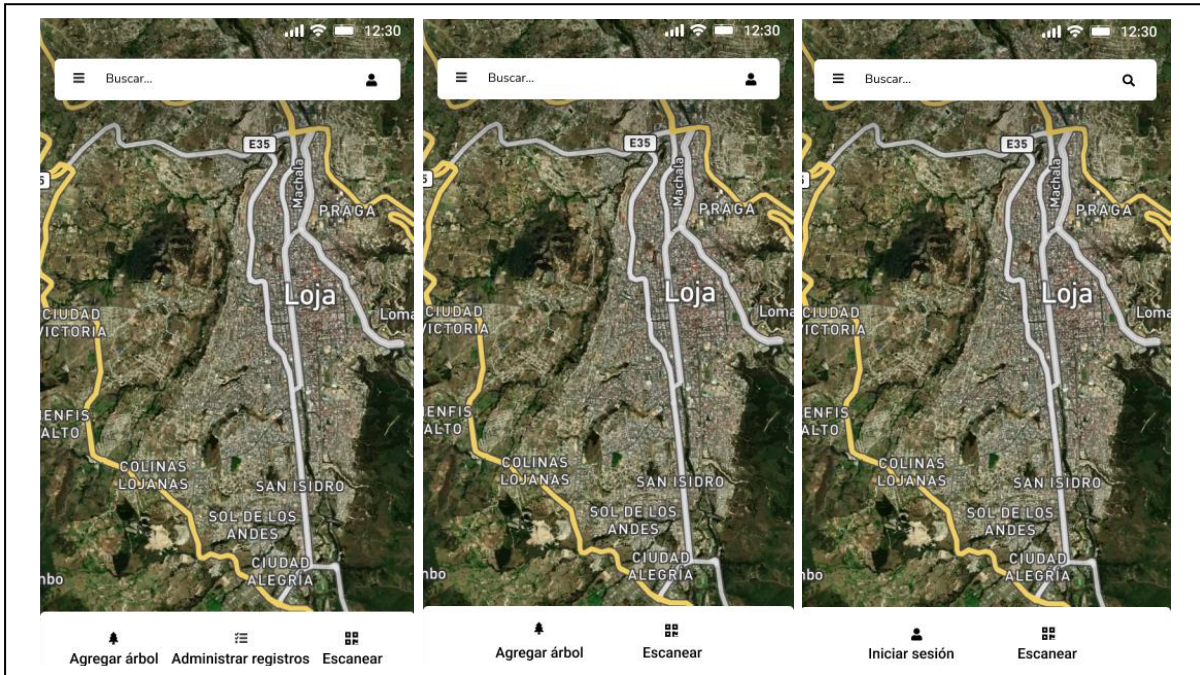


Figura 7. Interfaces generales

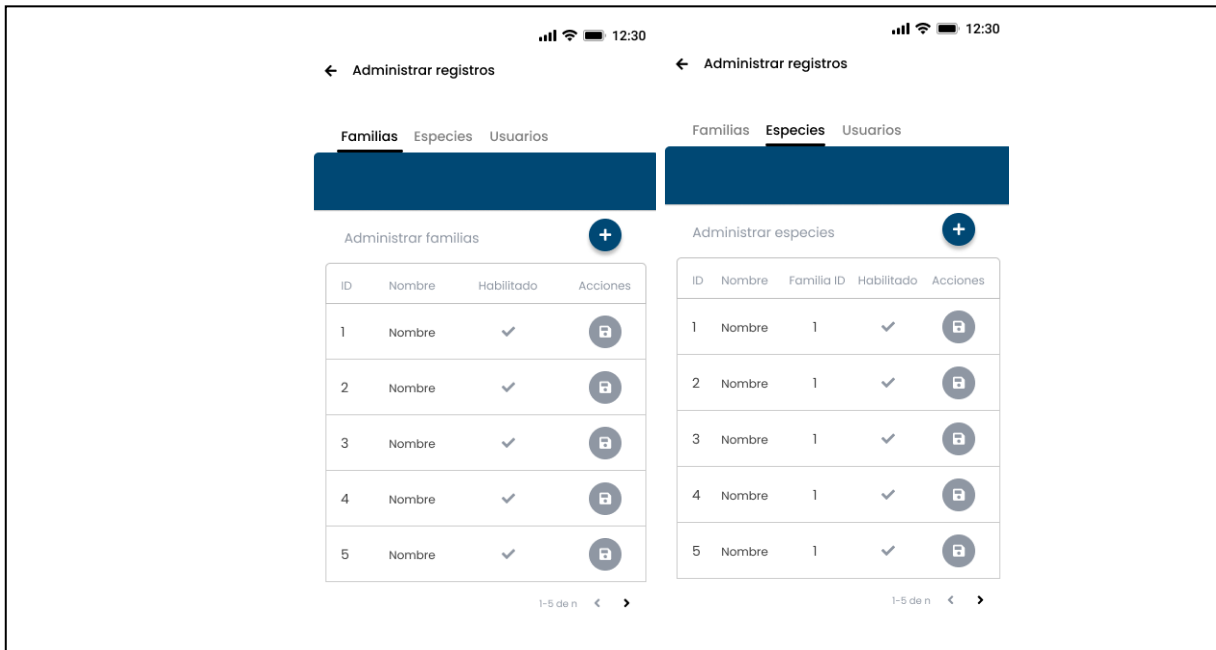


Figura 8. Interfaces de administración

⁴ <https://n9.cl/fctqna>

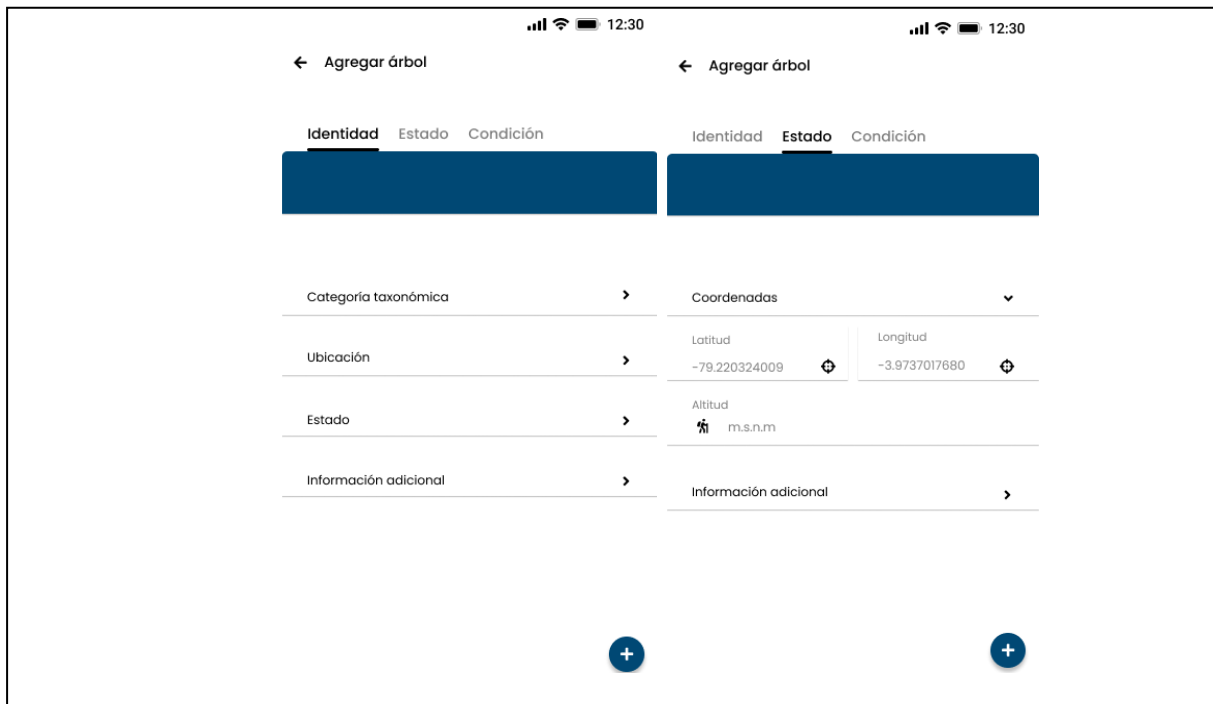


Figura 9. Interfaces para agregar árboles

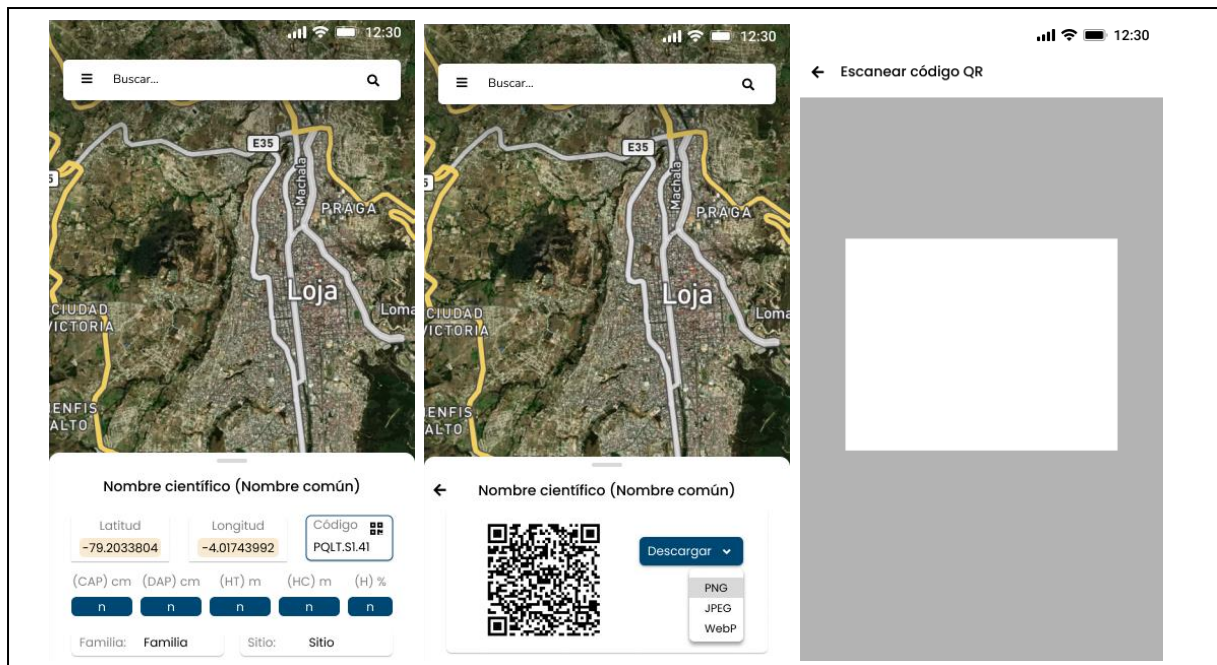


Figura 10. Interfaces para código QR

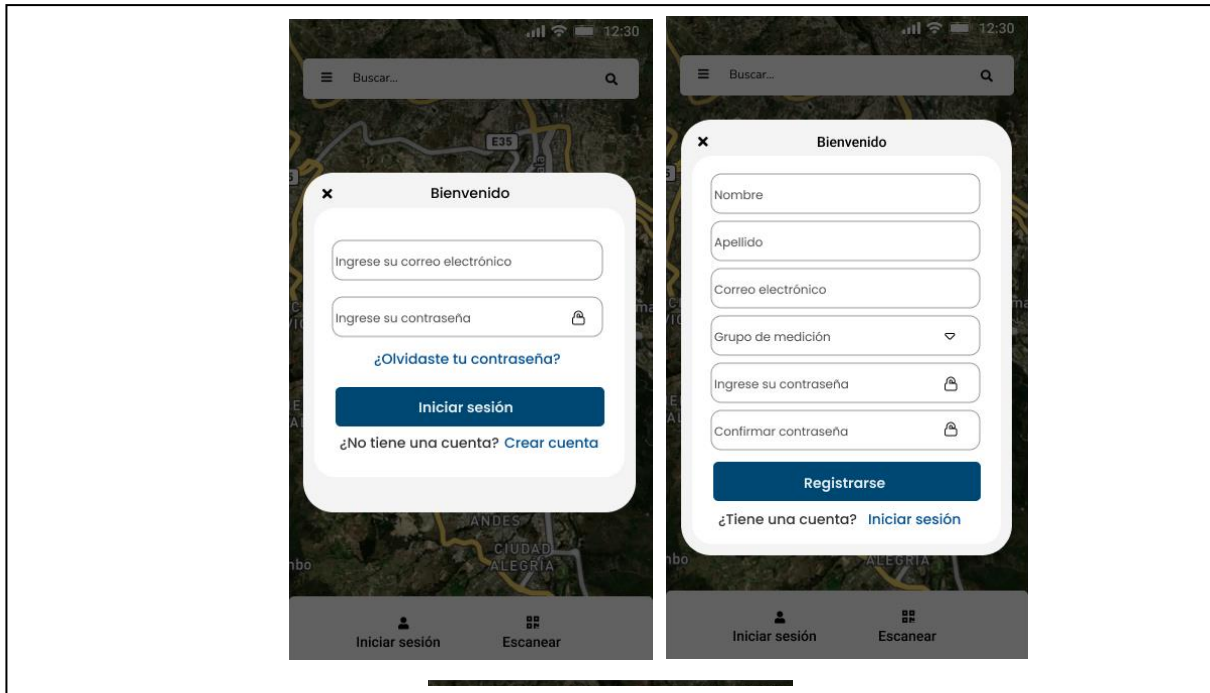


Figura 11. Interfaces para seguridad

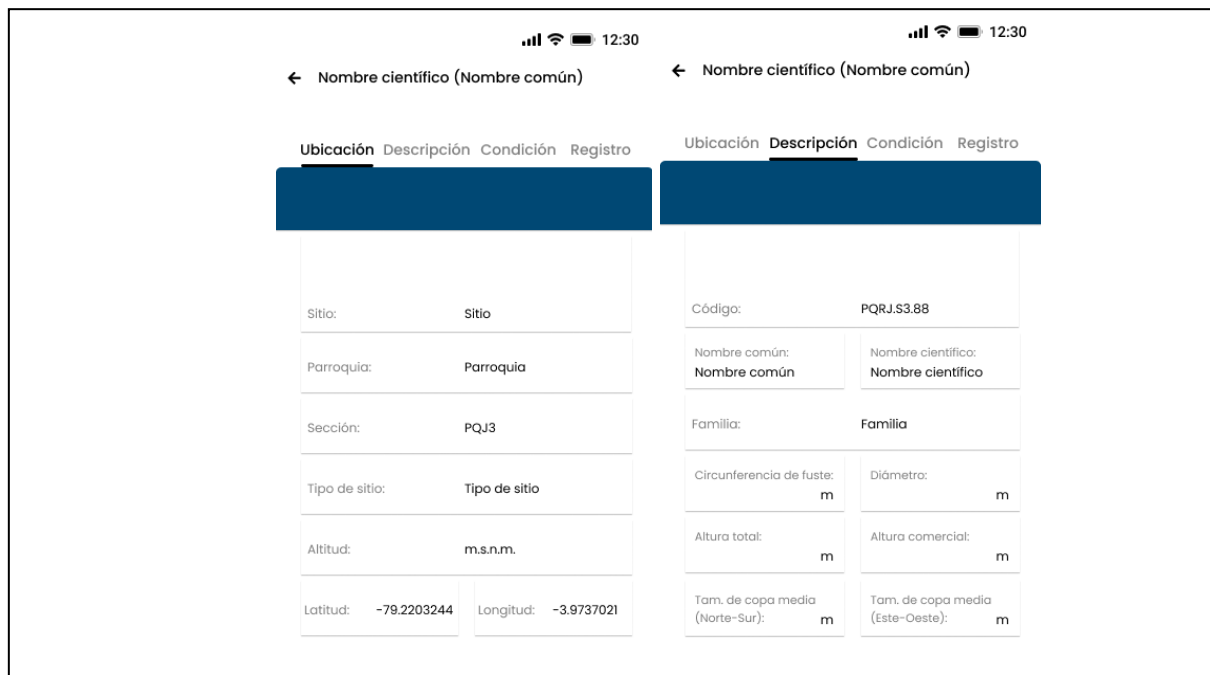


Figura 12. Interfaces para visualizar datos

6.1.3. Fase 3: Producción

6.1.3.1. Planificación de fases

La planificación de fases para el desarrollo del sistema tomó como base las historias de usuario, las cuales se dividen en iteraciones, y se listan en la **Tabla 10**. Puede revisarse el **Anexo 2**, que contiene el detalle de cada una de las historias de usuario.

Tabla 10. Planificación de fases a través de historias de usuario

Iteración	Referencia de requerimiento	Historia de Usuario
Iteración 1	RF001	Registrar cuenta
Iteración 1	RF002	Autenticar usuario
Iteración 1	RF003	Recuperar contraseña
Iteración 1	RF004	Actualizar Cuenta
Iteración 2	RF005	Administrar Usuarios
Iteración 3	RF006	Administrar Familias
Iteración 3	RF007	Administrar Especies
Iteración 4	RF008	Administrar Árboles
Iteración 4	RF009	Eliminar Árboles
Iteración 5	RF0010	Buscar Árboles
Iteración 6	RF0011	Visualizar árboles en el mapa
Iteración 6	RF0012	Visualizar datos de árboles
Iteración 7	RF0013	Descargar código QR
Iteración 7	RF0014	Escanear código QR

Requerimiento Funcional (RF): Define una función del sistema.

6.1.3.2. Estructura de directorios

En la Figura 13⁵ se presenta la estructura del directorio del sistema.

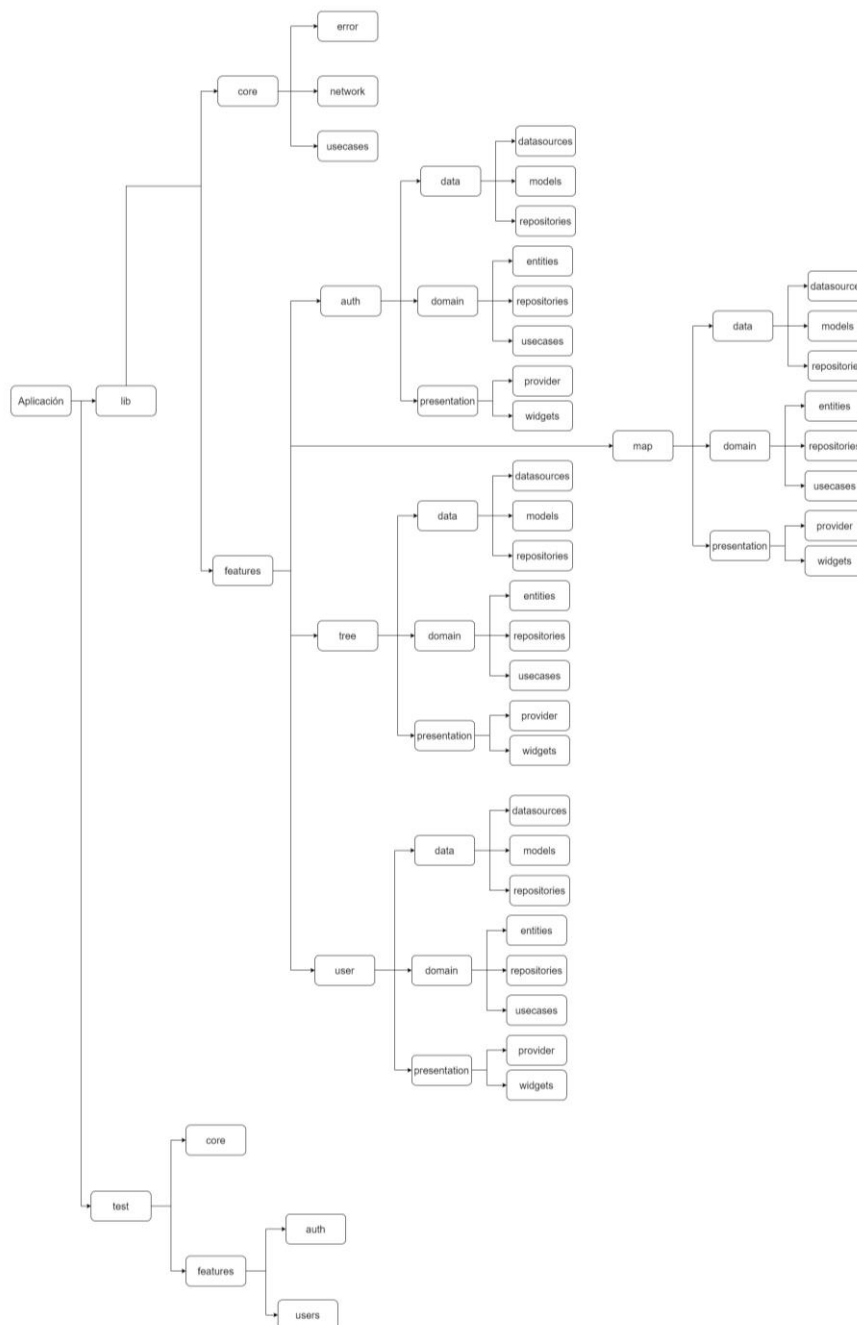


Figura 13. Estructura de directorios

⁵ <https://n9.cl/m5z36>

6.1.3.3. Implementación

A continuación, se muestra un ejemplo de la implementación del sistema, siguiendo la arquitectura establecida en la sección 6.1.2. En primer lugar, se estructuró cada módulo como se muestra en la **Figura 14**, partiendo de la arquitectura por capas del diagrama de componentes de la **Figura 5**.

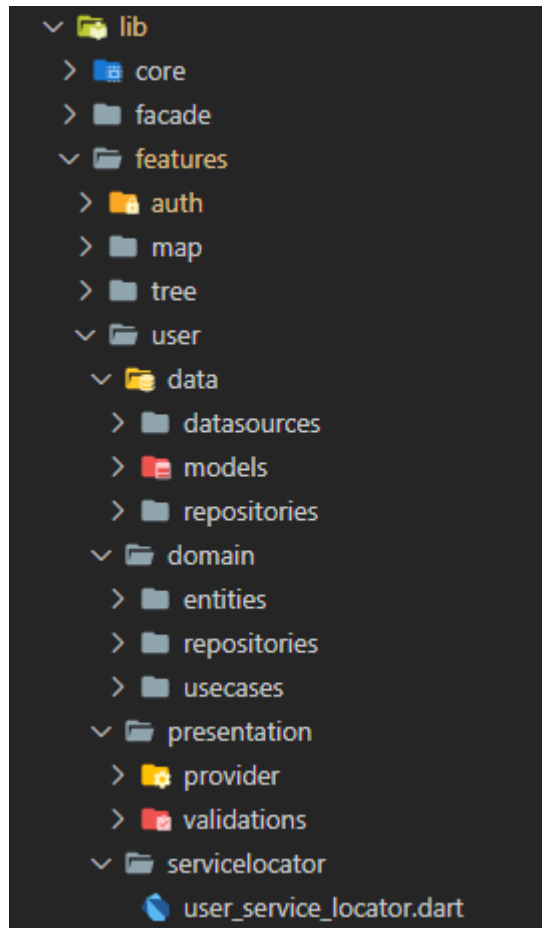


Figura 14. Estructuración del módulo User

Siguiendo la arquitectura, se especificó la entidad *user* como se muestra en la **Figura 15**, en donde se identificaron sus atributos.

```
1 import 'package:equatable/equatable.dart';
   You, last week | 1 author (You)
2 class User extends Equatable {
3     final String id;
4     String nombre;
5     String apellido;
6     String email;
7     int? idGrupoMedicion;
8     dynamic habilitado;
9     String idRol;
10    User({
11        required this.id,
12        this.nombre = '',
13        this.apellido = '',
14        this.email = '',
15        this.idGrupoMedicion,
16        this.habilitado,
17        this.idRol = '',
18    });
19    @override
20    List<Object?> get props => [
21        id,
22        nombre,
23        apellido,
24        email,
25        idGrupoMedicion,
26        habilitado,
27        idRol,
28    ];
29 }
```

Figura 15. Entidad user

Posteriormente, se estructuró el repositorio de dominio como se muestra en la **Figura 16** con lo que se pudo efectuar el primer test.

```
1 import 'package:dartz/dartz.dart';
2 import 'package:proyecto/features/user/domain/entities/grupo_medicion.dart';
3 import 'package:proyecto/features/user/domain/entities/rol.dart';
4 import 'package:proyecto/features/user/domain/entities/user.dart';
5
6 import '../../../../../core/error/failure.dart';
7
8 You, 10 seconds ago | 1 author (You)
9 abstract class UserRepository {
10     Future<Either<Failure, bool>>> register_user(String nombre, String apellido, String email, String contrasena, String idGrupoMedicion);
11     Future<Either<Failure, List<GrupoMedicion>>>> get_grupos_medicion();
12     Future<Either<Failure, Rol>>> get_rol();
13     Future<Either<Failure, List<User>>>> get_users();
14     Future<Either<Failure, List<User>>>> get_local_users();
15     Future<Either<Failure, bool>>> update_user(String userId, String nombre, String apellido, String email, int idGrupoMedicion, bool habil
```

Figura 16. Repositorio de user

Posteriormente, se codificó el caso de uso como se evidencia en la **Figura 17**, con lo que el test estructurado inicialmente puede completarse como se muestra en la **Figura 18**.

```
1 import 'package:dartz/dartz.dart';
2 import 'package:proyecto/features/user/domain/entities/user.dart';
3 import 'package:proyecto/features/user/domain/repositories/user_repository.dart';
4 import '../core/error/failure.dart';
5 import '../core/usecases/usecases.dart';
6
7 You, 4 weeks ago | 1 author (You)
8 class GetUsers implements UseCase<List<User>, NoParams>{
9   final UserRepository repository;
10
11   GetUsers(this.repository);
12
13   Future<Either<Failure, List<User>>>? call(NoParams params) async {
14     return repository.get_users();
15   }
16 }
```

Figura 17. Caso de uso de user

```
9 class MockUserRepository extends Mock implements UserRepository {}
10
11 Run | Debug
12 void main() {
13   late GetUsers usecase;
14   late MockUserRepository mockUserRepository;
15   setUp(() {
16     mockUserRepository = MockUserRepository();
17     usecase = GetUsers(mockUserRepository);
18   });
19   final tUser = [
20     User(
21       id: "id",
22       nombre: 'Nombre',
23       apellido: 'Apellido',
24       email: 'correo@example.com',
25       idGrupoMedicion: 1,
26       idRol: 'idRol',
27       habilitado: 1)
28 ];
29 Run | Debug
30 test('should get users from the repository', () async {
31   // arrange future<Failure<L, R>> get_users();
32   when(() => mockUserRepository.get_users())
33     .thenAnswer((_) async => Right(tUser));
34   // act
35   final result = await usecase(NoParams());
36   // assert
37   expect(result, Right(tUser));
38   verify(() => mockUserRepository.get_users());
39   verifyNoMoreInteractions(mockUserRepository);
40 });
41 }
```

Figura 18. Test del caso de uso user

De igual manera, continuando con la arquitectura establecida, para cada *data* de los módulos, se partió definiendo su modelo (ver **Figura 19**), luego la implementación de su repositorio (ver **Figura 20**); con lo que se pudo efectuar el primer test (ver **Figura 21**), para finalmente definir su *datasource* (ver **Figura 22**), y ejecutar el segundo test (ver **Figura 23**).

```

class UserModel extends User {
  UserModel({
    required String id,
    required String nombre,
    required String apellido,
    required String email,
    int? idGrupoMedicion,
    required dynamic habilitado,
    required String idRol,
  }) : super(
    id: id,
    nombre: nombre,
    apellido: apellido,
    email: email,
    idGrupoMedicion: idGrupoMedicion,
    habilitado: habilitado,
    idRol: idRol,
  );

  factory UserModel.fromJson(Map<String, dynamic> json) {
    return UserModel(
      id: json['id_u'],
      nombre: json['nombre'],
      apellido: json['apellido'],
      email: json['email'],
      idGrupoMedicion: json['idGrupoMedicion'],
      habilitado: json['habilitado']['data'][0],
      idRol: json['idRol'],
    );
  }

  static List<UserModel> userListFromJson(Map<String, dynamic> json) {
    var users = json['users'] as List<dynamic>;
    return users.map((userJson) => UserModel.fromJson(userJson)).toList();
  }
}

```

You, 1 second ago • Uncommitted changes

Figura 19. Modelo de user

```

class UserRepositoryImpl implements UserRepository {
  final UserRemoteDataSource remoteDataSource;
  final UserLocalDataSource localDataSource;
  final NetworkInfo networkInfo;

  UserRepositoryImpl({
    required this.remoteDataSource,
    required this.localDataSource,
    required this.networkInfo,
  });

  @override
  Future<Either<Failure, List<User>>>> get_users() async {
    networkInfo.isConnected;
    try {
      final remote = await remoteDataSource.get_users();
      await localDataSource.cache_user(remote);
      return Right(remote);
    } on ServerException {
      return Left(ServerFailure(message: ''));
    }
  }
}

```

Figura 20. Implementación de repositorio de user

```

25 group('fromJson', () {
26     test('should return a valid model', () {
27         // arrange
28         final Map<String, dynamic> jsonMap = json.decode(fixture('user.json'));
29         // act
30         final result = UserModel.fromJson(jsonMap);
31         // assert
32         expect(result, tUserModel);
33     });
34 }
35 test('should return a JSON map containing the proper data', () {
36     // act
37     final result = tUserModel.toJson();
38     // assert
39     final expectedMap = {
40         "id_u": "id_u",
41         "nombre": "Nombre",
42         "apellido": "Apellido",
43         "email": "correo@example.com",
44         "habilitado": {'data': [1]},
45         "idGrupoMedicacion": 1,
46         "idRol": "idRol"
47     };
48     expect(result, expectedMap);
49 });
50 });
51 }

```

Figura 21. Test del modelo de user

```

abstract class UserRemoteDataSource {
    Future<bool> register user(String nombre, String apellido, String email,
        String contrasena, String idGrupoMedicacion);
    Future<List<GrupoMedicacionModel>> get_grupos_medicacion();
    Future<RolModel> get_rol();
    Future<List<UserModel>> get_users();
    Future<bool> update user(String userId, String nombre, String apellido,
        String email, int idGrupoMedicacion, bool habilitado);
}
You, 1 second ago | 1 author (You)
class UserRemoteDataSourceImpl implements UserRemoteDataSource {
    final http.Client client;
    UserRemoteDataSourceImpl({required this.client});
    @override
    Future<List<UserModel>> get_users() async {
        String authToken = await getTokenFromSharedPreferences();
        const url = '${ApiUrls.BASE_URL}/app-user';
        final response = await client.get(
            Uri.parse(url),
            headers: {
                'Authorization': 'Bearer ${authToken}',
                'Content-Type': 'application/json',
                'Accept': 'application/json',
            },
        );
        if (response.statusCode == 200) {
            final jsonBody = json.decode(response.body);
            List<UserModel> listado =
                UserModel.userListFromJson(jsonBody);
            return listado;
        } else {
            print(response.statusCode);
            throw ServerException(message: 'Error al obtener usuarios');
        }
    }
}

```

Figura 22. Datasource de user

```

27 group('getUsers', () {
28
29   final tUserModel = UserModel.userListFromJson(json.decode(fixture('users.json')));
30   test('should return UserModel whe the response code is 200', () async {
31     // arrange
32     when(() => mockHttpClient.get(any(), headers: any(named: 'headers')))
33     | .thenAnswer(_) async => http.Response(fixture('users.json'), 200);
34     // act
35     final result = await dataSource.get_users();
36     // assert
37     expect(result, equals(tUserModel));
38   });
39
40   // Test para cuando el response code es 400 o un error
41   test('should throw a ServerException when the response is 400', () {
42     // arrange
43     when(() => mockHttpClient.get(any(), headers: any(named: 'headers')))
44     | .thenAnswer(_) async => http.Response('Error', 400);
45     // act
46     final call = dataSource.get_users();
47     // assert
48     expect(() => call(), throwsA(TypeMatcher<ServerException>()));
49   });
50 });
51 }

```

Figura 23. Test de datasource de user

6.1.3.4. Pruebas del sistema

- Pruebas unitarias

En la Figura 24 se muestra el conjunto de 11 pruebas unitarias ejecutadas con éxito.

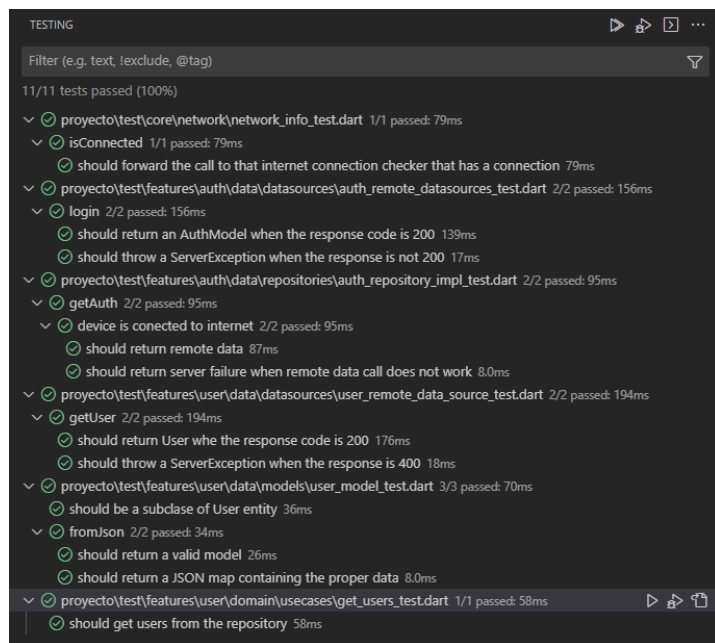


Figura 24. Pruebas unitarias

En la **Tabla 11** se presenta el resumen de las pruebas unitarias realizadas en la aplicación móvil, que hace referencia al **Recurso digital**.

Anexo 3 en donde se evidencia la ejecución de cada una de estas pruebas.

Tabla 11. Validación pruebas unitarias

Código	Feature	Componente	Descripción de la prueba	Resultado
PU-001	User	Caso de uso: get_users	Obtener usuarios desde el repositorio	Aprobado
PU-002	User	Modelos: user_model	Verificar que el modelo User es una subclase de la entidad Usuario	Aprobado
PU-003	User	Modelos: user_model	Devolver un modelo válido	Aprobado
PU-004	User	Modelos: user_model	Devolver un mapa JSON que contenga los datos adecuados	Aprobado
PU-005	Auth	Repositorio: auth_repository_impl	Cuando el dispositivo no está conectado a Internet: - Devolver datos remotos	Aprobado
PU-006	Auth	Repositorio: auth_repository_impl	Cuando el dispositivo no está conectado a Internet: - Devolver fallo de servidor cuando la llamada a datos remotos no funciona	Aprobado
PU-007	Auth	Data source: auth_remote_data_source	Logi: Devolver un AuthModel cuando el código de respuesta es 200.	Aprobado
PU-008	Auth	Data source: auth_remote_data_source	Login: Lanzar una ServerException cuando la respuesta no sea 200.	Aprobado
PU-009	User	Data source: user_remote_data_source	getUsers: Devolver listado de UserModel cuando el código de respuesta es 200.	Aprobado
PU-010	User	Data source: user_remote_data_source	getUsers: Lanzar una ServerException cuando la respuesta no sea 200.	Aprobado
PU-011	Core	Network info	Reenviar la llamada al verificador de conexión a Internet que tiene una conexión	Aprobado

Prueba Unitaria (PU): Técnica para validar un fragmento de código.

- **Pruebas de aceptación**

En la **Tabla 12** se presentan las pruebas de aceptación realizadas por el Docente Investigador encargado de la parte tecnológica del proyecto, el Ing. Oscar Miguel Cumbicus Pineda Mg.Sc., junto con otros miembros técnicos y directivos, incluido el PhD. Darwin Pucha Cofrep. En el **Recurso digital**.

Anexo 4 se evidencia la ejecución de cada una de estas pruebas.

Tabla 12. Pruebas de aceptación en base a las historias de usuario

Referencia de requerimiento	Historia de Usuario	Estado
RF001	Registrar cuenta	Aceptado
RF002	Autenticar usuario	Aceptado
RF003	Recuperar contraseña	Aceptado
RF004	Actualizar Cuenta	Aceptado
RF005	Administrar Usuarios	Aceptado
RF006	Administrar Familias	Aceptado
RF007	Administrar Especies	Aceptado
RF008	Administrar Árboles	Aceptado
RF009	Eliminar Árboles	Aceptado
RF0010	Buscar Árboles	Aceptado
RF0011	Visualizar árboles en el mapa	Aceptado
RF0012	Visualizar datos de árboles	Aceptado
RF0013	Descargar código QR	Aceptado
RF0014	Escanear código QR	Aceptado

Requerimiento Funcional (RF): Define una función del sistema.

6.1.4. Fase 4: Estabilización

La presentación del sistema terminado se evidencia con el acta de entrega que puede visualizarse en el **Recurso digital**.

Anexo 8. Además, el **Recurso digital**.

Anexo 6 y **Recurso digital**.

Anexo 7 contienen los manuales del sistema, para usuario y desarrollador, respectivamente.

En la **Tabla 13** se muestran las características del sistema.

Tabla 13. Características del sistema

Sistema Operativo	Android
Dispositivos móviles	Celulares, tablets.

Versión	Android 4.4 KitKat (API nivel 20) o superior.
Enlace de Google Play	https://play.google.com/store/apps/details?id=com.arbolqr.proyecto

En la **Figura 25** se presenta la aplicación móvil denominada “ÁrbolQR” disponible para su descarga en Google Play.

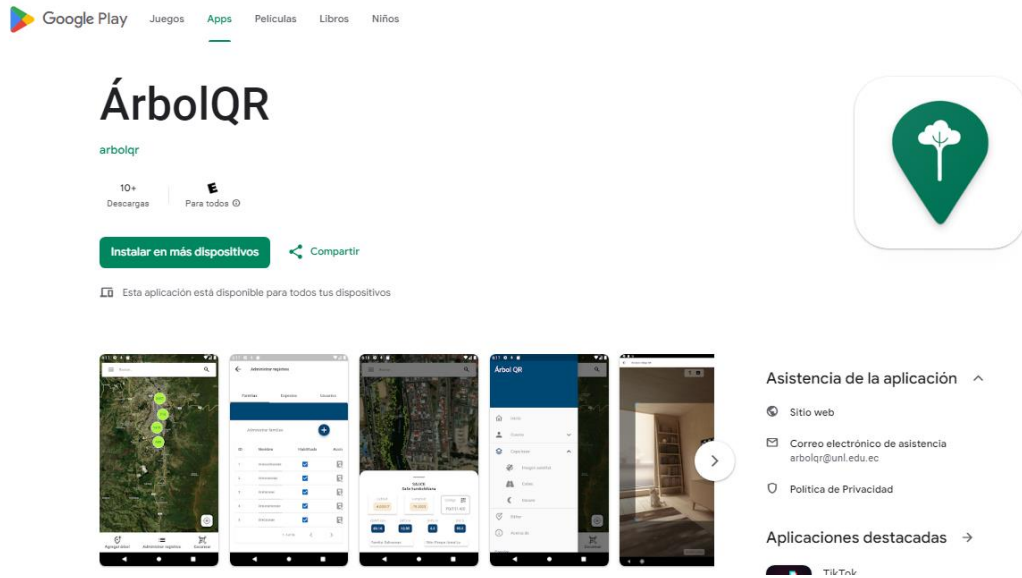


Figura 25. Aplicación móvil disponible en Google Play

Desde la **Figura 26** hasta la **Figura 28** se ilustran las principales interfaces de usuario de la aplicación móvil.

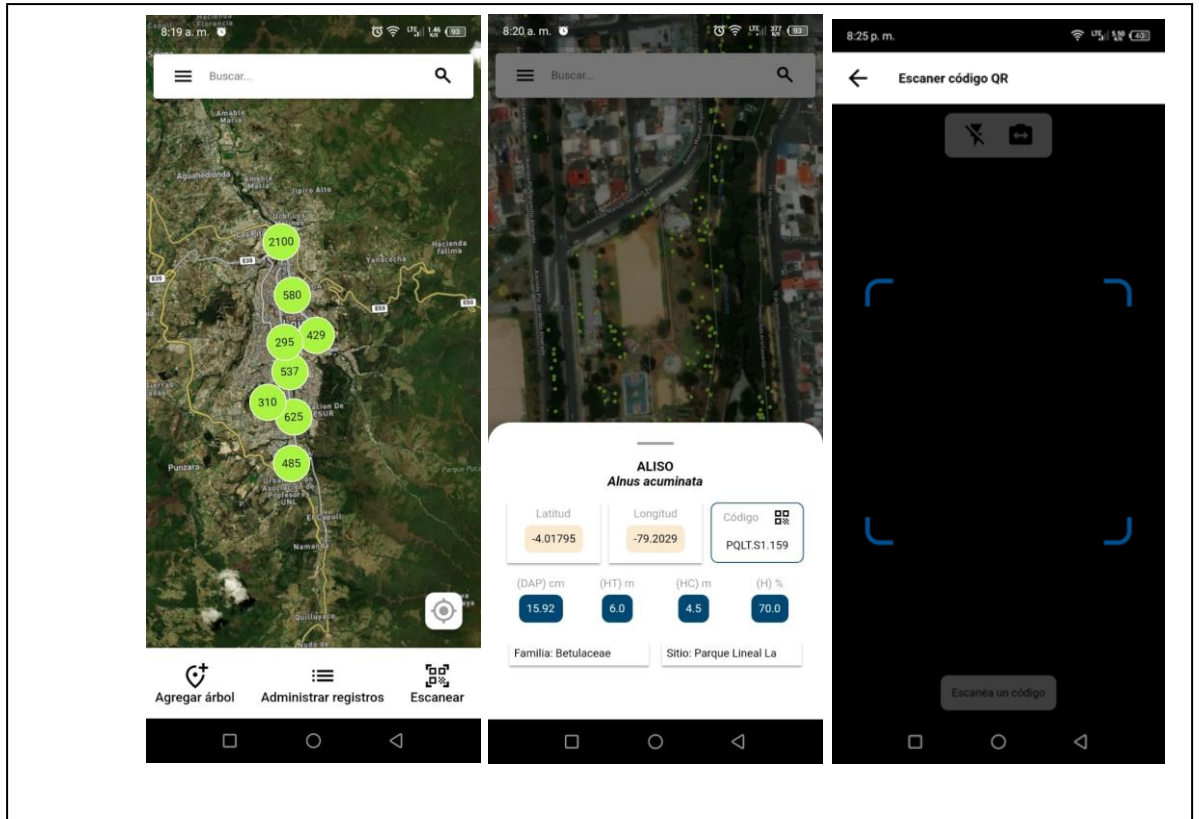


Figura 26. Interfaces para visualizar información.

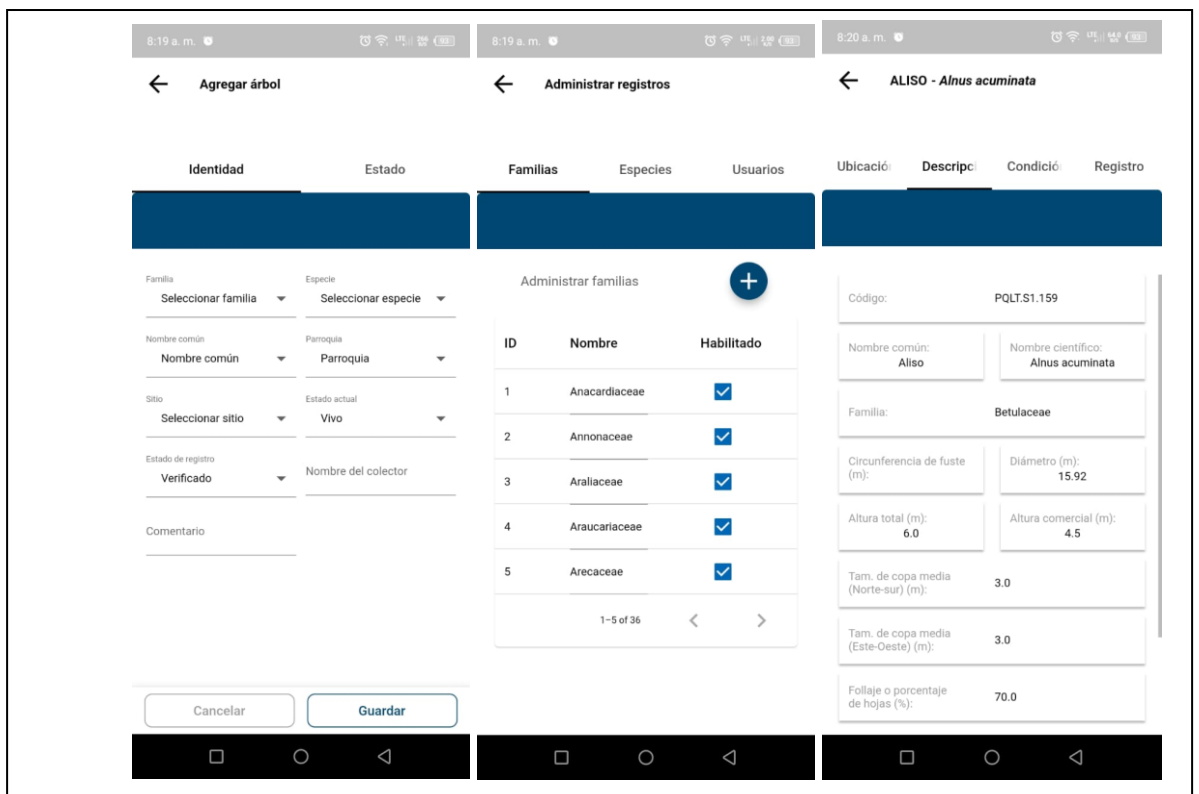


Figura 27. Interfaces para administrar y visualizar registros.

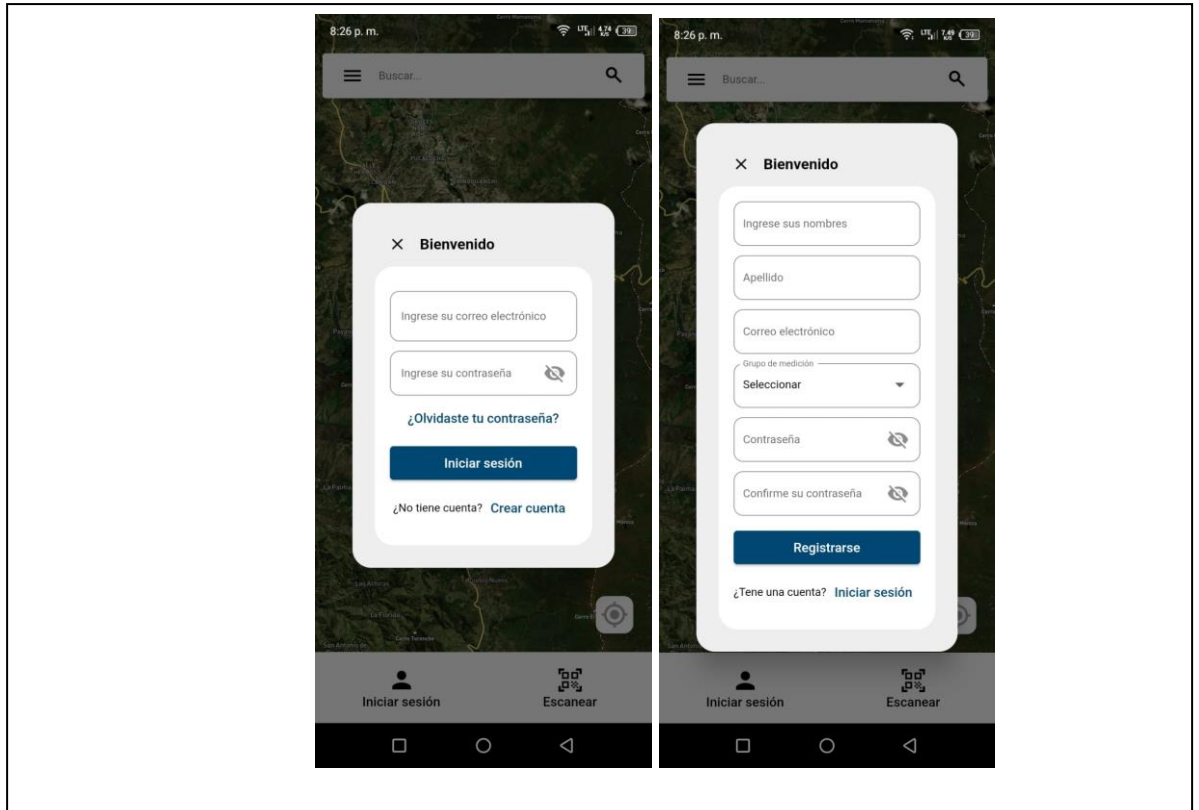


Figura 28. Interfaces de inicio de sesión y registro.

En la siguiente **Figura 29** se resumen los entregables por cada fase de la metodología Mobile-D.

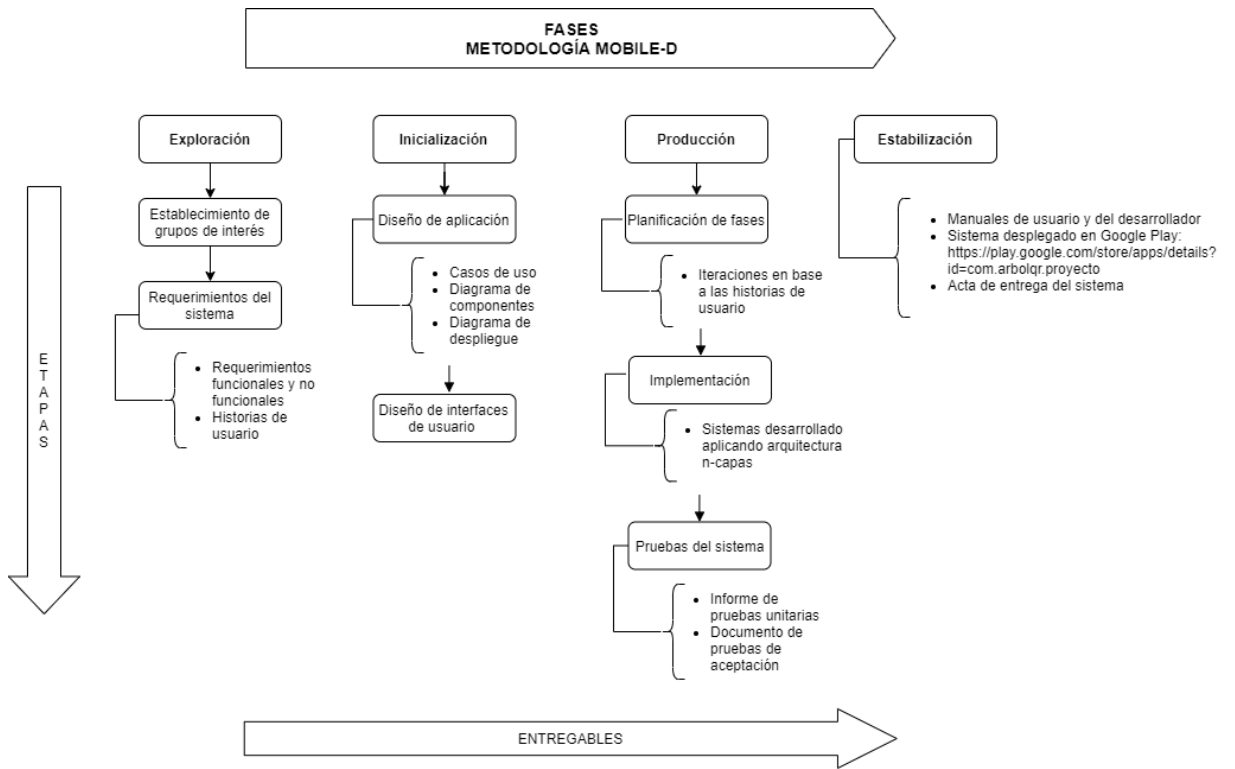


Figura 29. Diagrama de entregables de la metodología Mobile-D

6.2. Objetivo 2: Evaluar la aplicación móvil mediante la utilidad estimada del modelo de aceptación de tecnología (TAM) para determinar el grado de soporte que brinda en el inventario in-situ.

6.2.1. Fase 1: Elaboración de cuestionario de evaluación en base a la utilidad estimada del modelo TAM.

En **Tabla 14** se muestra el cuestionario realizado para evaluar la utilidad estimada de acuerdo al modelo TAM del sistema desarrollado.

Tabla 14. Cuestionario para la utilidad estimada del modelo TAM

Cuestionario para la utilidad estimada	
Identificador	Pregunta
UE01	¿En qué grado considera usted que esta aplicación móvil facilita la recopilación de datos en comparación con métodos tradicionales (papel y lápiz, hojas de cálculo, etc.)?
UE02	¿En qué grado considera usted que esta aplicación móvil mejora la precisión y la calidad de los datos que recopila en comparación con métodos manuales?
UE03	¿En qué grado considera usted que esta aplicación móvil agiliza el proceso de captura de datos y ahorra tiempo en comparación con métodos convencionales?
UE04	¿En qué grado considera usted que esta aplicación móvil brinda una mayor flexibilidad y accesibilidad para capturar datos en cualquier momento y lugar?
UE05	¿En qué grado considera usted que esta aplicación móvil ayuda a minimizar errores y omisiones en la recopilación de datos?

Utilidad Estimada (UE): Percepción de mejora en actividades.

En base a la **Tabla 14** se realizó un formulario⁶ de Google Forms como se muestra en la **Figura 30** para los miembros del proyecto “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”. En la cual se incluyó una pregunta para identificar el rol (administrador/técnico) de los encuestados. Mediante esta encuesta se espera que los encuestados declaren su grado de aceptación de la aplicación móvil referente a la utilidad estimada.

⁶ <https://forms.gle/xc4KDc3sRPuLndRr8>

Evaluación de la aceptación de tecnología para la aplicación móvil desarrollada como parte del trabajo de titulación Front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”.

La siguiente encuesta se realizará como parte del trabajo de titulación denominado *Front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”*; y su objetivo es determinar la aceptación de tecnología por parte de los miembros del proyecto de investigación mencionado ante una aplicación móvil como herramienta de apoyo.

carmen.toledo@unl.edu.ec [Cambiar de cuenta](#)

* Indica que la pregunta es obligatoria

Figura 30. Cuestionario para evaluar la utilidad estimada en Google Forms

6.2.2. Fase 2: Tabulación de resultados considerando la escala Likert.

A continuación, en la **Tabla 15** se presentan los resultados generales de la evaluación TAM, lo cual puede evidenciarse en el **Recurso digital**.

Anexo 5. Para aplicarlo se utilizó un muestreo probabilístico por conveniencia. En el que participaron todos los miembros del proyecto “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” hasta la fecha, totalizando cinco personas. Entre ellos, el Dr. Darwin Pucha Cofrep, director del proyecto.

Tabla 15. Resultados de la evaluación TAM

Utilidad estimada	Nada/Totalmente en desacuerdo (%)	Bajo/En desacuerdo (%)	Normal/Ni de acuerdo, ni en desacuerdo (%)	Medio/De acuerdo (%)	Alto/Totalmente de acuerdo (%)
UE01	-	-	-	20	80
UE02	-	-	-	60	40
UE03	-	-	-	40	60
UE04	-	-	20	40	40
UE05	-	-	20	60	20

Utilidad Estimada (UE): Percepción de mejora en actividades.

En la **Tabla 16** y en la **Figura 31** se presentan los resultados únicamente del rol administrador.

Tabla 16. Resultados de la evaluación TAM por administradores

Utilidad estimada	Nada/Totalmente en desacuerdo (%)	Bajo/En desacuerdo (%)	Normal/Ni de acuerdo, ni en desacuerdo (%)	Medio/De acuerdo (%)	Alto/Totalmente de acuerdo (%)
UE01	-	-	-	50	50
UE02	-	-	-	100	-
UE03	-	-	-	50	50
UE04	-	-	50	50	-
UE05	-	-	-	100	-

Utilidad Estimada (UE): Percepción de mejora en actividades.

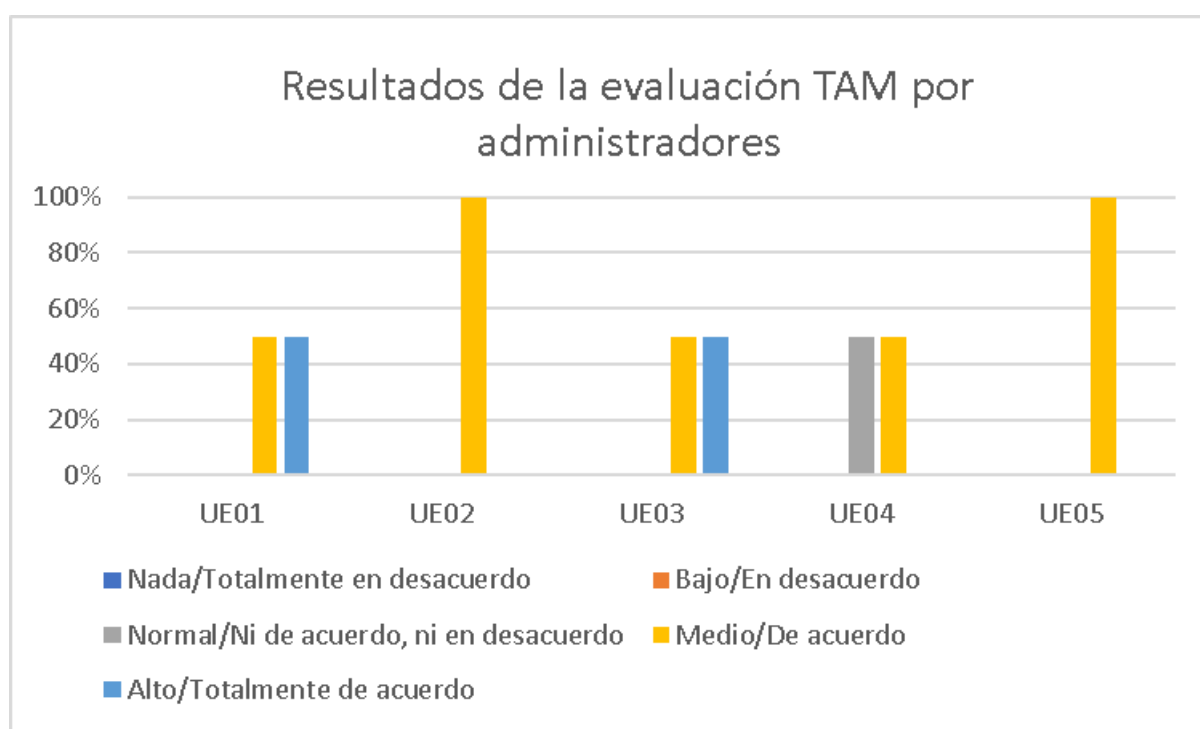


Figura 31. Resultados de la evaluación TAM por administradores

En la **Tabla 17** y en la **Figura 32** se presentan los resultados dados por los técnicos encuestados.

Tabla 17. Resultados de la evaluación TAM por técnicos

Utilidad estimada	Nada/Totalmente en desacuerdo (%)	Bajo/En desacuerdo (%)	Normal/Ni de acuerdo, ni en desacuerdo (%)	Medio/De acuerdo (%)	Alto/Totalmente de acuerdo (%)
UE01	-	-	-	-	100
UE02	-	-	-	66.66	33.33
UE03	-	-	-	33.33	66.66
UE04	-	-	-	33.33	66.66
UE05	-	-	33.33	33.33	33.33

Utilidad Estimada (UE): Percepción de mejora en actividades.

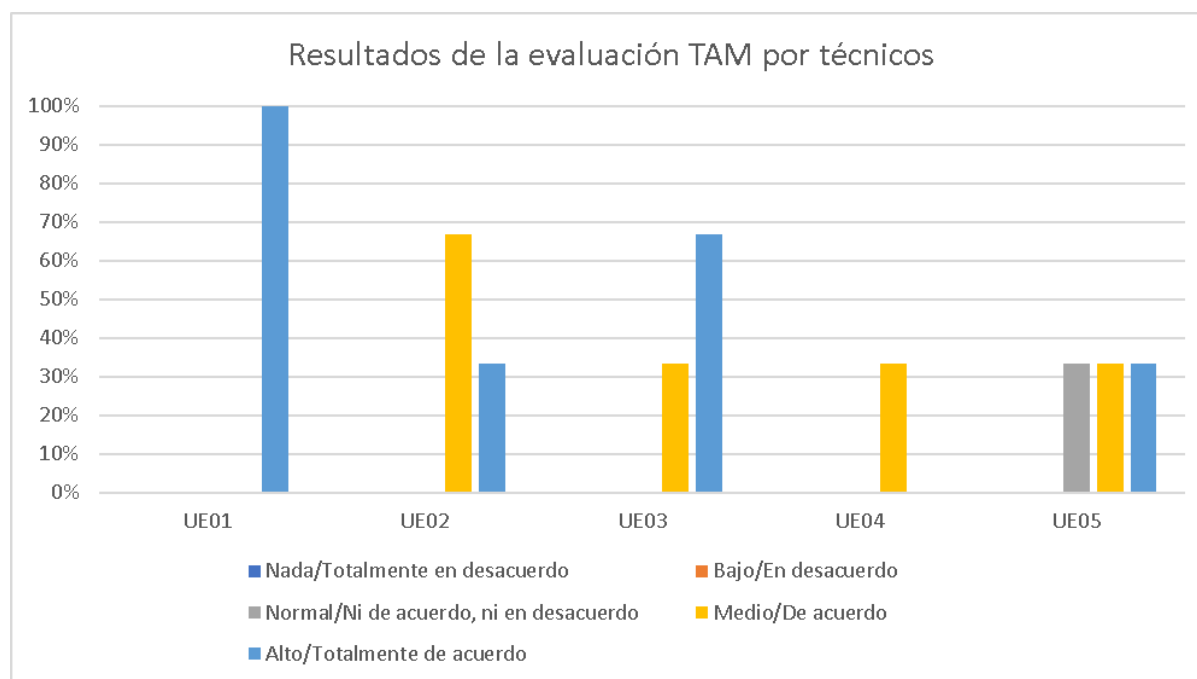


Figura 32. Resultados de la evaluación TAM por técnicos

En la **Tabla 18** se muestran los promedios ponderados por cada pregunta de la encuesta (UE01 a UE05) utilizando los porcentajes de las respuestas (incluyendo a técnicos y administradores) y los valores asignados a la escala Likert. En donde se aplica la siguiente fórmula.

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$$

En donde,

- \bar{x} es la media ponderada,
- x_i son los valores a promediar,
- w_i son los pesos correspondientes a cada valor,

- y n es el número total de valores.

Tabla 18. Resultados de la media ponderada

Pregunta	UE01	UE02	UE03	UE04	UE05
Valor	4.8	4.4	4.6	4.2	4.0
Escala	Alto/Totalmente de acuerdo	Alto/Totalmente de acuerdo	Alto/Totalmente de acuerdo	Medio/De acuerdo	Medio/De acuerdo

Media ponderada: Promedio que considera el peso de cada valor.

Con lo cual se obtiene el promedio general de 4.4 de los resultados, aplicando la fórmula de la media aritmética:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

En donde,

- \bar{x} es la media aritmética,
- x_i son los valores a promediar,
- y n es el número total de valores.

6.2.3. Fase 3: Análisis de resultados

Como se presentó en la **Tabla 15**, los encuestados manifestaron su postura respecto a que la aplicación móvil facilita la recopilación de datos en comparación con métodos tradicionales (papel y lápiz, hojas de cálculo, etc.) (UE01), las cuales fueron: el 80% de los encuestados indicaron estar en un grado alto/totalmente de acuerdo, mientras que el 20% indicaron estar en un grado medio/de acuerdo como se muestra en la **Figura 33**. Respecto al rol de los encuestados, y tomando en consideración la **Tabla 16** y **Tabla 17**, se evidencia que: el 50% de los administradores indicaron estar en un grado medio/de acuerdo, y el otro 50% estar en un grado alto/totalmente de acuerdo como se presenta en la **Figura 31**. Por otra parte, el 100% de los técnicos manifestaron estar en un grado alto/totalmente de acuerdo como se muestra en la **Figura 32**.

¿En qué grado considera usted que esta aplicación móvil facilita la recopilación de datos en comparación con métodos tradicionales (papel y lápiz, hojas de cálculo, etc.)?

5 respuestas

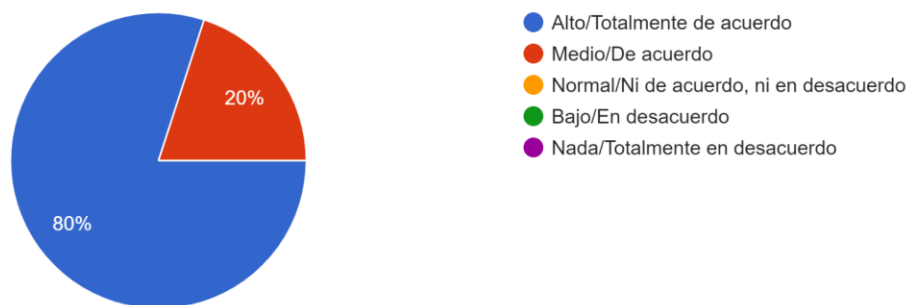


Figura 33. Pregunta UE01

Con respecto a que la aplicación móvil mejora la precisión y la calidad de los datos que recopila en comparación con métodos manuales (UE02) los encuestados manifestaron: el 60% están en un grado medio/de acuerdo y el 40% en un grado alto/totalmente de acuerdo, evidenciado en la **Figura 34**. Por parte de los administradores, el 100% manifestaron estar en un grado medio/de acuerdo; y, por parte de los técnicos, el 66.66% manifestaron estar en un grado medio/de acuerdo, el 33.33% restante en un grado alto/totalmente de acuerdo, como consta en la **Figura 31** y **Figura 32**.

¿En qué grado considera usted que esta aplicación móvil mejora la precisión y la calidad de los datos que recopila en comparación con métodos manuales?

5 respuestas



Figura 34. Pregunta UE02

En lo relacionado a que la aplicación móvil agiliza el proceso de captura de datos y ahorra tiempo en comparación con métodos convencionales (UE03), los encuestados manifestaron lo siguiente: el 60% mencionaron estar en un grado alto/totalmente de acuerdo y el 40% en un grado medio/de acuerdo, mostrado en la **Figura 35**. Por parte de los

administradores, el 50% están en un grado alto/totalmente de acuerdo y el otro 50% en un grado medio/de acuerdo, ilustrado en la **Figura 31**. Con respecto a los técnicos, el 66.66% están en un grado alto/totalmente de acuerdo y el 33.33% en un grado medio/de acuerdo, como se observa en la **Figura 32**.

¿En qué grado considera usted que esta aplicación móvil agiliza el proceso de captura de datos y ahorra tiempo en comparación con métodos convencionales?

5 respuestas

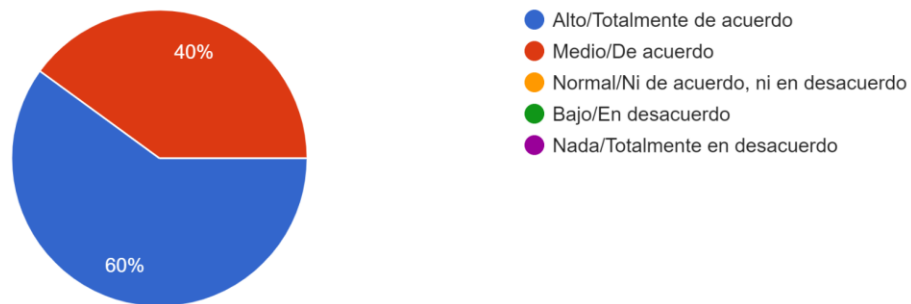


Figura 35. Pregunta UE03

Con respecto a la que la aplicación móvil brinda una mayor flexibilidad y accesibilidad para capturar datos en cualquier momento y lugar (UE04), los encuestados manifestaron lo siguiente: con un 20% están en un grado normal/ni de acuerdo, ni en desacuerdo, el 40% en un grado medio/de acuerdo y el 40% en un grado alto/totalmente de acuerdo, tal como se ilustra en la **Figura 36**. Por parte de los administradores el 50% manifestaron estar en un grado normal/ni de acuerdo, ni en desacuerdo y el otro 50% en un grado medio/de acuerdo evidenciado en la **Figura 31**. Por parte de los técnicos en un 33.33% están en un grado medio/de acuerdo y el 66.66% en un grado alto/totalmente de acuerdo presente en la **Figura 32**.

¿En qué grado considera usted que esta aplicación móvil brinda una mayor flexibilidad y accesibilidad para capturar datos en cualquier momento y lugar?

5 respuestas

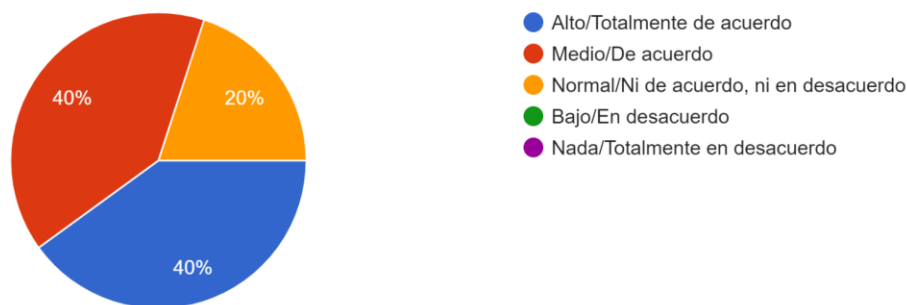


Figura 36. Pregunta UE04

Finalmente, en lo referente a que la aplicación móvil ayuda a minimizar errores y omisiones en la recopilación de datos (UE05), los encuestados manifestaron lo siguiente: el 20% están en un grado alto/totalmente de acuerdo, el 60% en un grado medio/de acuerdo y el restante 20% en un grado normal/ni de acuerdo, ni en desacuerdo que puede revisarse en la **Figura 37**. Por parte de los administradores, el 100% están en un grado alto/totalmente de acuerdo como se evidencia en la **Figura 31**. En contraste, el 33.33% de los técnicos están en un grado alto/totalmente de acuerdo, el 33.33% en un grado medio/de acuerdo y el restante 33.33% en un grado normal/ni de acuerdo, ni en desacuerdo ilustrado en la **Figura 32**.

¿En qué grado considera usted que esta aplicación móvil ayuda a minimizar errores y omisiones en la recopilación de datos?

5 respuestas

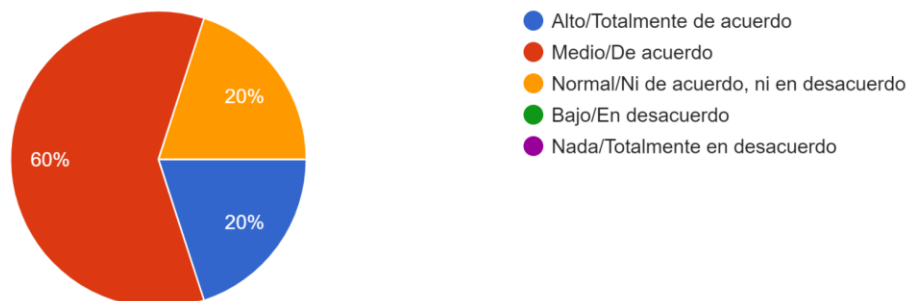


Figura 37. Pregunta UE05

7. Discusión

La realización del front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” se realizó trabajando de la mano con el Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc., como Docente Investigador encargado de la parte tecnológica del proyecto. Se utilizó la metodología Mobile-D para guiar el desarrollo y ejecución del TIC. El presente trabajo fue dividido en 2 fases, que corresponden a cada uno de los objetivos planteados con el fin de contestar a la pregunta de investigación. Las respectivas fases se discuten a continuación:

7.1. **Objetivo 1: Desarrollar el front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja” en base a los requerimientos actuales del proyecto.**

En base a la **Tabla 1** de la sección **3.2.4**, en donde se realizó una comparación entre las metodologías ágiles Mobile-D, XP y Scrum, se pudo determinar la factibilidad del uso de la metodología Mobile-D. En los estudios revisados [9], [10] y en el trabajo relacionado [37] se manifiesta el uso de esta metodología para el desarrollo móvil, lo cual se comprueba para el presente TIC, debido al grado de conocimiento del equipo de trabajo, su adaptabilidad a cambios y su relación con los casos de uso o historias de usuario. Como se menciona en el estudio [11] esta metodología se divide en 4 fases: exploración, inicialización, producción y estabilización. En la primera fase, para el establecimiento de grupos de interés, se empleó la técnica de entrevista (ver **Anexo 1**), con lo cual pudo efectuarse la elicitación de requerimientos documentado mediante el estándar IEEE-830, presente en el **Anexo 2**.

En la fase de inicialización, se realizó el diseño de la aplicación mediante los diagramas de casos de uso, de despliegue y componentes (ver sección **6.1.2.1**); diagramados mediante la herramienta Enterprise Architecture. Por otro lado, para el diseño de interfaces (ver sección **6.1.2.2**), se utilizó la herramienta Figma. Posteriormente, en la fase de producción, la planificación de iteraciones (ver sección **6.1.3.1**) estuvo guiada por las historias de usuarios previamente definidas en la elicitación de requerimientos. En esta fase se utilizó el SDK Flutter y el lenguaje de programación Dart (ver sección **3.3**), aplicando la arquitectura limpia propuesta por Robert C. Martin, mencionada en la sección **3.4.1**. Al aplicar la arquitectura, surgieron problemas referentes al flujo de llamadas del sistema, lo cual se realiza mediante el paquete `get_it` para la localización de servicios y el acceso de manera global a ellos. Dado

que el sistema está dividido en módulos, la localización de los servicios debe realizarse para cada uno de ellos; para solucionar este problema se aplicó el patrón Facade mencionado en la sección **3.4.2**, que ayudó a simplificar su organización, realizando una única instancia desde la función principal.

Cabe recalcar que otras limitaciones presentes en esta fase surgieron durante la implementación, en la que existieron problemas con las librerías relacionadas con la visualización de datos en mapas; la gran mayoría de estas librerías se encontraban depreciadas (deprecates) y sin mantenimiento por parte de la comunidad de Flutter; este inconveniente se pudo solventar al incluir criterios de popularidad y revisión de la frecuencia de lanzamiento de versiones estables para la selección de las librerías.

Uno de los principios de la metodología Mobile-D es que su desarrollo está guiado por pruebas (TDD) [11]. Por ello, durante la etapa de pruebas del sistema, se crearon pruebas unitarias previas al desarrollo de las funcionalidades, que ayudaron a entender su comportamiento esperado y aseguraron su correcto funcionamiento. Dichas pruebas fueron estructuradas siguiendo el patrón Arrange Act Assert, en donde pudo constatar que se adaptan a TDD como se menciona en la sección **3.4.2**. De esta manera, se realizaron un total de 11 pruebas unitarias para verificar el correcto funcionamiento e integración de módulos. Es importante destacar que no fue necesario realizar pruebas unitarias a lo largo de todo el proceso de desarrollo. Esto debido a que, al implementar la arquitectura limpia, cada módulo fue estructurado de manera similar, de tal manera que, al ejecutar estas pruebas en los módulos importantes como: autorización-autenticación de usuarios, y gestión de usuarios, quedó establecido el comportamiento esperado para los módulos restantes.

7.2. Objetivo 2: Evaluar la aplicación móvil mediante la utilidad estimada del modelo de aceptación de tecnología (TAM) para determinar el grado de soporte que brinda en el inventario in-situ.

Conforme los estudios analizados [31], [32], se comprueba que el modelo de aceptación de tecnología TAM ayuda a determinar la aceptación de una tecnología por parte de un grupo determinado; para el caso del sistema, se evaluó la utilidad estimada conforme este modelo, tal como se menciona en el trabajo relacionado [33], empleando la escala Likert como parte de la elaboración del cuestionario presentado en la **Tabla 14**, fue posible determinar el grado de soporte de la aplicación móvil para el inventario in-situ como herramienta del proyecto de investigación.

De acuerdo a los resultados de la encuesta aplicada, que se pueden visualizar en la **Figura 38**. En lo referente a que la aplicación móvil facilita la recopilación de datos en comparación con métodos tradicionales de la pregunta UE01, el 80% de los encuestados manifiestan estar en un grado alto/totalmente de acuerdo; por otro lado, respecto a que la aplicación móvil mejora la precisión y la calidad de los datos que recopila en comparación con métodos manuales de la pregunta UE02, el 60% están en un grado medio/de acuerdo; en lo relacionado a que la aplicación móvil agiliza el proceso de captura de datos y ahorra tiempo en comparación con métodos convencionales de la pregunta UE03, el 60% mencionaron estar en un grado alto/totalmente de acuerdo; con respecto a que la aplicación móvil brinda una mayor flexibilidad y accesibilidad para capturar datos en cualquier momento y lugar de la pregunta UE04, cuenta con un 80% de aceptación dentro de los rangos entre medio y alto; finalmente, en lo referente a que la aplicación móvil ayuda a minimizar errores y omisiones en la recopilación de datos de la pregunta UE05, el 80% manifestó estar entre un grado medio y alto. Tomando en cuenta el promedio de 4.4 obtenido en la ecuación para la media aritmética, y considerando que este resultado corresponde a un nivel “Alto/Totalmente de acuerdo” en la escala de Likert según lo valores definidos en la **Tabla 4**, se puede concluir que la aplicación móvil se encuentra en un grado alto de soporte para el inventario in-situ como herramienta del proyecto de investigación; dando así contestación a la pregunta de investigación planteada: ¿En qué grado el front-end versión2.0 brinda soporte para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”?

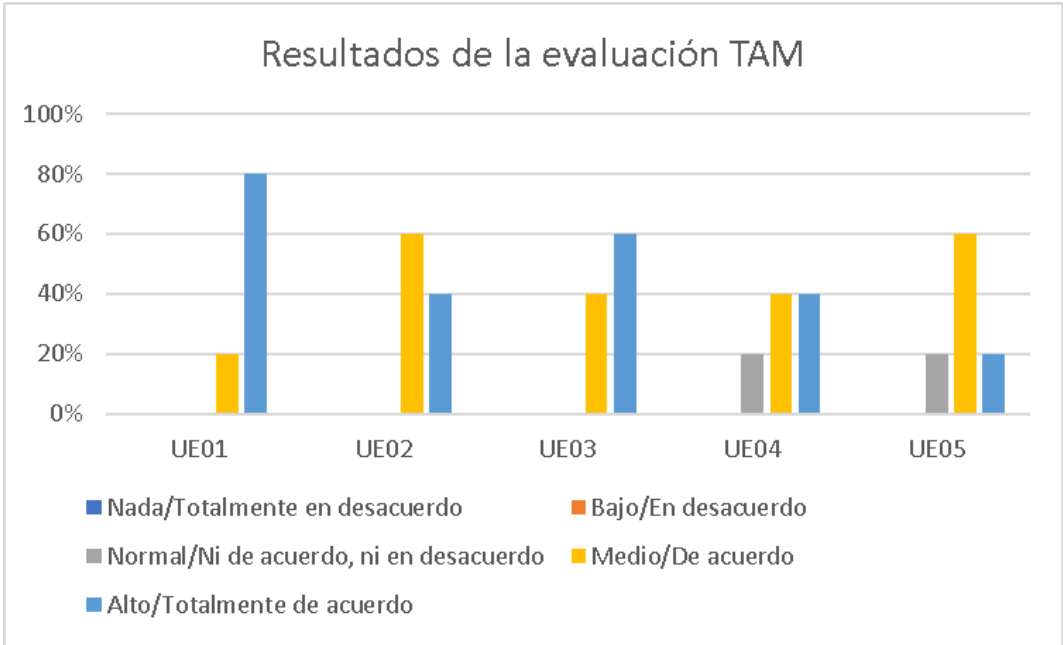


Figura 38. Resultados de la evaluación TAM

8. Conclusiones

Tras finalizar el presente TIC, se ha llegado a las siguientes conclusiones:

- La elección de la metodología ágil Mobile-D para guiar el desarrollo de la aplicación demostró ser acertada, debido a que permitió el cumplimiento de las 14 historias de usuario definidas, mediante la ejecución de las fases de exploración, inicialización, producción y estabilización; facilitando la documentación del diseño arquitectónico y la codificación.
- La implementación de una arquitectura limpia a través de un estilo arquitectónico por capas demostró ser beneficiosa al estructurar los módulos de manera homogénea, lo que permitió el uso y reutilización de pruebas unitarias en la resolución de problemas. Además, la aplicación del patrón de diseño Facade permitió acceder a las llamadas de los servicios de negocio de una manera organizada y centralizada; mejorando así mantenibilidad del código.
- La adopción del desarrollo guiado por pruebas (TDD) y la aplicación del patrón de diseño Arrange Act Assert en las pruebas unitarias al inicio de la fase de producción, fue esencial para definir el comportamiento de los módulos previo al desarrollo, identificando el flujo normal y el flujo de posibles excepciones (errores) que pueden ocurrir en el comportamiento; con esto se garantizó que las funcionalidades operen de manera adecuada una vez ejecutadas.
- La aplicación del Modelo de Aceptación de Tecnología (TAM) ayudó al momento de evaluar el grado de soporte de la aplicación móvil en el levantamiento de datos in-situ, permitiendo identificar aspectos clave de la utilidad percibida, como la facilidad en la recopilación de datos, la mejora en la precisión y agilidad en el proceso de captura. Además, la escala Likert utilizada en el cuestionario proporcionó una métrica cualitativa para esta evaluación.
- Los resultados obtenidos a través de la evaluación del Modelo TAM indicaron que la aplicación móvil brinda un grado alto de soporte como herramienta del proyecto de investigación, teniendo altos niveles de aceptación en aspectos como la facilidad en la recopilación de datos y la agilidad en el proceso de captura, reforzando la contribución positiva de la herramienta al proyecto. Sin embargo, existen áreas donde se podría mejorar, por ejemplo, la precisión en la geolocalización de datos, lo que sugiere oportunidades para el desarrollo continuo y refinamiento de la aplicación.

9. Recomendaciones

Al finalizar el presente TIC, se enlistan las siguientes recomendaciones:

- Para optimizar el proceso de desarrollo en nuevos proyectos, se recomienda combinar de manera estratégica una arquitectura limpia, para facilitar la organización y mantenimiento del código a lo largo del desarrollo; TDD, para crear pruebas unitarias antes de escribir el código, asegurando la funcionalidad y robustez del sistema; y la metodología Mobile-D, ya que puede adaptarse a los cambios y requerimientos emergentes.
- Para el desarrollo de pruebas unitarias se recomienda el uso de la librería mocktail, puesto que permite la simulación del comportamiento de objetos de forma controlada facilitando así la ejecución de las pruebas unitarias en Flutter.
- La inclusión de diferentes perfiles de usuarios, como usuarios no técnicos, podría aportar perspectivas adicionales sobre la utilidad y el impacto de la aplicación en diferentes contextos de uso.
- Se puede mejorar la geoubicación de los árboles in-situ utilizando librerías certificadas empresariales, las cuales poseen una actualización continua de mapas y precisión en la geolocalización a través del GPS de los dispositivos móviles.

9.1. Trabajos futuros

A pesar de los avances significativos logrados en el desarrollo del front-end versión 2.0 para el inventario in-situ como herramienta del proyecto de investigación “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”, existen áreas de mejora y oportunidades para la expansión de funcionalidades que podrían abordarse en futuras etapas de desarrollo. Uno de los enfoques clave sería la integración de características avanzadas de captura de fotos y la implementación de filtros avanzados para la búsqueda de árboles. Lo cual podría realizarse a través de la colaboración con pasantes y expertos en áreas específicas.

10. Bibliografía

- [1] L. Pozo, “Implementación de un sistema de inventario forestal de parques urbanos en la ciudad de Cuenca”, *Universidad-Verdad*, núm. 73, pp. 79–89, 2017.
- [2] C. Konijnendijk, R. Ricard, A. Kenney, y T. Randrup, “Defining urban forestry A comparative perspective of North America and Europe”, *Urban For Urban Green*, vol. 4, núm. 3–4, pp. 93–103, 2006.
- [3] D. Pacheco y L. Ávila, “Inventario de parques y jardines de la ciudad de Cuenca con UAV y smartphones”, *Memorias y Boletines de la Universidad del Azuay*, vol. 1, núm. XVI, pp. 173–179, 2017.
- [4] Municipio De Loja, “Plan de desarrollo general”, 2021. Consultado: el 12 de mayo de 2023. [En línea]. Disponible en: <https://www.loja.gob.ec/contenido/plan-de-desarrollo-y-ordenamiento-territorial>
- [5] H. Flora y S. Chande, “A review and analysis on mobile application development processes using agile methodologies”, *International Journal of Research in Computer Science*, vol. 3, núm. 4, p. 9, 2013.
- [6] B. Choudhary y S. Rakesh, “An approach using agile method for software development”, en *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, 2016, pp. 155–158. doi: 10.1109/ICICCS.2016.7542304.
- [7] M. Navarro, M. Moreno, J. Aranda, L. Parra, J. Rueda, y J. Pantano, “Selección de Metodologías Ágiles e Integración de Arquitecturas de Software en el Desarrollo de Sistemas de Información”, en *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*, 2017.
- [8] A. Cadavid, J. Martínez, y J. Vélez, “Revisión de metodologías ágiles para el desarrollo de software”, *Prospectiva*, vol. 11, núm. 2, pp. 30–39, 2013.
- [9] P. Abrahamsson *et al.*, “Mobile-D: an agile approach for mobile application development”, en *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, 2004, pp. 174–175.
- [10] A. Asfour, S. Zain, N. Salleh, y J. Grundy, “Exploring agile mobile app development in industrial contexts: A qualitative study”, *International Journal of Technology in Education*

- and Science (IJTES)*, vol. 3, núm. 1, pp. 29–46, 2019, [En línea]. Disponible en: www.ijtes.net
- [11] R. Castilla, A. Pacheco, y J. Franco, “Digital government: Mobile applications and their impact on access to public information”, *SoftwareX*, vol. 22, p. 101382, 2023, doi: <https://doi.org/10.1016/j.softx.2023.101382>.
- [12] E. Bahit, *Scrum y eXtreme Programming para programadores*, vol. 1. Autoedición, 2012.
- [13] J. Highsmith, *Agile software development ecosystems*, vol. 1. Addison-Wesley Professional, 2002.
- [14] P. Letelier, “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)”, *Universidad Politécnica de Valencia*, vol. 17, 2012.
- [15] P. Blanco, J. Camarero, A. Fumero, A. Woterski, y P. Rodríguez, “Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone”, *Dr. en Ing. Sist. Telemáticos*, pp. 1–30, 2009.
- [16] K. Beck, *Extreme Programming Explained, Second.*, vol. 2. 2000.
- [17] O. Tinoco, P. Rosales, y J. Salas, “Criterios de selección de metodologías de desarrollo de software”, *Industrial data*, vol. 13, núm. 2, pp. 70–74, 2010.
- [18] M. Napoli, *Beginning flutter: a hands on guide to app development*. John Wiley & Sons, 2019.
- [19] R. Payne, *Beginning app development with flutter: Create cross-platform mobile apps*. Dallas: Springer, 2019.
- [20] Standart, *Dart Programming Language Specification*, vol. 6. Addison-Wesley Professional, 2023.
- [21] G. Bracha, *The Dart programming language*, vol. 1. Addison-Wesley Professional, 2015.
- [22] R. C. Martin, *Clean Architecture: A Craftsman’s Guide to Software Structure and Design*, vol. 1. Prentice Hall, 2019.

- [23] R. Martin, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*, vol. 1. Prentice Hall, 2019.
- [24] V. Khorikov, *Unit Testing Principles, Practices, and Patterns*. Simon and Schuster, 2020.
- [25] O. Blancarte, *Introducción a los patrones de diseño: Un enfoque práctico*, 1a ed. CreateSpace Independent Publishing Platform, 2016. [En línea]. Disponible en: <http://gen.lib.rus.ec/book/index.php?md5=57BD94E464E701F03B95F06BC02C1663>
- [26] M. Fowler, *Patterns of enterprise application architecture*, 21. print. en The Addison-Wesley signature series. Addison-Wesley, 2015. [En línea]. Disponible en: <http://gen.lib.rus.ec/book/index.php?md5=3606CDE4B85BF15BDE3FE0A376631EE8>
- [27] ISTQB, "Probador Certificado del ISTQB®", 2018.
- [28] D. Bolaños, A. Sierra, y M. Alarcón, *Pruebas de software y JUnit: un análisis en profundidad y ejemplos prácticos*. Pearson Educación, 2008. [En línea]. Disponible en: <http://gen.lib.rus.ec/book/index.php?md5=F86C4A0971C3A4B1A2C680B7605A4C0D>
- [29] R. Pressman, *Ingeniería del software: un enfoque práctico*, Séptima edición. 2013. [En línea]. Disponible en: <http://gen.lib.rus.ec/book/index.php?md5=927F3EBCBEEAD67C38E08FCE46279565>
- [30] A. Mora y J. Guazha, "Módulo para la gestión de trámites de la Procuraduría General de la Universidad Nacional de Loja", 2023.
- [31] P. Puello, V. Del Campo, y F. Scholborgh, "Modelo de Aceptación Tecnológica (TAM) en el Laboratorio de Física III basado en Internet de las Cosas en el Programa de Ingeniería de Sistemas de la Universidad de Cartagena, Colombia", *Revista ESPACIOS. ISSN*, vol. 798, p. 1015, 2020.
- [32] L. López y J. López, "Estudio comparado de las estimaciones de dos versiones del modelo de aceptación de la tecnología (TAM) mediante los programas AMOS y PLS", *Investigaciones europeas de dirección y economía de la empresa*, vol. 12, núm. 3, pp. 95–110, 2006.
- [33] D. Leyton, "Extensión al modelo de aceptación de tecnología TAM, para ser aplicado a sistemas colaborativos, en el contexto de pequeñas y medianas empresas", 2013.

- [34] X. Liang *et al.*, “In-situ measurements from mobile platforms: An emerging approach to address the old challenges associated with forest inventories”, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 143, pp. 97–107, 2018.
- [35] A. Tashildar, N. Shah, R. Gala, T. Giri, y P. Chavhan, “Application development using flutter”, *International Research Journal of Modernization in Engineering Technology and Science*, vol. 2, núm. 8, pp. 1262–1266, 2020.
- [36] V. Ibañez, D. Ramos, W. Jimenez, A. Apaza, M. Ibarra, y Y. Ponce, “Kusa-yupaq: Mobile application to improve the inventory of heritage assets”, en *2020 XLVI Latin American Computing Conference (CLEI)*, IEEE, 2020, pp. 387–392.
- [37] P. Bautista, “Aplicación móvil usando el framework Flutter para el control de las tutorías académicas del sistema integrado de la Universidad Técnica de Ambato”, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, 2022.

11. Anexos

Anexo 1. Entrevista para la elicitación de requerimientos

Recurso digital.

Anexo 2. Requerimientos del sistema y prototipado de interfaces de usuario

Recurso digital.

Anexo 3. Pruebas unitarias

Recurso digital.

Anexo 4. Pruebas de aceptación

Recurso digital.

Anexo 5. Resultados de la evaluación de la utilidad estimada del modelo TAM

Recurso digital.

Anexo 6. Manual de usuario

Recurso digital.

Anexo 7. Manual de Desarrollador

Recurso digital.

Anexo 8. Acta de entrega del sistema

Recurso digital.

Anexo 9. Informe de pertinencia del proyecto de Trabajo de Integración Curricular

Recurso digital.

Anexo 10. Certificado de traducción del resumen

Recurso digital.