Universidad Nacional de Loja

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

Carrera de Computación

Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja

Proposal for a self-managed academic digital identity using Blockchain technology for the Universidad Nacional de Loja

Trabajo de Integración Curricular, previo a la obtención del título de Ingeniero en Ciencias de la Computación

AUTOR
Carlos Alexis Armijos Rios

DIRECTOR:
Ing. Cristian Ramiro Narváez Guillen, Mg. Sc.

Loja – Ecuador
2024

Certificación

Loja, 13 de agosto de 2024

Ing. Cristian Ramiro Narváez Guillen, Mg. Sc.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Integración Curricular denominado: **Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja,** previo a la obtención del título de **Ingeniero en Ciencias de la Computación**, de la autoría del estudiante **Carlos Alexis Armijos Rios**, con **cédula de identidad** Nro. **1900549179**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Firmado electrónicamente por:
CRISTIAN RAMIRO
NARVAEZ GUILLEN

Ing. Cristian Ramiro Narváez Guillen, Mg. Sc.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

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

Firma: 

Cédula de identidad: 1900549179

Fecha: 13 de agosto de 2024

Correo electrónico: carlos.a.armijos@unl.edu.ec

Teléfono: (+593) 95 944 1416

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

Yo, **Carlos Alexis Armijos Rios**, declaro ser autor del Trabajo de Integración Curricular denominado: **Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja,** como requisito para optar el título de **Ingeniero 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 en el Repositorio Institucional.

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

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

Para constancia de esta autorización, suscribo, en la ciudad de Loja, a los trece días del mes de agosto de dos mil veinticuatro.

Firma: Firmado electrónicamente por: CARLOS ALEXIS ARMIJOS RIOS

Autor: Carlos Alexis Armijos Rios
Cédula: 1900549179
Dirección: Av. Eduardo Kingman, Loja – Ecuador.
Correo Electrónico: carlos.a.armijos@unl.edu.ec
Celular: (+593) 95 944 1416

DATOS COMPLEMENTARIOS:

Director del Trabajo de Integración Curricular: Ing. Cristian Ramiro Narváez Guillen, Mg.Sc.

Dedicatoria

Dedico este trabajo orgullo y especialmente a mí madre, Sandra, y a mí padre, Carlos, quienes a través de sus enseñanzas, educación y amor han logrado que su hijo pueda cumplir con cualquier meta que se proponga. A mi hermana María que, como ejemplo de hermano mayor, sepa que cualquier cosa que nos propongamos la podremos cumplir. Sin olvidarme de mis compañeros Beiker, Jaime y Jessica con quienes hemos compartido un montón de experiencias y hemos logrado seguir adelante a través de la carrera. Muchas gracias.

Carlos Alexis Armijos Rios

Agradecimiento

Agradecer a mi madre, padre y hermana por haberme apoyado a lo largo de esta meta.

De igual manera, agradecer a la Universidad Nacional de Loja y especialmente a la carrera de Ingeniería en Computación por estos años de formación académica, y a cada uno de los docentes que compartieron sus conocimientos y experiencias.

También agradecer a mi director del Trabajo de Integración Curricular, Ing. Cristian Ramiro Narváez Guillen, que gracias a sus conocimientos y apoyo se logró culminar con este desafiante proyecto.

Y, por último, a mis compañeros de clase y sobre todos a mis amigos por todos esos buenos momentos dentro y fuera del aula.

Muchas gracias por su apoyo y sus ánimos.

Carlos Alexis Armijos Rios

Índice de Contenidos

Índice de Tablas:

Índice de Figuras:

Índice de Anexos:

1. Título

Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja

Proposal for a self-managed academic digital identity using Blockchain technology for the Universidad Nacional de Loja

2. Resumen

La identidad de una persona se crea según sus acciones, sus comportamientos, y por su información personal que hace lo hace único dentro de la sociedad. Lo mismo se aplica en la identidad digital, pero que es más propensa a ser robada, divulgada y manipulada, siendo lo más crítico la suplantación de identidad por parte de otra persona. Para evitar que estos inconvenientes afecten y perjudiquen en el ámbito académico, se construye un modelo de identidad digital académica con las características de autogestión. Para lograrlo, se utiliza la Blockchain de Hyperledger Indy que ofrece características idóneas para asegurar la inmutabilidad, integridad y transparencia de la identidad digital, a través de los métodos Wallet y DID que conjuntamente se convierte en la billetera virtual que contendrá la información personal de los usuarios. Para llevar a cabo este proyecto se establecieron tres fases de acuerdo a los objetivos especificados aplicando el proceso de desarrollo de la metodología ABCDE. En la primera fase de definición, se obtuvieron los actores, requisitos funcionales y no funcionales mediante el estándar IEEE 830, además de la arquitectura de la DApp; en la segunda fase de desarrollo, se construyeron y se probaron los subsistemas de contratos inteligentes (chaincode) compuesta por el módulo “Back-end” que integra la red Blockchain, y de aplicación compuesto por los módulos “Front-end” y “Middleware”; y en la última fase de comprobación, se ejecutaron pruebas de integración y funcionales, y a partir de una muestra de estudiantes se determinó que la DApp es aceptable. Finalmente, la implementación de la Blockchain de Hyperledger Indy otorga a los usuarios una identidad digital válida y confiable, también permite que intercambien información mediante los Esquemas de Transcripción, pero sobre todo les concede el control total sobre su información, lo que implica el aseguramiento de la identidad digital académica.

Palabras clave: *Blockchain, Hyperledger Indy, Identidad Digital, ABCDE, DApp, IEEE*.

Abstract

The identity of a person is created by their actions, their behaviors performed, and by their personal information that makes them unique within society. The same applies to digital identity, but it is more prone to being stolen, disclosed and manipulated, the most critical being impersonation of identity by another person. In order to prevent these problems from affecting and damaging the academic environment, a model of academic digital identity with the characteristics of self-management is constructed. For that reason, the Hyperledger Indy Blockchain was used, which offers ideal characteristics to ensure the immutability, integrity, and transparency of the digital identity, through the Wallet and DID methods that together become the virtual wallet that will contain users' personal information. To carry out this project, three phases were established according to the specified objectives by applying the development process of the ABCDE methodology. In the first definition phase, the actors, functional and non-functional requirements were obtained through the IEEE 830 standard, in addition to the DApp architecture; in the second development phase, the smart contract subsystems (chain code) composed of the "Back-end" module that integrates the Blockchain network, which integrates the Blockchain network, and of application composed by "Front-end" and "Middleware" modules; and in the last verification phase, integration and functional tests were executed, and from a sample of students it was determined that the DApp is acceptable. Finally, the implementation of the Hyperledger Indy Blockchain granted users a valid and reliable digital identity, it also allowed them to exchange information through Transcription Schemes, but above all, it granted them full control over their information, which implied the assurance of the academic digital identity.

Keywords: *Blockchain, Hyperledger Indy, Digital Identity, ABCDE, DApp, IEEE*.

3. Introducción

En la actualidad es común recibir publicidad sin previo aviso. Este tipo de publicidad, ofrece un nuevo producto o una mejora del mismo, puede ser recibida por mensajes de texto, llamadas telefónicas o correos electrónicos, dónde este último elemento en 2008 tuvo un 85% de correos con publicidad mientras que el 2019 se redujo a un 39% [1]. Todo esto es posible debido a que la información personal de los usuarios está bajo control de empresas u organizaciones que buscan obtener beneficio, para ello crean sistemas de recopilación y retención con el objetivo de tener a sus clientes activos, como por ejemplo con regalos, ofertas especiales o rebajas, para llamar la atención de las personas, pero antes deben rellenar un formulario con sus datos [2]. Lo que los clientes no conocen es que están dando su consentimiento sobre el uso de su información.

En Ecuador, la información de al menos 20,8 millones de ciudadanos se divulgó el 16 de septiembre del 2019. Dónde sus datos personales fueron vendidos como una mercancía a empresas que comercializan bienes o servicios, esto ocurrió gracias a la vulnerabilidad de las bases de datos de las instituciones públicas y privadas [3]. Sin embargo, lo preocupante de esta filtración de datos fue el nivel de detalle sobre la identidad de cada ecuatoriano, nombres completos, fechas de nacimiento, direcciones de domicilio, número de cédula, el historial laboral, nivel educativo, datos crediticios, entre otra información personal.

La identidad digital concede a las personas una identidad en el mundo digital con la misma validez y confianza que en el mundo físico, además agregando la característica de autogestión se permite que el usuario tenga el control total de toda la información que compone su identidad, y con ayuda de la tecnología Blockchain se asegura que toda la información no podrá ser divulgada sin el consentimiento de la persona. Esta identidad digital autogestionada se busca conseguir con la propuesta de este Proyecto de Integración Curricular, enfocado al área académica.

Al orientar la identidad digital auto-gestionada a lo académico, asegura a los usuarios de universidades, colegios o demás instituciones educativas la confianza de que su información personal nunca será una moneda de cambio cuando necesiten acceder a los diferentes conocimientos digitales (libros digitales, monografías, revistas, etc.).

El Proyecto de Integración Curricular denominado “Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja”, tiene como objetivo general construir un prototipo de módulo de software para la identidad digital académica auto-gestionada mediante tecnología Blockchain que permita a sus usuarios tener una identificación digital válida y confiable gracias a las características que ofrece Hyperledger Indy en este ámbito. Para completar este proyecto se divide el objetivo

general en 3 objetivos específicos: 1. Definir el sistema para la identidad digital académica auto-gestionada usando la Ingeniería de Requisitos, 2. Desarrollar el sistema para la identidad digital académica auto-gestionada mediante tecnología Blockchain, y 3. Probar el sistema de identidad digital académica auto-gestionada en un ambiente controlado.

4. Marco teórico

4.1. Identidad

4.1.1. Identidad Humana

La identidad humana puede definirse como una serie de características que hacen a una persona quien es y la distinguen de los demás, que le permiten interactuar con su entorno [4]. Se construye sobre la base de la propia situación de la persona, pero también sobre la base de hechos y experiencias vividas, cambia con el tiempo. Además, la necesidad de tener una identidad es fundamental para el humano. Por lo tanto, la identidad es un núcleo maleable capaz de cambiar durante la vida, formado a partir de la interacción con el entorno, ya que las características individuales por sí solas no importan [5]. Finalmente, es en la interacción con los demás que las diferencias y las características individuales cobran valor y emergen como contribuciones a la interacción social.

4.1.2. Identidad Digital

La identidad digital consta de diferentes tipos de datos dependiendo de si el usuario quiere revelarlos, dando como resultado: una identidad pública, incluye información que una persona divulga, otra identidad actuante, en base a las acciones que una persona realiza, y otra identidad inferida, según el análisis de la sociedad hacia las acciones que realiza una persona. Toda esta información se puede utilizar para desarrollar una idea de quién es una persona [5].

Específicamente, los tipos de datos que ayudan a configurar esta identidad digital se pueden agrupar de la siguiente manera:

- **Datos de identificación personal:** son identificadores como nombre, DNI, número de licencia de conducir, número de tarjeta de crédito, fecha de nacimiento, identificadores sociales de los sitios web visitados, etc.
- **Datos de comportamiento:** información de transacciones, historial de navegación, datos de ubicación, registros de llamadas, historial de compras, etc.
- **Datos derivados:** estos son atributos modelados analíticamente que se utilizan para analizar personas, por ejemplo, para evaluar el riesgo de prestar dinero a un cliente, evaluar su impacto en algún ámbito, etc.

En la Tabla 1 se muestran algunos de los principales beneficios de tener una identidad digital dependiendo del enfoque.

Tabla 1: *Beneficios de la identidad digital* [6].

Beneficios para las personas	Beneficios para el sector público	Beneficios para el sector privado
Experiencia de usuarios	Mejor prestación de servicios	Reducción de costes por prestación de servicios
Conveniencia	Reducción de costes personales	Oportunidades comerciales en ciberseguridad
Utilidad	Reducción de costes de procesos en papel y almacenamiento	Oportunidades comerciales como proveedores de identidad
Reducción de costes	Reducción de costes de prestación de servicios	Mayor accesibilidad de cliente
Inclusión	Incremento de la seguridad	Facilitación de procesos de verificación de usuarios

La identidad digital permite a las personas escapar de las limitaciones del mundo físico y proporciona comunicaciones globales confiables. En un mundo cada vez más digital, se necesitan sistemas de información confiables, eficientes y escalables que brinden la seguridad y autenticidad para identificar con quién interactuamos y, sobre todo que permitan al usuario tener el control de sus datos.

4.1.3. Identidad Auto-Gestionada

La identidad auto-gestionada (IAG) es el reconocimiento digital de un individuo que posee y controla su identidad sin la intervención de otras entidades o usuarios, permitiendo que pueda interactuar en el mundo digital con total libertad y confianza que en el mundo físico [6]. Los 10 principios para la identidad auto-gestionada son:

- **Acceso**: los usuarios deben tener acceso a su propia información.
- **Consentimiento**: los usuarios deben dar su consentimiento para que su identidad sea compartida o usada por otros.
- **Control**: los usuarios deben tener el control de su identidad.
- **Existencia**: los usuarios deben poseer una existencia única, independiente y válida.
- **Interoperabilidad**: las identidades deben interactuar libremente y con seguridad con empresas, organizaciones, etc.
- **Minimización**: la divulgación de información de identidad debe ser mínima.
- **Persistencia**: las identidades deben ser permanentes.
- **Protección**: se deben proteger los derechos de los usuarios.
- **Portabilidad**: la información y los servicios de identidad deben ser portables.
- **Transparencia**: los sistemas y algoritmos deben ser transparentes.

En el modelo de IAG, los verificadores no necesitan solicitar información directamente al emisor ni confirmarla con autoridades de confianza. Esta es una de las principales diferencias y ventajas sobre otros modelos de gestión de identidades. Desde la perspectiva

de la autogestión, la validez de todos los activos digitales se puede verificar contra un registro de información descentralizado y confiable. Esto es gracias a que cada vez que se emiten estos activos (tokens, credenciales digitales, etc), el emisor registra en la red descentralizada (blockchain) una prueba criptográfica de la emisión, así como el tiempo acreditado. El emisor también registra el estado del activo, que puede ser modificado en cualquier momento por él mismo o por cualquier entidad autorizada por él, sujeto al reglamento y a ciertas reglas de transparencia, confidencialidad y seguridad [6].

Estos registros descentralizados en los que se guarda cualquier acción son inmutables, por lo que si se modifica o actualiza información como el estado de una credencial digital quedará constancia de la alteración, ya que todas las modificaciones son firmadas electrónicamente por la entidad y queda guardada en el registro descentralizado.

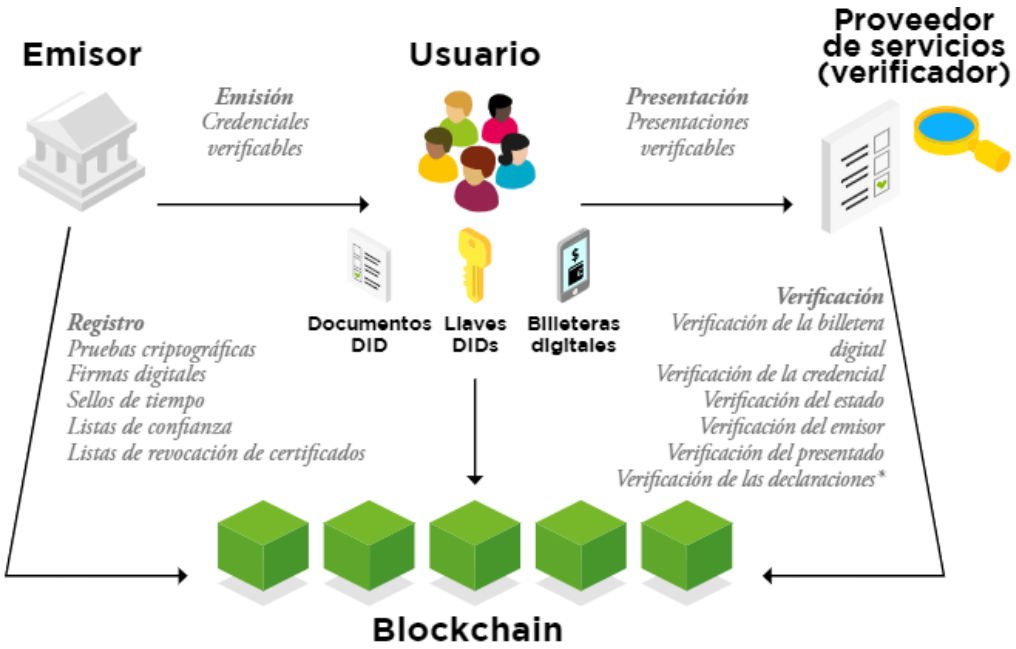


Figura 1: *Esquema básico para comprender la Identidad Digital Auto-gestionada.*
Fuente*:* [6].

4.2. Blockchain

Es una base de datos descentralizada que permite almacenar información inmutable, segura y verídica. La información solo puede ser añadida si existe un acuerdo entre la mayoría de los nodos que conforman la red blockchain [7].

4.2.1. Importancia

La blockchain es idónea para administrar la información de cualquier tipo de entidad o empresa, ya que permite que los usuarios autorizados puedan acceder a información confiable, también permite dar seguimiento a pedidos, pagos, cuentas, detalles de producción, etc. [8].

4.2.1.1. Elementos claves

Para entender el funcionamiento de una blockchain es importante conocer [8]:

- Todos los nodos que la conforman tienen acceso tanto para leer y escribir el registro de transacciones, además que cada transacción es un registro único lo que elimina la duplicidad o posibles alteraciones de la información.
- Ninguna transacción puede ser modificada, eliminada o alterada una vez registrada, en caso de existir algún inconveniente con la transacción se deberá registrar una nueva transacción con las correcciones necesarias.
- Los contratos inteligentes son un conjunto de reglas que determinan que información será transaccionada, pudiendo ser historiales clínicos, certificados o récords académicos, dinero digital, mensajes, etc.

4.2.2. ¿Cómo funciona?

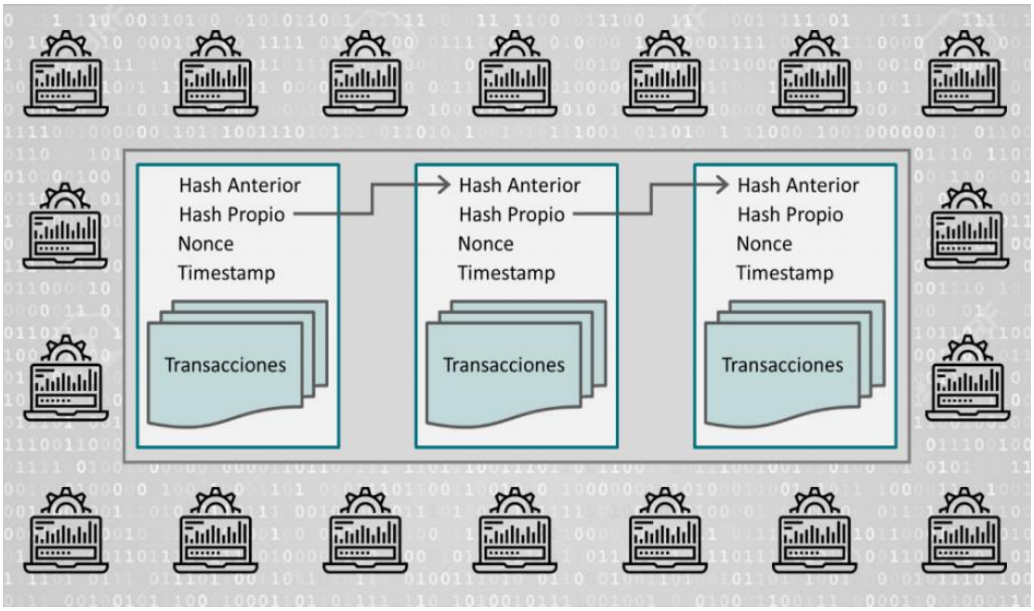


Figura 2: *Es una estructura simple de una Blockchain, cada bloque se conecta con el siguiente y guarda el hash de referencia del anterior.*
Fuente*:* https://www.welivesecurity.com/la-es/2022/05/13/blockchain-que-es-como-funciona-y-como-se-esta-usando-en-el-mercado/*.*

Inicia con una transacción que se registra con un bloque de datos, haciendo referencia a algún activo tangible o intangible. Este bloque de datos registra toda la información definida por un contrato inteligente: por ejemplo, fechas, nombres, contactos, precios, direcciones, etc. Cada bloque está enlazado con el anterior y con el siguiente formando una cadena de bloques que crece a medida que se registran nuevas transacciones. Finalmente, esta cadena de bloques se vuelve irreversible, debido a que cada bloque refuerza la validez del bloque anterior logrando asegurar la inmutabilidad, integridad y disponibilidad de la información [8].

4.2.3. Beneficios

Al implementar una blockchain se consigue lo siguiente [9]:

- Únicamente los usuarios autorizados tendrán acceso a la información.
- La información de las transacciones no podrá ser manipulada por terceros, únicamente su autor o dueño podrá manipularla.
- Toda la información de las transacciones es verídica lo cual evita perder tiempo en comprobarlo, además que los contratos inteligentes aceleran el proceso de registrar las transacciones.

4.2.4. Tipos de redes

Existen diversas redes de blockchain que se pueden utilizar dependiendo del proyecto [8], pero para este Proyecto de Integración Curricular se implementó la red permisionada pública la cual se compone de las siguientes redes:

- **Red de autorización**, utiliza restricciones y reglas para establecer los nodos que tendrán acceso y participación en la blockchain.
- **Red de consenso**, es una red de autorización que mediante protocolos de consenso consigue validar y registrar la información de las transacciones, siendo que todos los nodos participantes comparten la responsabilidad de votar para añadir nuevas transacciones.

4.2.5. Aplicaciones más frecuentes

Los casos más comunes donde se aplica este tipo de tecnología son los siguientes [8]:

- **Para el procesamiento de pagos,** permite agilizar el proceso de transferencia de dinero, y dependiendo de la blockchain se pueden eliminar las tarifas.
- **Para la identidad digital,** otorga a los usuarios una identidad tan válida, creíble, verídica y segura como en el mundo real, además que tendrán el control total de su información, evitando que terceros puedan manipularla o incluso divulgarla.
- **Para guardar información,** almacena información de forma segura, válida y transparente, por ende, ningún usuario podrá manipular información que no le corresponda.

4.3. Aplicación descentralizada (DApp)

Es una aplicación web, móvil o de escritorio que funciona en conjunto con una blockchain, esto reduce el riesgo de manipulación o alteración de la información. Además, que

en la lógica del servidor se utilizan contratos inteligentes [10]. Los usuarios mediante la interfaz pueden acceder a las características de la blockchain. (ver Figura 3).

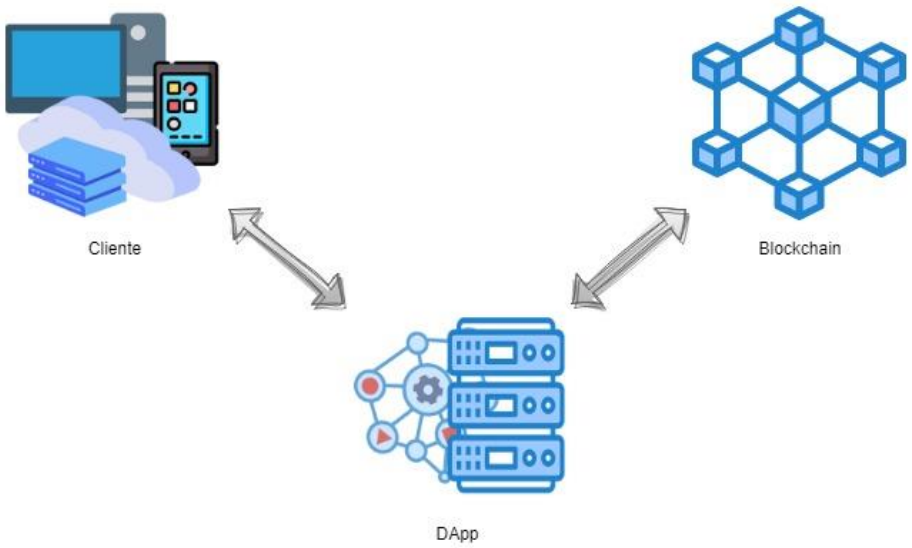


Figura 3: *Estructura simple de una aplicación descentralizada.*
Fuente: *Autor*.

Las DApps permiten la construcción de aplicaciones descentralizadas que aprovechan los beneficios de la blockchain para conseguir una mayor seguridad, transparencia y confiabilidad que las aplicaciones centralizadas. Para este Proyecto de Integración Curricular se tiene como objetivo construir una DApp sobre Identidad Digital.

4.4. Lenguajes de programación

Son un conjunto de instrucciones que sirven para la interacción entre un ser humano y una computadora, las instrucciones se construyen basándose en la sintaxis definida por el lenguaje. Tienen como objetivo principal crear software para ofrecer soluciones a necesidades o problemas de la sociedad. Existen un sinnúmero de lenguajes de programación, siendo los más populares: C++, C#, Visual Basic, Ruby, Go, Python, JavaScript y Java. [11].

4.4.1. JavaScript

Es un lenguaje de programación que permite la construcción de páginas web interactivas, dinámicas y que facilita la actualización de componentes HTML. Este lenguaje ayuda a mejorar la experiencia del usuario en las páginas web, también es utilizado para la creación de aplicaciones de escritorio, e incluso para servidores. Su código es interpretado, es decir, se traduce a código máquina y se ejecuta sobre la marcha [12].

4.4.2. Python

Es un lenguaje de programación interpretado que no necesita de compilación para funcionar, sino que por medio de un intérprete se va ejecutando. Se utiliza principalmente para

el desarrollo de aplicaciones web, para procesamiento de datos, machine learning e inteligencia artificial, entre otros. Sin olvidar que es un lenguaje sencillo de aprender gracias una sintaxis muy similar al lenguaje humano [13].

4.5. Protocolo de consenso

Son protocolos que permiten crear un entorno donde colaboran los nodos que la conforman y que se mantienen seguros. Toda la información de las transacciones de una blockchain debe ser aceptada por los nodos participantes, es decir, que el último bloque generado es el mismo para todos. Cada bloque no contiene información manipulada ni datos corruptos [14].

4.4.1. Bizantino tolerante a fallos (BTF)

Es un algoritmo donde todos los nodos participantes son autorizados, son confiables e intervienen en un proceso de votación para registrar un nuevo bloque, para ello más de dos tercios de todos los nodos deben aceptar el registro. Antes de agregar nuevos bloques, el algoritmo selecciona aleatoriamente los nodos que participarán y ocultará su identidad de los demás nodos para protegerlos de posibles amenazas [15].

BTF puede tolerar el comportamiento malicioso de hasta un tercio de los nodos de la blockchain, permitiendo que las operaciones del sistema no se vean afectadas. Además, el conceso entre los nodos participantes se logra de manera rápida y sin costos económicos. Y, también posee un alto rendimiento, una baja latencia, un uso bajo de los recursos de computación, proporciona seguridad y validez a los bloques que componen la blockchain. La blockchain que ofrece Hyperledger Indy hace uso de este algoritmo de conceso.

4.6. Criptografía

Son las herramientas o métodos utilizados para proteger la información sensible o crítica, lo cual soluciona los problemas relacionados con la autenticidad o confiabilidad de los datos de un sistema. Su implementación permite alcanzar niveles altos de confidencialidad en la información que procesan los sistemas, la información es encriptada y únicamente los usuarios autorizados podrán desencriptarla para tener acceso a ella [16].

4.5.1. Algoritmo chacha20

Este algoritmo necesita una clave de 256 bits y una nonce de 96 bits, utiliza estos datos para generar un flujo de claves del mismo tamaño que los datos a cifrar/descifrar. Después de que se crea la secuencia se combina con los datos mediante la operación XOR

para obtener el resultado. Dado que la operación XOR es reversible, los procesos de cifrado y descifrado son iguales [17].

Este algoritmo trabaja con unidades de 32 bits que se tratan como un elemento único y las operaciones realizadas con ese elemento son un solo elemento. El flujo de claves se compone de bloques de 512 bits, y para cada bloque generado, se modifica un valor inicial para que sean diferentes.

Para inicializar un bloque se necesita lo siguiente:

- Una constante de 128 bits.
- Una clave de 256 bits.
- El contador de bloques de 32 bits incrementable, comenzando en 1.
- Una nonce de 96 bits.

Al reunir estos valores se obtiene el bloque original que tiene que pasar por 20 rondas de encriptación para generar un nuevo bloque. Finalmente, el nuevo bloque se suma al original.

Hyperledger Indy hace uso de este algoritmo criptográfico para mantener segura la información de la blockchain.

4.7. Aplicación de una sola página

Es una aplicación web (Single Page Application - SPA) que está compuesta por un sinnúmero de componentes individuales y dinámicos que dependiendo de la interacción del usuario se pueden eliminar, agregar, reemplazar o actualizar sin requerir la recarga de la página completa [18].

Características principales:

a. **Componentes individuales,** la aplicación está compuesta por componentes individuales que interactúan entre ellos y tienen funcionalidades diferentes.
b. **Actualización de componentes,** cualquier componente podrá ser reemplazado o actualizado sin afectar el funcionamiento de los demás, dependiendo de la acción del usuario con el sistema.
c. **Recarga por componentes,** las páginas nunca se actualizan o se recargan por completo, sino que sus componentes son los únicos que podrán actualizarse con nueva información.
d. **Acciones del usuario,** la aplicación se encarga de controlar todas las interacciones del usuario (clics en botones, ingreso de datos, entre otros.) lo que resulta en una aplicación fluida y dinámica.

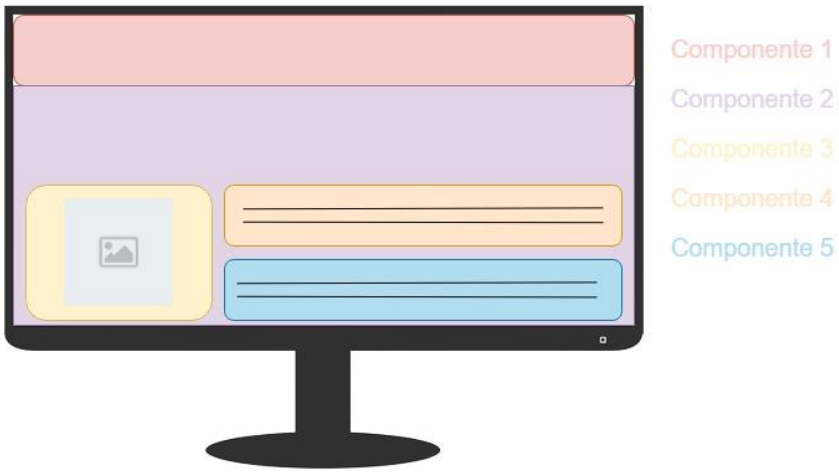


Figura 4: *Modelo básico de una página SPA, los componentes están marcados con diferentes colores.*
Fuente: *Autor.*

Una SPA ofrece un manejo más flexible y dinámico de la información entre el usuario y el sistema. Además, permite la actualización de únicamente los componentes afectados por la interacción del usuario, esto reduce el tiempo de espera para visualizar la nueva información. En conclusión, al desarrollar una SPA se consigue una mejor experiencia de usuario.

4.8. Frameworks

Es un entorno de trabajo que tiene como objetivo facilitar el desarrollo de un producto de software, debido a que posee herramientas y características útiles que agilizan la construcción del sistema. En otras palabras, busca facilitar el tiempo de desarrollo de las aplicaciones, reduciendo la mayor cantidad de errores dentro de su alcance y proporciona un software de mayor calidad. Por lo tanto, un framework es un conjunto de librerías que permiten la construcción de software más eficiente y eficaz [19].

4.4.1. Hyperledger Indy

Es un proyecto bajo el dominio de Hyperledger, el cual está respaldado por la Fundación Linux, siendo una blockchain construida y enfocada para la identidad digital descentralizada, permitiendo crear identidades digitales que estarían registradas en la blockchain. Además, Indy ofrece herramientas, librerías y componentes reutilizables que permiten construir identidades digitales interoperables entre diferentes dominios administrativos y aplicaciones. Sus principales características son [20]:

- Blockchain enfocada especialmente para la identidad digital.
- Estructura de nodos sólida y segura, todos están autorizados.

- Uso de DID (identificadores descentralizados) que son globalmente únicos y confiables, no necesitan de alguna autoridad centralizada.
- Usa el protocolo Zero Knowledge Proof (ZKP), el cual puede verificar datos específicos dentro de un conjunto de datos sin exponerlo.

4.4.1.1. Componentes fundamentales

Los siguientes componentes son fundamentales para entender y conocer cómo funciona Hyperledger Indy [6]:

- **Identificadores descentralizados (DIDs):** es un identificador único que proporciona una identidad digital verificable a un sujeto (persona, empresa, organización, etc.), y que es la firma digital del usuario en la red blockchain. Además, al registrar el DID en la billetera digital otorgará acceso al usuario a la gestión y control total de su información.
- **Esquemas de transcripción:** es un archivo digital compuesto por: el nombre del esquema, los atributos y la versión. Este archivo contiene la información que la entidad "solicitante" crea necesaria para la entidad denominada "destinatario". Las entidades, al hacer uso de algún esquema, seguirán un proceso de transacción en cual definirán y confirmarán credenciales de transcripción, similar a la realización de un contrato donde la información será el activo valioso a intercambiar.
- **Red descentralizada permisionada pública:** es un conjunto de nodos que inicia la red y permite la integración de nuevos nodos si cumplen con los requisitos de autenticidad y de regulación establecidos. Este tipo de red es autosuficiente y se caracterizan por su transparencia y por sus transacciones sin costos de operación.

4.4.1.2. Indy SDK

Indy SDK proporciona un ledger distribuido para la identidad digital auto-gestionable, es decir, una identidad digital portátil y permanente que no depende de ninguna entidad centralizada. Además, proporciona las herramientas necesarias para que los usuarios puedan crear identidades únicas, transparentes y seguras, a través de una librería, desarrollada en el lenguaje de programación C, denominada "libindy" [21].

Libindy: Es el componente más importante de Indy SDK, ya que proporciona el código base para el desarrollo de aplicaciones mediante Hyperledger Indy. Proporciona diversos wrappers (guías de cómo utilizarlo) en diferentes lenguajes de programación. En este caso, para el desarrollo del Proyecto de Integración Curricular se utilizó el wrapper disponible para Python.

Indy hace uso de una blockchain open-source, la cual funciona como una base de datos construida por el pool de nodos participantes. La información está protegida criptográficamente y es almacenada por todos los nodos participantes.

4.4.2. Nodejs

Es un entorno de ejecución en tiempo real, de código abierto y multiplataforma, lo que permite a los desarrolladores crear toda tipo de herramientas y aplicaciones utilizando el lenguaje de programación JavaScript. Su característica de ejecución en tiempo real está enfocada en utilizarse directamente en una computadora, o servidor. Además, tiene una gran comunidad activa que colabora con nuevas librerías enfocadas a las diversas áreas de la informática [22].

4.4.3. Vuejs

Es un framework progresivo enfocado a la construcción de interfaces de usuario y diseñado para ser incremental y escalable. La librería central está enfocada en la capa de visualización, y, además, es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, también es perfecto para construir sofisticadas Single-Page Applications cuando se utiliza en combinación con herramientas modernas y librerías de apoyo [23].

4.4.4. Expressjs

Es un framework de código abierto, gratuito, simple, ágil, escalable y eficaz que posee herramientas enfocadas al enrutamiento, procesamiento de las peticiones y respuestas HTTP, construcción de middlewares, entre otros. Sus principales usos son para construir endpoints, aplicaciones en tiempo real, aplicaciones de streaming, etc, [24].

4.4.5. Flask

Es un framework construido con Python para facilitar el desarrollo de Aplicaciones Web mediante el patrón MVC, lo que permite construir y controlar mediante un controlador las peticiones y respuestas HTTP, permite realizar pruebas en un entorno de servidor para ir comprobando y validando lo resultados, posee una gran variedad de librerías y extensiones que lo vuelve apta para el objetivo a desarrollar [25].

4.9. Modelo de identidad digital auto-gestionada

El modelo de identidad digital auto-gestionada propone tres elementos necesarios para que el usuario obtenga el control total de la información que compone su identidad digital, estos son: control individual, seguridad y portabilidad. Se cambia el sistema centralizado de información (bases de datos) por el sistema descentralizado de información (blockchain), esto

permite al usuario poseer, controlar y administrar completamente su identidad. Por lo tanto, las entidades externas no podrán manipular, proporcionar o quitar la identidad del usuario.

El usuario bajo este modelo podrá compartir información sobre su identidad, ya sea por partes o en su totalidad según lo disponga. Además, con su consentimiento podrá compartir con otros usuarios su información, según el evento de transacción que se realice. Las transacciones de información únicamente ocurrirán entre las partes involucradas sin intervención de terceros [26].

En la Figura 5 se muestra el modelo de identidad digital auto-gestionada acoplada a las características de la blockchain de Hyperledger Indy.

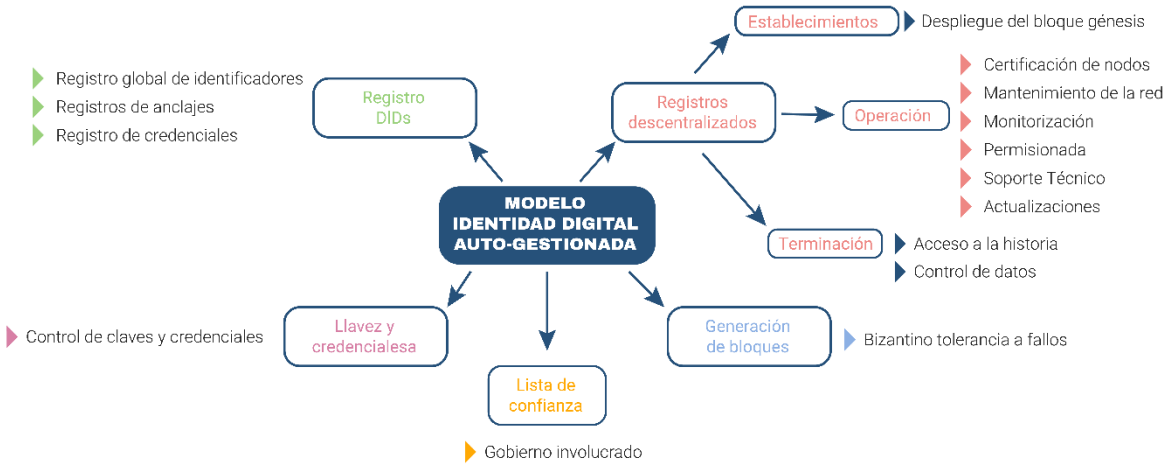


Figura 5: *Modelo de identidad digital auto-gestionada.*
Fuente: *Autor.*

El modelo de identidad digital auto-gestionada posee los siguientes componentes [6]:

- **Registros descentralizados:** la importancia de los registros descentralizados se divide en tres fases con sus respectivas tareas:
  - **Establecimiento**
    - **Despliegue del bloque génesis:** es el primer bloque de la red, el cual contiene los nodos validadores y observadores, y el protocolo de consenso que utilizarán los nodos para generar nuevos bloques.
  - **Operación**
    - **Certificación de nodos:** los nodos participantes (validadores y observadores) deben ser confiables y definidos idóneamente según las características de la blockchain.
    - **Mantenimiento de la red:** el mantenimiento es fundamental para garantizar que la red funcione sin problemas, evitando fallos o colapsos de los nodos.

- **Permisionada:** únicamente los nodos y cuentas de usuario autorizadas podrán interactuar con la blokchain.
- **Soporte técnico:** la asistencia técnica es necesaria cuando la implementación falla, el rendimiento de los nodos no es óptimo o si la aplacación presenta errores.
- **Actualizaciones:** la investigación de nuevas tecnologías, medidas, estándares, entre otros, ayudarán a mejor la red descentralizada en seguridad, escalabilidad, rendimiento, interoperabilidad y eficiencia.

➢ **Terminación**

- **Acceso a la historia:** los usuarios tendrán acceso al historial de sus transacciones realizadas.
- **Control de datos:** el usuario tiene el control total de su información, por lo tanto, determinará si sus datos pueden ser transferidos, eliminados, modificados, expuestos o si crea nuevos.

- **Generación de bloques:** para la generación de nuevos bloques en una red blockchain se hace uso de los protocolos de consenso, el gobierno de la red es independiente a este proceso, dónde únicamente participan los nodos validadores.
  - **Bizantino tolerante a fallos:** los nodos validadores de la red se turnan para generar nuevos bloques, el algoritmo utilizado minimiza que algún nodo corrupto manipule los bloques.
- **Listas de confianza:** se determina la participación del gobierno establecido en la red para las interacciones de los usuarios.
  - **Gobierno involucrado:** el gobierno es la autoridad certificadora principal, todos los usuarios que se registren bajo su dominio tendrán acceso a las transacciones en la red. También podrá designar que usuarios pueden crear, actualizar o eliminar contratos inteligentes.
- **Llaves y credenciales:** la identidad digital del usuario es todo lo que contiene su billetera digital, la cual sirve para administrar, almacenar y presentar ya sea credenciales, transacciones o información personal. Por lo tanto, el usuario es el único que podrá controlar su identidad digital y, además, determinará con quién comparte su información.
- **Registros DIDs:** son los identificadores descentralizados asignados a cada entidad (usuarios, organizaciones, empresas, etc) que vaya a interactuar con la red blockchain, otorgando una identidad digital verificable.
  - **Registro global de identificadores:** los identificadores al ser claves para la identidad digital de los usuarios, estos deberán ser almacenados en la misma red descentralizada, o en alguna otra red centralizada de requerirlo.
  - **Registro de anclajes:** al crear la identidad digital del usuario, se determinará si es idóneo para utilizar o administrar los contratos inteligentes de la red blockchain.

- **Registro de credenciales:** el usuario al momento de utilizar algún contrato inteligente se registrará su identificador, esto permitirá al usuario tener control sobre el contrato hasta que termine de usarlo.

4.10. Metodología de desarrollo

4.5.1. ABCDE (agile block chain DApp engineering)

La metodología ABCDE, tiene en cuenta la diferencia entre el desarrollo de software tradicional (sistema de aplicaciones) y el desarrollo de contratos inteligentes (chaincode), y los separa en dos subsistemas. Para cada uno, se utiliza un enfoque ágil debido a que en las DApps el desarrollo de los requisitos puede ir cambiando según se avanza en el proyecto [27].

En la Figura 6 se muestran los pasos esta metodología donde la mayoría se realizan varias veces, debido a su enfoque iterativo e incremental. Los círculos rosas representan las reuniones de planificación que se realizan al inicio de cada iteración (SPM) y las reuniones de revisión (SRM) que se realizan al final de cada iteración. Cada reunión es esencial para conocer lo que se va a desarrollar durante la iteración, y también para conocer sobre los errores o fallos que se tuvieron con el producto desarrollado al final de la iteración.

La metodología ABCDE tiene las siguientes fases para el desarrollo de una DApp [27]:

1. **Definir el objetivo del sistema:** Es un resumen de 10 a 30 palabras que definen el objetivo de la DApp.
2. **Identificar a los actores:** Se deben identificar los actores que van a interactuar con la DApp, pudiendo ser roles humanos, de sistemas o incluso dispositivos externos.
3. **Realizar las historias de usuario:** Los requisitos de la DApp se deben realizar como historias de usuario para aplicar el enfoque ágil al proyecto. Además, es recomendable para indicar gráficamente cuales son las relaciones entre los actores y los requisitos del sistema, pudiendo ser un Diagrama de Casos de Usos UML.
4. **Dividir el sistema en dos subsistemas:** La DApp se tiene que dividir en los siguientes subsistemas:
   - Los contratos inteligentes que utilizarán en la blockchain (pasos 5-6).
   - El sistema de aplicaciones que es el sistema externo que va a permitir la interacción entre los usuarios y la blockchain, dicha interacción será mediante las transacciones y los contratos inteligentes (pasos 7-8).

   Además, se debe diseñar la arquitectura al completo de la DApp, donde se deben resaltar los datos, métodos o tecnologías que se utilizarán en el subsistema de chaincode y en el subsistema de aplicación.

5. **Diseño de los contratos inteligentes**: Se basa en la elaboración del diseño de los contratos inteligentes, la definición de la estructura de datos que tendrá y los eventos que provocará si es utilizado.
6. **Codificación y prueba del sistema de contratos inteligentes:** Una vez diseñado el contrato inteligente, se procede a codificar y probar, en este caso de utiliza el lenguaje de programación Python. Las actividades a realizar son las siguientes:
    a. Se codifica el contrato inteligente, en base a la definición realizada en su diseño.
    b. Se elaboran las pruebas unitarias para comprobar que los contratos inteligentes satisfacen su diseño, y los requisitos en caso de haber sido incluidos. En caso de que las pruebas den resultados negativos, es necesario volver a la etapa de diseño.
7. **Diseño del subsistema de interacción externa (Sistema de aplicaciones):** Este paso se basa en el diseño del sistema de aplicaciones, el cual se encargará de interactuar con los usuarios, enviará peticiones a la blockchain y además podrá gestionar su propia base de datos o documentación en caso de necesitarlo, se debe incluir el nivel más alto de seguridad debido a que este subsistema es más propenso a recibir ataques de hackers. También se debe definir cómo será el acceso a la blockchain, la interfaz responsiva del usuario, los componentes o módulos que implementará, las medidas de seguridad, etc.
8. **Codificación y prueba del sistema de aplicaciones:** Tanto el sistema de aplicaciones como el sistema de contratos se codifican y prueban al mismo tiempo, en este caso se utiliza el lenguaje de programación JavaScript. Se recomienda que cada dos o tres iteraciones se realice la integración de los resultados de los dos subsistemas. Las actividades a realizar son las siguientes:
    a. Se codifica en base a los requisitos para el sistema de aplicaciones.
    b. Se evalúa la seguridad del código.
    c. Se escriben pruebas unitarias (Pus) para comprobar que los requisitos codificados satisfaces y cumplen con los requisitos establecidos. En caso de que las pruebas den resultados negativos, es necesario volver a la etapa de diseño.
9. **Integrar, probar y desplegar el Sistema DApp:** Para integrar los sistemas de los Contratos Inteligentes y el Sistema de Aplicaciones, la DApp construida hasta ese momento debe ser desplegada ya sea una red local o en una red de pruebas, a continuación, se deben ejecutar las pruebas de integración para comprobar y validar que todos los subsistemas funcionen correctamente en conjunto. En caso de que las pruebas de integración den resultados negativos, se realiza una revisión en busca de errores y se vuelve a la etapa de diseño de cada subsistema.

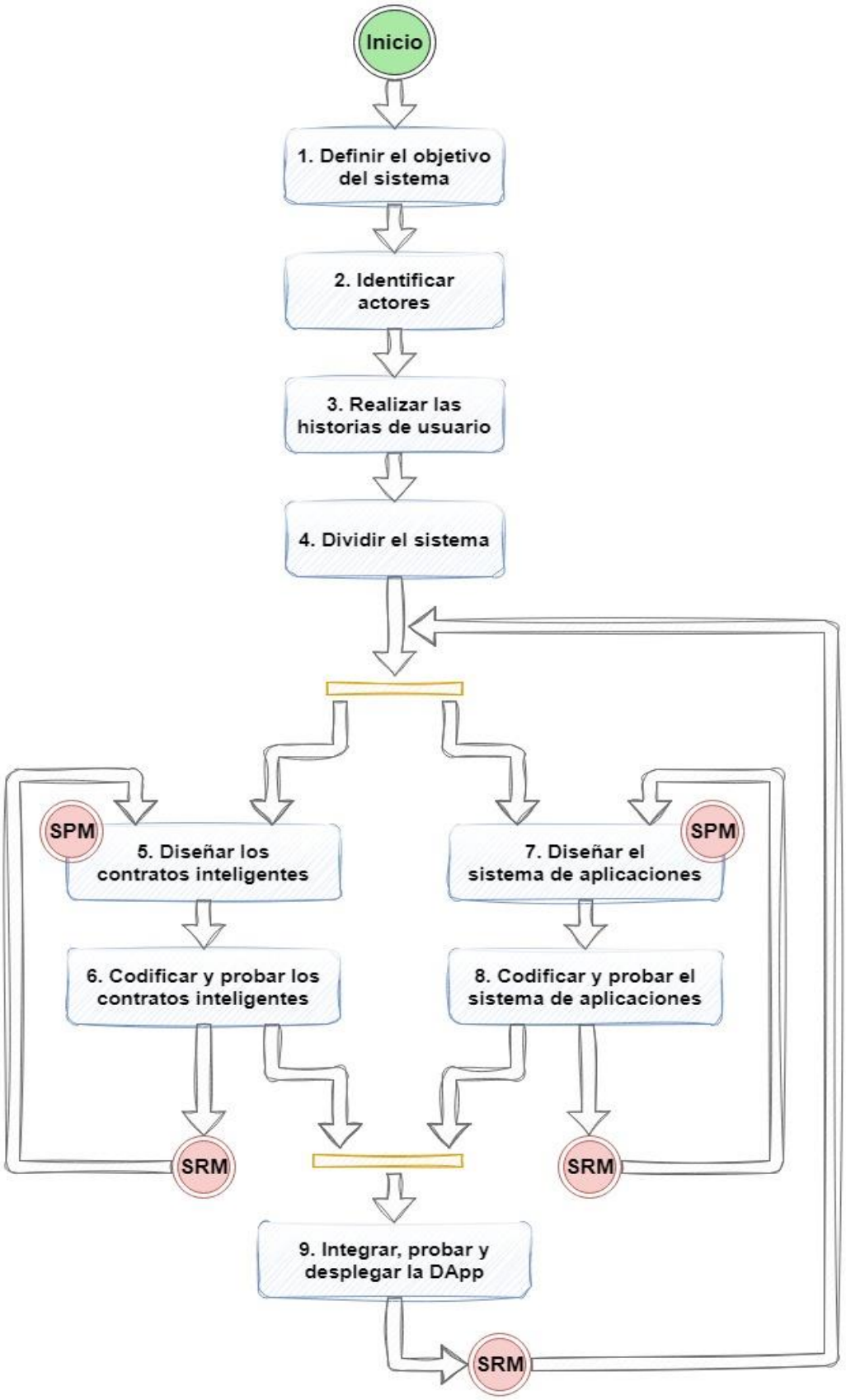


Figura 6: *Fases de la metodología de desarrollo ABCDE.*
Fuente: *Autor.*

4.11. Trabajos relacionados

Según la bibliografía revisada, en Ecuador no existe un proyecto o solución informática relacionada con la identidad digital académica auto-gestionada mediante tecnología blockchain. A continuación, en la Tabla 2 se muestran los trabajos relacionados a esta temática:

Tabla 2: Trabajos relacionados.

Título	Resumen	Ref.
"Identidad digital basada en blockchain en instituciones educativas"	Este proyecto realiza una simulación de identidad digital basada en blockchain utilizando Hyperledger Fabric e Indy para instituciones educativas, la cual otorga a los miembros de la Universidad de los Andas identidad digital con la que podrán ingresar al campus y a las plataformas virtuales.	[21]
"Propuesta de identidad digital para historial clínico unificado utilizando tecnología blockchain"	Este proyecto realiza una propuesta de identidad digital basada en blockchain utilizando Hyperledger Fabric aplicada a los historiales clínicos, la cual otorga a los pacientes identidad digital permitiéndoles decidir quien tiene acceso a su historial clínico.	[28]
"Diseño e implementación de un sistema de identidad digital descentralizada para ciudadanos de la Unión Europea en el ámbito sanitario"	Este proyecto implementa identidad digital basada en blockchain utilizando Hyperledger Indy aplicada a la sanidad de los ciudadanos, la cual otorga a los ciudadanos identidad digital permitiéndoles presentar certificados de salud verídicos.	[29]
"Implementación de la tecnología Blockchain para la validación de autenticidad de los certificados académicos digitales"	Este proyecto implementa tecnología blockchain de Ethereum para la validación de autenticidad de certificados académicos.	[30]

5. Metodología

En esta sección se exponen el área de estudio, procedimiento y recursos que están involucrados en el desarrollo del Proyecto de Integración Curricular (PIC).

5.1. Área de estudio

El Proyecto de Integración Curricular se desarrolló en la Universidad Nacional de Loja, Ecuador, en la Facultad de Energía, Carrera de Ingeniería en Computación; las pruebas se llevaron a cabo con los estudiantes de la misma carrera.

Figura 7: *Área de desarrollo del PIC.*

5.2. Procedimiento

Para la elaboración del presente Proyecto de Integración Curricular, se utilizó la metodología de desarrollo agile block chain DApp engineering (ABCDE), la cual es la más apta de acuerdo a las características que contiene sobre la integración de una aplicación tradicional con una aplicación descentralizada. Para conocer con más detalle sobre esta metodología ir al **punto 4.10 de la sección Marco teórico**.

Para cumplir con el objetivo general se completaron las siguientes actividades de acuerdo a cada objetivo:

1. **Definir el sistema para la identidad digital académica auto-gestionada usando la Ingeniería de Requisitos.**

   a. Análisis del tema propuesto, investigación y comprensión sobre la identidad digital y cómo la blockchain puede ayudar a asegurar la identidad de sus usuarios mediante su red descentralizada y protocolos de consenso, dando como resultado el **Marco teórico** con la información más relevante para comprender este proyecto.

b. Definición del objetivo sistema que es equivalente al objetivo del PIC, correspondiente a la fase 1 de la metodología de desarrollo de software ABCDE (ver el **punto 6.1.1 de la sección Resultados**).

c. Determinación de los actores que intervienen en el sistema, correspondiente a la fase 2 de la metodología de desarrollo de software ABCDE (ver el **punto 6.1.2 de la sección Resultados**), se utilizó la especificación de requisitos según el estándar de IEEE 830.

d. Proposición, especificación y validación de los requerimientos del sistema, correspondiente a la fase 3 de la metodología de desarrollo de software ABCDE (ver el **punto 6.1.3 de la sección Resultados**), se utilizó el documento de especificación de requisitos de la IEEE 830.

e. División del sistema, además de determinar las tecnologías necesarias para continuar con el siguiente objetivo, correspondiente a la fase 4 de la metodología de desarrollo de software ABCDE (ver el **punto 6.1.4 de la sección Resultados**), se diseñó la arquitectura de la DApp y explicó la funcionalidad de cada módulo.

2. Desarrollar el sistema para la identidad digital académica auto-gestionada mediante tecnología Blockchain.

a. Diseño del subsistema de contratos inteligentes, correspondiente a las fases 5 de la metodología de desarrollo ABCDE (ver el **punto 6.2.1 de la sección Resultados**), se diseñó el diagrama de clases del módulo “Back-end” y también el proceso de las transacciones.

b. Codificación y pruebas unitarias del subsistema de contratos inteligentes, correspondiente a la fase 6 de la metodología de desarrollo ABCDE (ver el **punto 6.2.2 de la sección Resultados**), se utilizó Python para la codificación y el documento de plan de pruebas unitarias para las validaciones.

c. Diseño del subsistema de aplicación, correspondiente a las fases 7 de la metodología de desarrollo ABCDE (ver el **punto 6.2.3 de la sección Resultados**), se utilizó el patrón de diseño MVC y se diseñó el diagrama de tablas para la base de datos del módulo “Middleware”.

d. Codificación y pruebas unitarias del subsistema de aplicación, correspondiente a la fase 8 de la metodología de desarrollo ABCDE (ver el **punto 6.2.4 de la sección Resultados**), se utilizó JavaScript, Nodejs, Expressjs y Vuejs para la codificación y el documento de pruebas unitarias para las validaciones.

3. **Probar el sistema de identidad digital académica auto-gestionada en un ambiente controlado.**

   a. Elaboración, validación y ejecución de las Pruebas de Integración, correspondiente a la fase 9 de la metodología de desarrollo ABCDE (ver el **punto 6.3.1 de la sección Resultados**), se utilizó el documento de plan de pruebas de integración para validar el correcto funcionamiento de la DApp.

   b. Elaboración, validación y ejecución de las Pruebas de Funcionales y las Pruebas de Aceptación, correspondiente a la fase 9 de la metodología de desarrollo ABCDE (ver el **punto 6.3.2 de la sección Resultados**), se utilizó el documento de plan de pruebas de funcionales para validar el cumplimiento de los requisitos especificados y se aplicó una encuesta a los estudiantes para conocer el nivel de aceptación de la DApp.

   c. Despliegue del sistema, correspondiente a la fase 9 de la metodología de desarrollo ABCDE (ver el **punto 6.3.3 de la sección Resultados**), se utilizó nginx y pm2 para desplegar la DApp mediante Docker.

5.3. Recursos

5.3.1. Métodos

a. Analítico

Es un método que mediante un procedimiento se divide la complejidad de un todo en partes más sencillas [31], se utilizó para dividir el Proyecto de Integración Curricular en partes más sencillas para reducir su complejidad, dando como resultado los objetivos específicos junto a sus actividades.

b. Investigación-Acción:

Es un método enfocado al aprendizaje mientras se está realizando alguna actividad o tarea. Primero se identifica el problema, a continuación, se realizan algunas acciones para resolverlo, luego, se comprueba la eficacia de las acciones aplicadas y por último se comprueba los resultados, sino son adecuados se vuelve a repetir el proceso [32]. Fue aplicado en las diversas fases de la metodología del Proyecto de Integración Curricular.

c. Replicación:

Es un método que permite desarrollar varias veces el mismo producto hasta lograr lo deseado, siendo aplicado al desarrollo del sistema para satisfacer y cumplir con los requisitos funcionales especificados.

d. Análisis estático:

Es un método dedico a la revisión minuciosa hacia la estructura del producto, para este proyecto se enfocó a la validación del sistema con el objetivo de descubrir y corregir posibles errores.

5.3.2. Técnicas

a. Encuestas

Es una técnica utilizada para obtener información sobre algún tema aplicada a una muestra de personas [33]. Para este proyecto se aplicaron encuestas para conocer el nivel de aceptación del sistema desarrollado.

b. Reuniones

Es una técnica que permite la revisión de algún producto entre dos o más personas interesadas dando como resultado en posibles mejoras o aceptación. En este caso se utilizó para revisar los avances del Proyecto de Integración Curricular.

5.3.3. Estándares

a. IEEE 830

Es un documento recomendado para la especificación de requerimientos o requisititos tanto funcionales como no funcionales. Este estándar sirvió para cumplir con el primer objetivo específico de este proyecto.

6. Resultados

En esta sección se detallan los resultados del desarrollo del Proyecto de Integración Curricular, completado de acuerdo a los objetivos específicos planteado siguiendo metodología de desarrollo ABCDE.

6.1. Objetivo 1: Definir el sistema para la identidad digital académica auto-gestionada usando la Ingeniería de Requisitos.

Para completar este objetivo, se cumplieron las fases de acuerdo a la metodología de desarrollo ABCDE para el Proyecto de Integración Curricular, cada fase fue adaptada al proyecto.

6.1.1. Fase 1: Objetivo del sistema.

A partir del objetivo general del Proyecto de Integración Curricular se establece como objetivo del sistema "**Desarrollar un sistema de identidad digital mediante tecnología Blockchain para la Universidad Nacional de Loja**".

6.1.2. Fase 2: Actores del sistema.

El sistema es operado por los siguientes actores:

- **Usuario final:** es el usuario (estudiante, docente o personal administrativo) que utilizará el sistema, sus principales funcionalidades son de administrar su propia información y cursos que haya participado en la **página de la Carrera de Computación**, y podrá utilizar los Esquemas de Transcripción para solicitar información a otros usuarios.
- **Usuario creador:** es el usuario con las mismas funcionalidades del usuario final, además de poder administrar los Esquemas de Transcripción.
- **Usuario administrador:** es el usuario con las mismas funcionalidades del usuario creador, además de poder asignar roles o desactivar cuentas de los demás usuarios.

6.1.3. Fase 3: Requerimientos del sistema

Esta fase presenta de manera resumida los siguientes componentes: requisitos funcionales, no funcionales y diagrama de casos de uso (obtenidos del **Anexo 1: Especificación de requisitos de software**) siguiendo el estándar IEEE 830.

a. Requisitos funcionales

En la Tabla 3, se muestran los requisitos funcionales del sistema para la identidad digital académica auto-gestionada.

Tabla 3: *Requisitos funcionales del sistema.*

ID	Nombre	Descripción	Prioridad
RF 01	Autenticar	El sistema permitirá autenticarse al usuario	Alta
RF 02	Iniciar sesión	El sistema permitirá al usuario iniciar sesión	Alta
RF 03	Registrar	El sistema permitirá al usuario registrarse	Alta
RF 04	Resetear contraseña	El sistema permitirá al usuario resetear su contraseña	Alta
RF 05	Administrar cursos	El sistema permitirá al usuario administrar sus cursos	Alta
RF 06	Administrar información	El usuario podrá solicitar información de otro usuario mediante esquemas de transcripción definidos en la Blockchain.	Alta
RF 07	Utilizar Esquemas de Transcripción	El sistema permitirá al usuario utilizar Esquemas de Transcripción	Alta
RF 08	Buscar perfil de usuarios	El sistema permitirá al usuario buscar información de otros usuarios, siendo expuesta si tienen dicho perfil de información público	Alta
RF 09	Administrar perfil de información	El sistema permitirá al usuario administrador su perfil de información	Alta
RF 10	Administrar Esquemas de Transcripción	El sistema permitirá al usuario creador administrar Esquemas de Transcripción	Alta
RF 11	Administrar usuarios	El sistema permitirá al rol administrador administrar a los demás usuarios	Alta

b. Requisitos no funcionales

En la Tabla 4, se presentan los requisitos no funcionales del sistema para la identidad digital académica auto-gestionada.

Tabla 4: *Requisitos no funcionales del sistema.*

ID	Nombre	Descripción	Prioridad
RNF 01	Usabilidad	El sistema tendrá una interfaz amigable, responsiva y sencilla para lograr una interacción amena con los usuarios.	Alta
RNF 02	Seguridad	El sistema utilizará tokens para el envío de información y la información sensible será almacenada en la wallet de cada usuario, dentro de la Blockchain, para mantener su confidencialidad e integridad.	Alta
RNF 03	Fiabilidad	La información de los usuarios será almacenada en la Blockchain, por tal motivo la información no podrá ser manipulada por usuarios ajenos a ella.	Alta
RNF 04	Tiempo de respuesta	El sistema responderá a las solicitudes de los usuarios en un tiempo adecuado y eficiente, de 3 a 10 segundos máximo, siempre y cuando la conexión a internet sea estable (banda ancha de 10 Mbps). Este tiempo se prolonga mayormente por los servicios de la Blockchain, en caso de que la petición no ocupe dichos servicios se responderá en un tiempo mínimo.	Alta
RNF 05	Disponibilidad	El sistema funcionará en lo posible durante las 24 horas del día, durante los 7 días de la semana, siempre y cuando la infraestructura de los servidores y la red de Blockchain esté en condiciones óptimas. Para las actualizaciones del sistema, no habrá disponibilidad hasta que sean completadas.	Alta
RNF 06	Trazabilidad	El sistema guardará la información pertinente de la utilización de los Esquemas de Transcripción en la base de datos y Blockchain, quedando como constancia de su uso.	Alta
RNF 07	Confidencialidad	El sistema al ser construido en base a la Blockchain de Hyperledger Indy, toda la información de los usuarios se mantendrá encriptada (algoritmo ChaCha20-Poly1305 con HMAC-256) y no podrá ser modificada si no es el mismo usuario.	Alta

c. Diagrama de casos de usos

En la Figura 8, se muestran los diferentes actores con sus respectivos casos de usos.

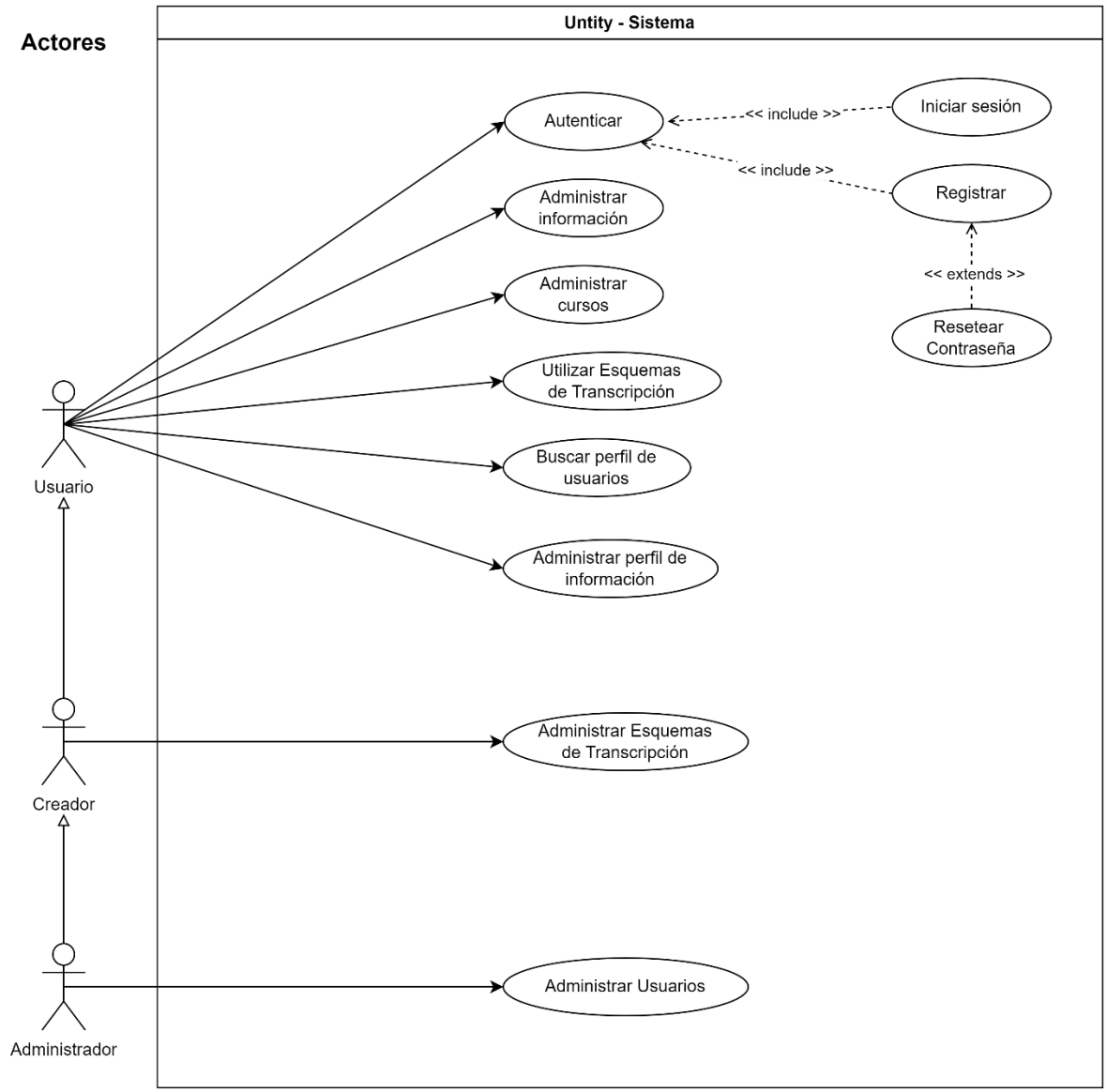


Figura 8: *Diagrama de casos de uso del sistema.*
Fuente: *Autor.*

6.1.4. Fase 4: Dividir el sistema.

El sistema se segmentó en los siguientes subsistemas:

- Subsistema de los contratos inteligentes que se ejecutan en la Blockchain, comprendido por el módulo “Back-end” (las fases que intervienen en este punto son 5 y 6).
- Subsistema de aplicación que interactúa con el usuario final, comprendidos por los módulos “Front-end” y “Middleware” (las fases que intervienen en este punto son 7 y 8).

Con base en los dos puntos anteriores se procede a determinar la arquitectura del sistema, la cual se presenta en la Figura 9.

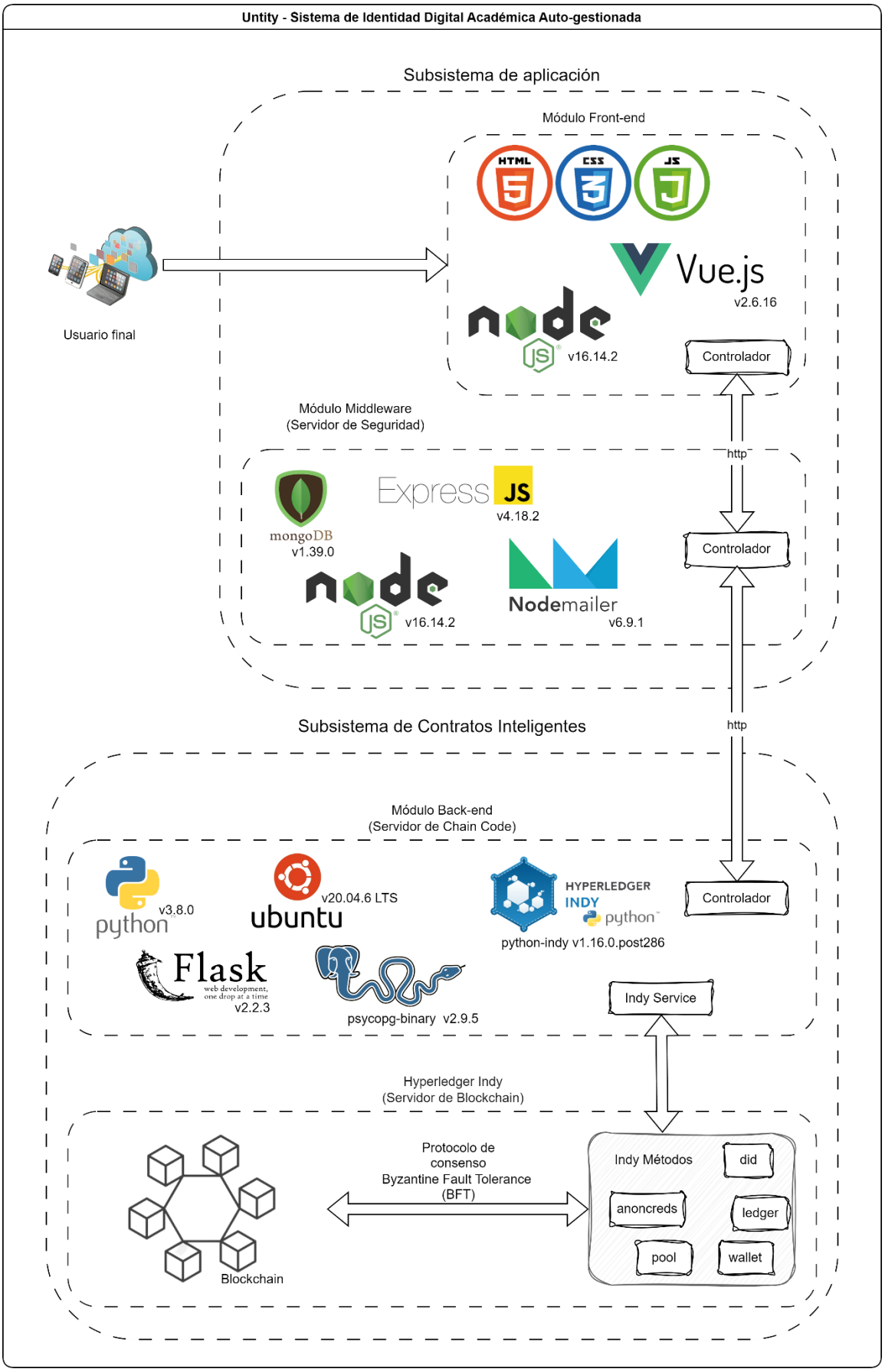


Figura 9: *Arquitectura del sistema.*
Fuente*: Autor.*

A continuación, se describe las principales funcionalidades de cada módulo:

- **Módulo "Front-end":** este módulo se encarga de presentar una interfaz amigable, sencilla y responsiva. El usuario, dependiendo de su rol, podrá: registrarse e iniciar sesión, gestionar su propia información, solicitar información de otros usuarios, ver perfiles de otros usuarios, conseguir certificados por haber completado los cursos de la Carrera, asignar roles, desactivar cuentas, entre otras funciones.

  La finalidad de este módulo es enviar información hacia el Middleware para interactuar con el sistema, además de presentar al usuario información adecuada y mensajes según la interacción por realizar o realizada.

  La comunicación con el Middleware es altamente segura para evitar cualquier tipo de amenaza, se utilizó Json Web Tokens (JWT). JWT permite encriptar con una clave la información a enviar, logrando tener una comunicación entre Frontend y Middleware más segura y confidencial.

  En la Tabla 5, se presentan las principales herramientas tecnológicas y librerías utilizadas para el desarrollo de este módulo.

Tabla 5: *Herramientas tecnológicas del módulo "Front-end".*

Categoría	Nombre	Versión
Tecnología	Vue js	2.6.14
Tecnología	Node js	16.14.2
Librería de estilos	Vuetify	2.6.13
Librería de routing	Vue-router	3.5.1

- **Módulo "Middleware":** este módulo es el puente entre "Front-end" y "Back-end", su propósito es de mitigar o evitar cualquier tipo de amenaza hacia la información almacenada en la red de Blockchain (Back-end). Además, se encarga de validar a los usuarios que quieran acceder a su wallet.

  La comunicación con el Front-end es altamente segura para evitar cualquier tipo de amenaza hacia la confidencialidad de los usuarios, se utilizó Json Web Tokens (JWT). En cambio, la comunicación con el Back-end es más simple porque únicamente este módulo podrá enviar y recibir información de la red Blockchain.

  En la Tabla 6, se presentan las principales herramientas tecnológicas y librerías utilizadas para el desarrollo de este módulo.

Tabla 6: *Herramientas tecnológicas del módulo "Middleware".*

Categoría	Nombre	Versión
Tecnología	Express js	4.18.2
Tecnología	Node js	16.14.2
Librería de seguridad	jsonwebtoken	9.0.0
Librería de email	nodemailer	6.9.1
Librería de conexión a base de datos	mongoose	6.6.5

- **Módulo "Back-end":** este módulo es la red Blockchain, su objetivo es almacenar información de los usuarios en una red descentralizada utilizando 4 nodos. Cada usuario podrá almacenar, eliminar o hacer visible su información personal, además de poder conseguir información de otros usuarios por medio de transacciones gracias a los Esquemas de Transcripción y podrá acceder a certificados por haber culminado los cursos disponibles de la Carrera.

  En caso de llegar a fallar algún nodo, el sistema no dejará de funcionar porque los demás nodos siguen en funcionamiento permitiendo asegurar una disponibilidad perfecta.

  En caso de sufrir algún ataque hacia la integridad de la información almacenada en la red Blockchain, todos los nodos tendrán que ser atacados al mismo tiempo, caso contrario la información no podrá ser alterada. Esto garantiza que toda información guardada en la red no podrá ser modificada por otro usuario que no sea su propietario.

  Toda la información almacenada en la red Blockchain está encriptada, lo cual asegura la confidencialidad en caso de ser expuesta.

  En la Tabla 7, se presentan las principales herramientas tecnológicas y librerías utilizadas para el desarrollo de este módulo.

Tabla 7: *Herramientas tecnológicas del módulo "Back-end".*

Categoría	Nombre	Versión
Tecnología	Ubuntu	20.04.6 LTS
Tecnología	Flask	2.2.3
Lenguaje programación	Python	3.8
Librería de conexión con la blockchain	python-indy	1.16.0. post286
Librería de conexión a base de datos	psycopg-binary	2.9.5

6.2. Objetivo 2: Desarrollar el sistema para la identidad digital académica auto-gestionada mediante tecnología Blockchain.

6.2.1. Fase 5: Diseño de los contratos inteligentes:

En esta fase se diseñó un diagrama de clases para comprender el funcionamiento de la estructura del back-end con sus respectivos atributos, operaciones y las relaciones más importantes, esto se observa en la Figura 10 (Para una mejor visualización ir al **Anexo 2: Diagrama de Clases del módulo "Back-end"**.)

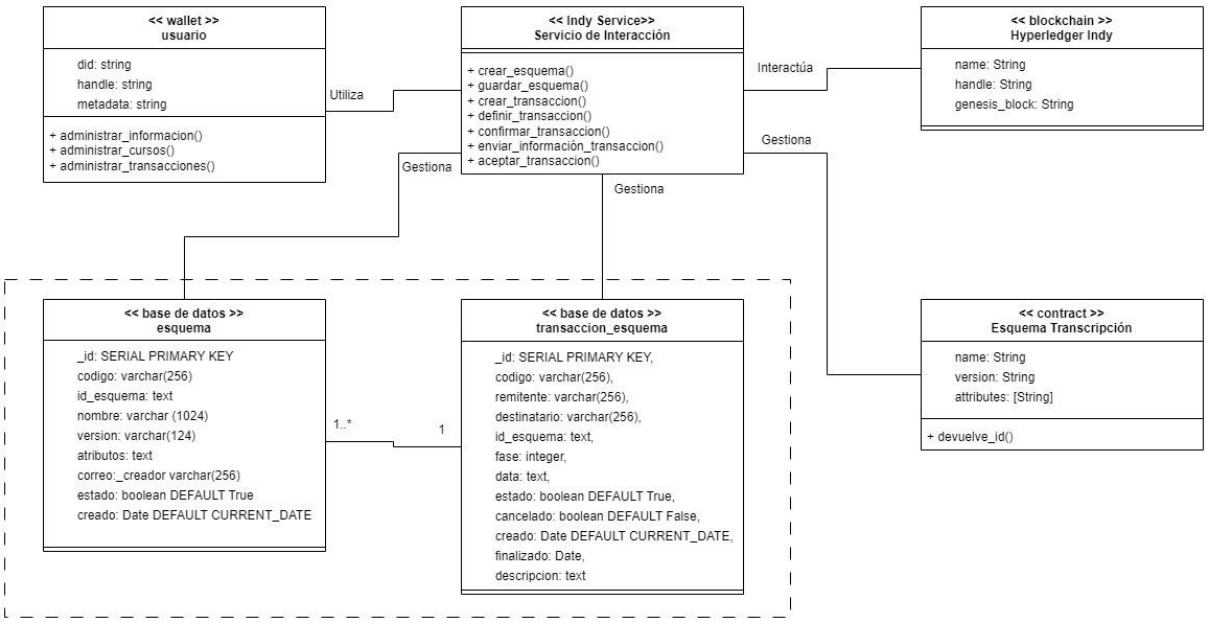


Figura 10: *Diagrama de Clases del módulo "Back-end".*
Fuente: *Autor*

Los Esquemas de Transcripción son los contratos inteligentes que permiten el intercambio de información entre los usuarios de la blockchain, construidos en base a un nombre, una versión y los atributos que son la información que deberá ser completada durante el proceso de transacción.

El proceso de transacción está compuesto por 6 fases, los participantes serán el remitente siendo el usuario interesado en adquirir la información, mientras que el destinatario es el usuario objetivo quien proporcionará la información, y la transacción que es la blockchain que dará seguridad, inmutabilidad y transparencia a la información manejada durante este proceso, todo esto se observa en la Figura 11.

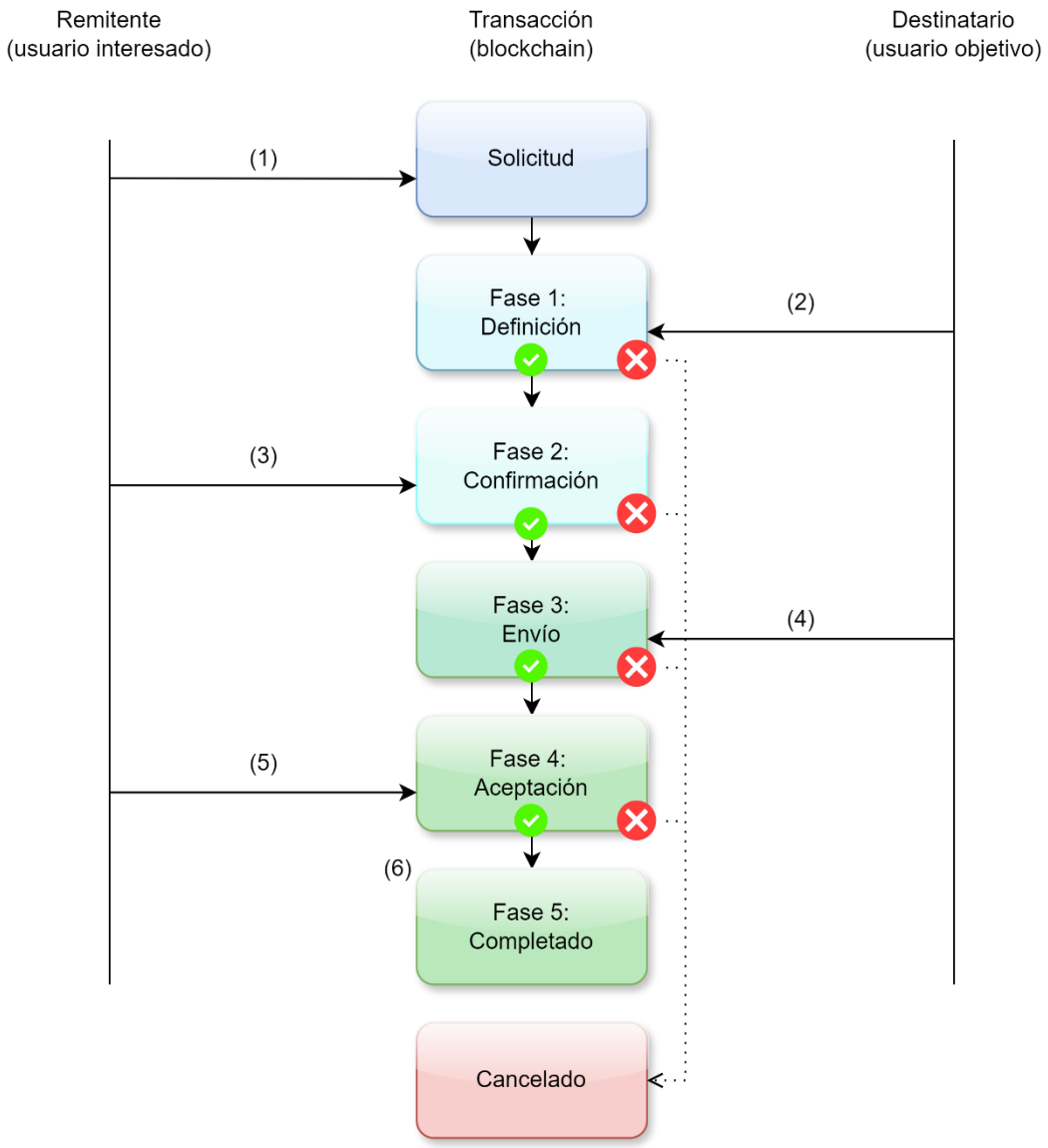


Figura 11: *Fases del proceso de transacción.*
Fuente: *Autor*

El **usuario interesado** escogerá el esquema de transcripción a utilizar y seleccionará al **usuario objetivo** para que proporcione la información correspondiente a los atributos de dicho esquema.

Cada **usuario** podrá cancelar la **transacción** si tiene permitida la acción, en caso de ser cancelada la transacción quedará cerrado el proceso y la información identificativa almacenada en la base de datos será eliminada, quedando únicamente los usuarios pertenecientes al proceso de transacción como constancia de haber iniciado, pero no completado la misma.

Cada fase del proceso de transacción cumple con las siguientes responsabilidades:

- **Solicitud:** es el inicio del proceso de transacción a nivel de base de datos, tiene como responsabilidad enviar al usuario objetivo el esquema de transcripción, con sus atributos, y tiempo de validez que tendrá la información que enviará al usuario interesado.
- **Fase 1, definición:** es el inicio del proceso de transacción a nivel de blockchain, el usuario objetivo, al aceptar la solicitud del usuario interesado, definirá las credenciales de transcripción para la transacción. Esto se puede interpretar como la definición de un contrato donde consta que tipo de información se proporcionará al usuario interesado y durante cuánto tiempo será válido.
- **Fase 2, confirmación:** en esta fase el usuario interesado confirmará las credenciales de transcripción generadas por el usuario objetivo en la fase anterior. Esto se puede interpretar como la confirmación del contrato definido por el usuario objetivo.
- **Fase 3, envío:** durante esta fase el usuario objetivo deberá completar el esquema de transcripción con los atributos especificados, con información idónea. Dicha información podrá ser en formato texto o archivo (pdf) y deben establecerse con base a los atributos del esquema.
- **Fase 4, aceptación:** al llegar a esta fase el usuario interesado podrá revisar la información proporcionada por el usuario objetivo. Una vez revisada la información, el usuario interesado decidirá si acepta la transacción, en caso de aceptarla, su wallet se abrirá para guardar la información de la transacción.
- **Fase 5, completado:** finalmente esta fase representa que la transacción ha sido completada con éxito, toda la información identificativa generada durante este proceso de transacción será eliminada para reducir su peso en la base de datos, quedando únicamente los datos del esquema de transcripción utilizado, el usuario interesado, el usuario objetivo, las fechas de inicio y fin, y estados de finalizado o cancelado.

En las fases que son aptas para cancelar la transacción son fase 1, fase 2, fase 3 y fase 4, mismas que están marcadas con una x (cancelar la transacción).

6.2.2. Fase 6: Codificación y pruebas de los contratos inteligentes:

a. Codificación del contrato inteligente

El contrato inteligente codificado es el **Esquema de Transcripción**, el código fuente de programación está detallado en el **Anexo 3:** Documentación de Codificación del Subsistema de Contratos Inteligente. Para lograr un mayor entendimiento se realizó el pseudocódigo de este contrato inteligente, ver Tabla 8.

Tabla 8: *Pseudocódigo del contrato inteligente.*

```
INICIO
   // Estructura del Esquema de Transcripción
   esquema: REG
    id: string
    nombre: string
    version: string
    atributos: string
   FIN REG

   // Estructura de la Wallet del Usuario
   usuario: REG
    did: string
    handle: integer
   FIN REG

   usuario.handle = REALIZAR [abrir_wallet(usuario)]
   crear_esquema(esquema, usuario)

   FUNC crear_esquema(esquema: REG, usuario: REG) RET: vacio
      id_esquema: string
      SI usuario.handle == 0 ENTONCES
         ESCRIBIR "Wallet no válida"
      FIN SI
      id_esquema = REALIZAR [crear_esquema_transcripcion(esqeuma, usuario)]
      SI id_esquema == "error" ENTONCES
         ESCRIBIR "Esquema de Transcripción no creado"
      FIN SI
      ESCRIBIR "Esquema de Transcripción creado"
      ESCRIBIR "Id del esquema: " + id_esquema
   FIN FUNC

FIN
```

Durante la ejecución del esquema de transcripción, existe dos funciones importantes:

- La funcionalidad "**abrir_wallet**", permite al usuario abrir su wallet en la blockchain la cual devuelve su identificador denominado handle.
- La funcionalidad "**crear_esquema_transcripción**", una vez el usuario hay abierto su wallet, este podrá crear esquemas de transcripción en la blockchain, para ello se debe enviar tanto la wallet del usuario como la estructura del esquema. Esta función devuelve el id del esquema de transcripción creado en la blockchain.

b. Pruebas del contrato inteligente

Las pruebas unitarias realizadas permitieron verificar el correcto funcionamiento del subsistema de contratos inteligentes separando el código principal en bloques de código que cumplen con un objetivo clave, asegurando que cada bloque funcione correctamente y eficientemente por separado. Donde el primer paso es configurar y crear la wallet del Gobierno en la cual todas las wallets creadas para los usuarios se registrarán en su dominio. Como segundo paso es crear los esquemas de transcripción para el intercambio de información a

través de la blockchain, en caso de haber algún error las transacciones de información no se realizarán convirtiéndose en el primer indicador de que existen errores en el código de los contratos inteligentes. Por último, se validó el proceso de transacciones comprobando que todas sus fases sean completadas con éxito. Se generaron 9 casos de prueba relacionadas con las funciones del contrato inteligente, ver en la Tabla 9. Para mayor detalle ver el **Anexo 4:** Plan de Pruebas Unitarias para el Subsistema de Contratos Inteligentes.

Tabla 9: *Resumen de las Pruebas Unitarias del subsistema de contrato inteligente.*

N. ° de Prueba Unitaria	Descripción de lo comprobado	Estado
PU-01	Se creó la wallet del usuario, y se verificó que se obtiene el identificador de la wallet	Correcto
PU-02	Se creó y almacenó el DID para la wallet del usuario	Correcto
PU-03	Se registró la wallet del usuario bajo el dominio del Gobierno	Correcto
PU-04	Se creó el Esquema de Transcripción	Correcto
PU-05	Se firmó y envió el esquema de transcripción a la blockchain	Correcto
PU-06	Se creó y envió la Oferta de Credencial de Transcripción	Correcto
PU-07	Se creó y envió la Solicitud de Credencial de Transcripción	Correcto
PU-08	Se creó y envió la Credencial de Transcripción	Correcto
PU-09	Se almacenó la Credencial de Transcripción en la wallet	Correcto

6.2.3. Fase 7: Diseño del subsistema de aplicación:

En esta fase se implementó el patrón de diseño Modelo-Vista-Controlador (MVC) a la DApp, ver Figura 12, cada componente contiene las siguientes características:

- **Modelo:** está compuesto por el diagrama de tablas de base de datos del módulo “Middleware”, necesario para que los usuarios puedan ingresar e interactuar con el sistema.
- **Vista:** está compuesto por el módulo “Front-end” desarrollado como una SPA, el cual proporciona las vistas al usuario para poder interactuar con el sistema.
- **Controlador:** está compuesto tanto por el controlador del módulo “Front-end” como el controlador del módulo “Middleware”, siendo su objetivo permitir que las interacciones de los usuarios con el sistema sean posibles, realizando acciones tanto el modelo como enviando peticiones hacia el módulo “Back-end”.

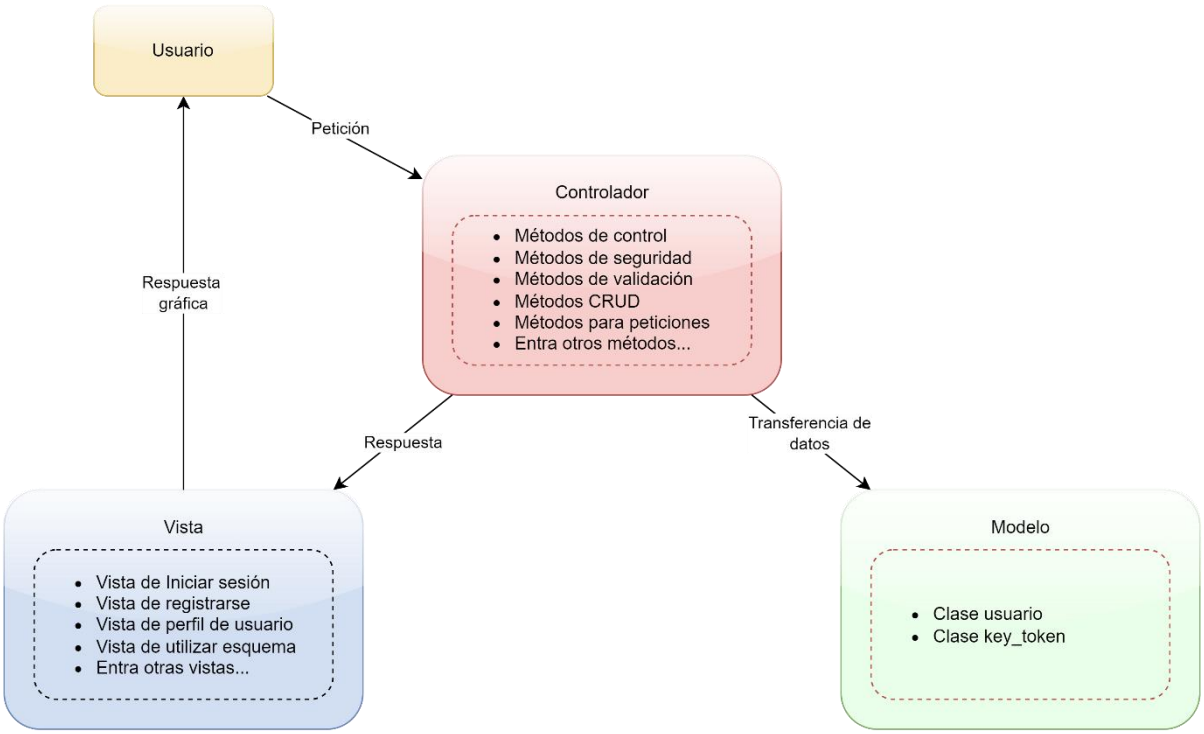


Figura 12: *Modelo-Vista-Controlador del subsistema de aplicación.*
Fuente: *Autor.*

En la Figura 13, se muestra el diagrama de tablas de la base de datos del módulo “Middleware”, compuesto por dos clases fundamentales: usuario y key_token. La clase usuario contiene información clave para el inicio de sesión en el sistema y para poder interactuar con la blockchain, mientras que key_token es la clase que brinda a los usuarios el control de seguridad para poder realizar acciones en el sistema, evitando que agentes maliciosos puedan manipular o corromper la información, o a su vez ejecutar acciones pretendiendo ser un usuario específico. Las dos clases tiene una relación de 1 a 1, por lo tanto, a cada usuario le pertenece una única instancia de key_token.

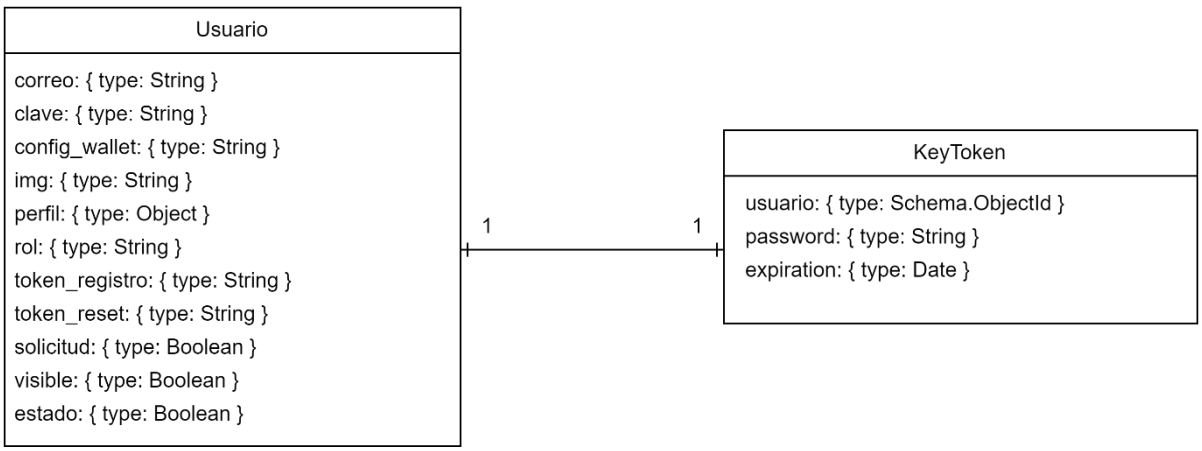


Figura 13: *Diagrama de tablas de Base de Datos del módulo "Middleware".*
Fuente: *Autor.*

6.2.4. Fase 8: Codificación y prueba del subsistema de aplicación:

a. Codificación

La codificación del subsistema de aplicación está desarrollada mediante el lenguaje de programación JavaScript, utilizando el entorno de ejecución Nodejs. Para el módulo “Front-end” se utilizó el framework Vuejs para crear todas las vistas y funcionalidades que permiten al usuario interactuar con el sistema, mientras que para el módulo “Middleware” se utilizó el framework de Expressjs para crear las APIs, lógica del negocio y conexiones con el módulo “Back-end” que permiten al usuario utilizar las funcionalidades del sistema. El código fuente de programación está detallado en el **Anexo 5:** Documentación de Codificación del Subsistema de Aplicación.

En la Tabla 10 se muestra el pseudocódigo de las principales funcionalidades del usuario, dependiendo de la opción se actualiza la información personal, la información de cursos o la información de transacciones. Cada opción de actualizar envía una petición al módulo “Back-end”, el cual se encarga de procesar la información, validarla y ejecuta la acción en la blockchain, y devuelve un mensaje con el estado de la petición.

Tabla 10: *Pseudocódigo de las funcionalidades de actualizar del usuario.*

```
INICIO
   // Estructura de la Wallet del Usuario
   usuario: REG
    did: string
    handle: integer
   FIN REG

   opcion: integer              // Variable para conocer la acción a realizar
   informacion: string          // Información personal del usuario
   cursos: string               // Información de cursos del usuario
   transaccion: string          // Información de transacciones del usuario

   SI opcion == 1 ENTONCES
      actualizar_informacion(usuario, informacion)
   FIN
   SI opcion == 2 ENTONCES
      actualizar_cursos(usuario, cursos)
   FIN
   SI opcion == 3 ENTONCES
      actualizar_transaccion(usuario, transaccion)
   FIN

   FUNC actualizar_informacion(usuario: REG, informacion: string) RET: null
      mensaje = string
      mensaje = REALIZAR [actualizar_informacion_usuario_wallet(usuario, informacion)]
      SI mensaje == "error" ENTONCES
         ESCRIBIR "Información personal no actualizada"
      FIN
      ESCRIBIR "Información personal actualizada"
   FIN FUNC

   FUNC actualizar_cursos(usuario: REG, cursos: string) RET: null
```

```
    mensaje = string

    mensaje = REALIZAR [actualizar_cursos_usuario_wallet(usuario, cursos)]
    SI mensaje == "error" ENTONCES
       ESCRIBIR "Información de cursos no actualizada"
    FIN
    ESCRIBIR "Información de cursos actualizada"
  FIN FUNC

  FUNC actualizar_transaccion(usuario: REG, transaccion: string) RET: null
    mensaje = string
    mensaje = REALIZAR [actualizar_transaccion_usuario_wallet(usuario, transaccion)]
    SI mensaje == "error" ENTONCES
       ESCRIBIR "Información de transaccion no actualizada"
    FIN
    ESCRIBIR "Información de transaccion actualizada"
  FIN FUNC

FIN
```

En la Tabla 11 se muestra el pseudocódigo de las principales funcionalidades en la transacción, dependiendo de la fase de la transacción se ejecuta la función para: definir, confirmar, enviar la información o aceptar la transacción. Cada función envía una petición al módulo “Back-end”, el cual se encarga de procesar la información, validarla y ejecuta la acción en la blockchain, y devuelve un mensaje con el estado de la petición.

Tabla 11: *Pseudocódigo de las funcionalidades en la transacción.*

```
INICIO
  // Estructura de la Wallet del Usuario
  usuario: REG
   did: string
   handle: integer
  FIN REG

  fase: integer            // Variable para conocer la acción a realizar
  codigo: string           // Variable para la transacción en proceso
  informacion: string      // Información solicitada en la transacción

  SI fase == 1 ENTONCES
     definir_transaccion(usuario, codigo)
  FIN
  SI fase == 2 ENTONCES
     confirmar_transaccion(usuario, codigo)
  FIN
  SI fase == 3 ENTONCES
     enviar_informacion_transaccion(usuario, codigo, informacion)
  FIN
  SI fase == 4 ENTONCES
     aceptar_transaccion(usuario, codigo)
  FIN

  FUNC definir_transaccion(usuario: REG, codigo: string) RET: null
     mensaje = string
     mensaje = REALIZAR [definir_transaccion_blockchain(usuario, codigo)]
     SI mensaje == "error" ENTONCES
           ESCRIBIR "Transacción no definida"
     FIN
```

```
    ESCRIBIR "Transacción definida"
  FIN FUNC

  FUNC confirmar_transaccion(usuario: REG, codigo: string) RET: null
    mensaje = string
    mensaje = REALIZAR [confirmar_transaccion_blockchain(usuario, codigo)]
    SI mensaje == "error" ENTONCES
        ESCRIBIR "Transacción no confirmada"
    FIN
    ESCRIBIR "Transacción confirmada"
  FIN FUNC

  FUNC enviar_informacion_transaccion_transaccion(usuario: REG, codigo: string, informacion:
string) RET: null
    mensaje = string
    mensaje = REALIZAR [confirmar_transaccion_blockchain(usuario, codigo, informacion)]
    SI mensaje == "error" ENTONCES
        ESCRIBIR "Información no enviada a la transacción"
    FIN
    ESCRIBIR "Información enviada a la transacción"
  FIN FUNC

  FUNC aceptar_transaccion(usuario: REG, codigo: string) RET: null
    mensaje = string
    mensaje = REALIZAR [aceptar_transaccion_blockchain(usuario, codigo)]
    SI mensaje == "error" ENTONCES
        ESCRIBIR "Transacción no aceptada"
    FIN
    ESCRIBIR "Transacción aceptada"
  FIN FUNC

FIN
```

b. Pruebas

Las pruebas unitarias realizadas permitieron verificar el correcto funcionamiento del subsistema de aplicación, estas abarcaron el módulo “Front-end” y el módulo “Middleware”. Debido a la gran complejidad de los dos módulos, se decidió seleccionar las funcionalidades más relevantes de cada módulo para ser evaluadas. Se separó el código de las principales funcionalidades en bloques de código, asegurando que cada bloque funcione correcta y eficientemente por separado. Se generaron 9 casos de prueba relacionadas con las funciones del contrato inteligente, ver en la Tabla 12. Para mayor detalle ver el **Anexo 6:** Plan de Pruebas Unitarias para el Subsistema de Aplicación.

Tabla 12: *Resumen de las Pruebas Unitarias del subsistema de aplicación.*

N. ° de Prueba Unitaria	Descripción de lo comprobado	Estado
PU-01	Se creó el token dinámico al iniciar sesión el usuario	Correcto
PU-02	Se envió el correo con el token de registro al usuario	Correcto
PU-03	Se envió el correo con el token de resetear contraseña al usuario	Correcto
PU-04	Validar únicamente usuarios de la Universidad Nacional de Loja	Correcto
PU-05	Se estableció la cookie de sesión con su debido tiempo de expiración	Correcto

PU-06	Se borró la cookie de sesión del usuario	Correcto
PU-07	Se comprobó que las rutas de las vistas principales estén protegidas	Correcto
PU-08	Se validó el rol del usuario antes de realizar alguna acción	Correcto
PU-09	Se actualizó la información personal de usuario en su wallet	Correcto
PU-10	Se actualizó la información de los cursos del usuario en su wallet	Correcto
PU-11	Se actualizó la información de la transacción del usuario en su wallet	Correcto
PU-12	Se definió la transacción en la blockchain	Correcto
PU-13	Se confirmó la transacción en la blockchain	Correcto
PU-14	Se envió la información solicitada de la transacción en la blockchain	Correcto
PU-15	Se aceptó la información de la transacción en la blockchain	Correcto

6.3. **Objetivo 3:** Probar el sistema de identidad digital auto-gestionada en un ambiente controlado.

La fase 9 de la metodología ABCDE se compone por:

6.3.1. Integrar el sistema (DApp)

En este punto se realizó el Plan de Pruebas de Integración, las cuales permitieron comprobar que los módulos del sistema en conjunto funcionan correctamente, y, por lo tanto, se validó que los subsistemas de contratos inteligentes y aplicación se integraron exitosamente. La Tabla 13 muestra un resumen de los casos de pruebas de integración generados y ejecutados para los principales requerimientos funcionales del sistema. Para mayor detalle de las pruebas de integración generadas para el sistema, ver el **Anexo 7:** Plan de Pruebas de Integración.

Tabla 13: *Resultados de las Pruebas de Integración.*

Id. Caso de Prueba	Componente	Descripción del Caso de Prueba	Estado
CP01	Subsistema de contratos inteligentes – Subsistema de aplicación	Se registra correctamente el usuario en el sistema	Correcto
CP02	Subsistema de contratos inteligentes – Subsistema de aplicación	Se inicia sesión correctamente en el sistema	Correcto
CP03	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario administra su información personal	Correcto
CP04	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario administra sus cursos de la página de la Carrera de Computación	Correcto
CP05	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario administra sus transacciones	Correcto
CP06	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario utiliza los esquemas	Correcto

CP07	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario visualiza su historial de transacciones	Correcto
CP08	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario realiza el proceso de transacción correctamente	Correcto

6.3.2. Probar el sistema (DApp)

En este punto se realizaron el Plan de Pruebas Funcionales (ver **Anexo 8:** Plan de Pruebas Funcionales) y el Plan de Pruebas de Aceptación (ver **Anexo 9:** Plan de Pruebas de Aceptación) del sistema.

a. Plan de Pruebas Funcionales

El Plan de Pruebas Funcionales permitió comprobar que el sistema desarrollado cumple y satisface con todos los requisitos funcionales especificados en el **Objetivo 1** del proyecto. En la Tabla 13 se muestra el resumen de los casos de prueba ejecutados para la validación de requisitos funcionales.

Tabla 14: *Resumen de los casos de pruebas del Plan de Pruebas Funcionales.*

Id. Caso de Prueba	Descripción del caso de prueba	RF Relacionados	Estado
CP01	Se probará la respuesta del sistema cuando el usuario vaya a iniciar sesión	RF01 - RF02	Correcto
CP02	Se probará la respuesta del sistema cuando el usuario vaya a registrarse	RF01 - RF03	Correcto
CP03	Se probará la respuesta del sistema cuando el usuario vaya a resetear su contraseña	RF01 - RF03 - RF04	Correcto
CP04	Se probará la respuesta del sistema cuando el usuario vaya a administrar su información personal	RF01 - RF02 - RF05 - RF09	Correcto
CP05	Se probará la respuesta del sistema cuando el usuario vaya a administrar sus cursos	RF01 - RF02 - RF06 - RF09	Correcto
CP06	Se probará la respuesta del sistema cuando el usuario vaya a administrar sus transacciones	RF01 - RF02 - RF09	Correcto
CP07	Se probará la respuesta del sistema cuando el usuario vaya a utilizar algún esquema	RF01 - RF02 - RF07	Correcto
CP08	Se probará la respuesta del sistema cuando el usuario vaya a buscar el perfil de otro usuario	RF01 - RF02 - RF08 - RF09	Correcto
CP09	Se probará la respuesta del sistema cuando el usuario vaya a administrar su perfil	RF01 - RF02 - RF05 - RF06 - RF09	Correcto
CP10	Se probará la respuesta del sistema cuando el usuario vaya a administrar los esquemas	RF01 - RF02 - RF10	Correcto

CP11	Se probará la respuesta del sistema cuando el usuario vaya a administrar a los usuarios	RF01 - RF02 - RF11	Correcto

b. Plan de Pruebas de Aceptación

El Plan de Pruebas de Aceptación permitió determinar el nivel de aceptación que tiene el sistema desarrollado, por lo tanto, se aplicó una encuesta a una muestra de 50 estudiantes de la Universidad Nacional de Loja pertenecientes a la Carrera de Ingeniería en Computación. La muestra de estudiantes realizó la manipulación del sistema después de la presentación sobre sus funcionalidades principales y de conocer la necesidad social a solventar. En la Tabla 15 se muestran las preguntas evaluadoras y sus resultados al aplicar la encuesta.

Tabla 15: *Resumen de las Pruebas de Aceptación.*

Nro.	Pregunta	Porcentajes (%)		
		Sí	**Parcialmente**	**No**
1	¿Es simple el vocabulario utilizado?	82%	18%	0%
2	¿Se proporciona el tiempo suficiente para realizar las entradas de información?	90%	10%	0%
3	¿Se entienden la interfaz y su contenido?	66%	28%	6%
4	¿Resulta fácil identificar alguna acción?	61.2%	38.8%	0%
5	¿Resulta fácil entender el resultado de una acción?	66%	34&	0%
6	¿Está diseñada la interfaz para la realización de las tareas de forma eficiente?	80%	20%	0%
7	¿Son apropiados los mensajes presentados por el sistema?	86%	12%	2%
8	¿Actúa el sistema en la prevención de errores?	78%	22%	0%
9	¿El sistema informa claramente sobre los errores presentados?	80%	16%	4%
10	¿Permite una cómoda navegación dentro del sistema y una fácil salida de este?	82%	14%	4%
11	¿Se presenta al usuario la información que sólo necesita?	84%	14%	2%
	Promedio	**77.74%**	**20.61%**	**1.45%**

Con los siguientes resultados obtenidos en los criterios: sí (77.74%), parcialmente (20.61%) y no (1.45%), y aplicando los criterios de aceptación que se muestran en la Tabla 16, se determinó que el nivel de aceptación del sistema de identidad digital auto-gestionada es **positivo.**

Tabla 16: *Criterio de aceptación del sistema.*

Pregunta	Porcentajes (%)		
	Sí	**Parcialmente**	**No**
Positivo	>=70%	<=30%	<=3%
Negativo	<70%	>30%	>3%

6.3.3. Desplegar el sistema (DApp)

En este punto se realizó el despliegue de la DApp de forma local, para ello se utilizaron 4 ambientes para los módulos “Front-end”, “Middleware”, “Back-end” y para la red Blockchain. La conexión entre los 3 módulos es mediante peticiones y respuestas HTTP, mientras que la conexión a la red Blockchain la realiza el módulo de “Back-end” mediante los métodos proporcionados por Hyperledger Indy.

En las Figuras 14, 15 y 16 se puede observar la estructura de los archivos en el panel izquierdo, mientras que en el panel derecho se muestra la información del despliegue de cada módulo en el servidor local (computadora del autor), todos se mantienen el ambiente de desarrollo.

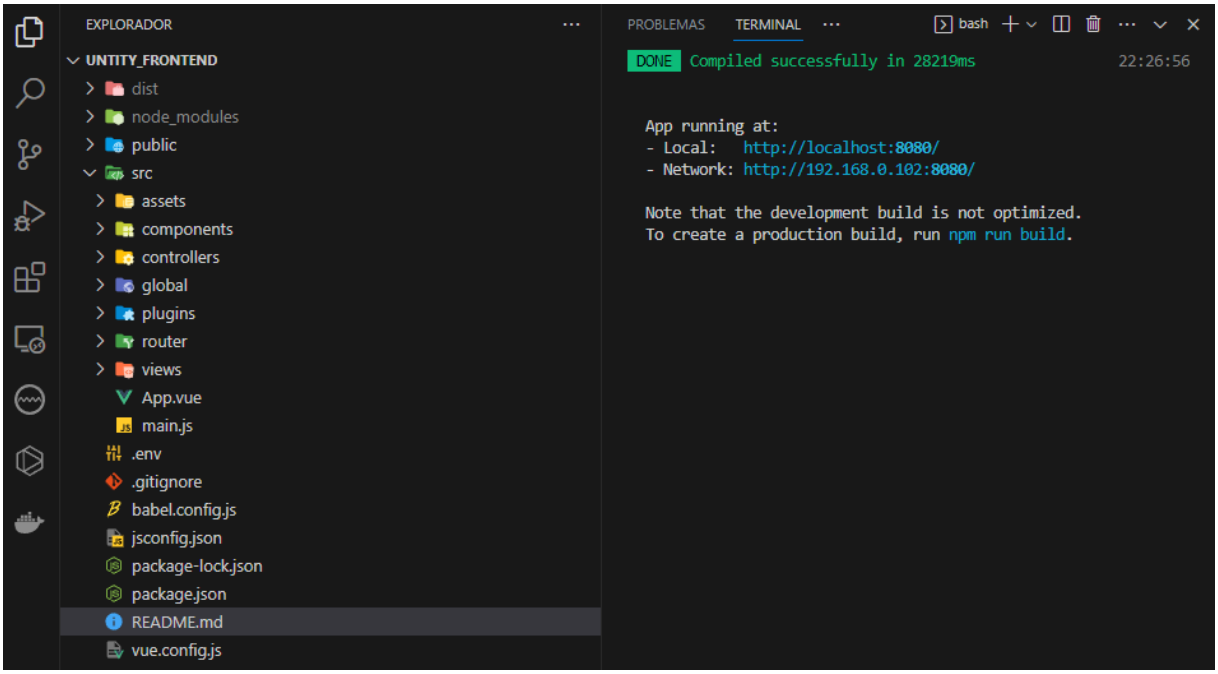


Figura 14: *Despliegue local de módulo "Front-end".*
Fuente: *Autor*

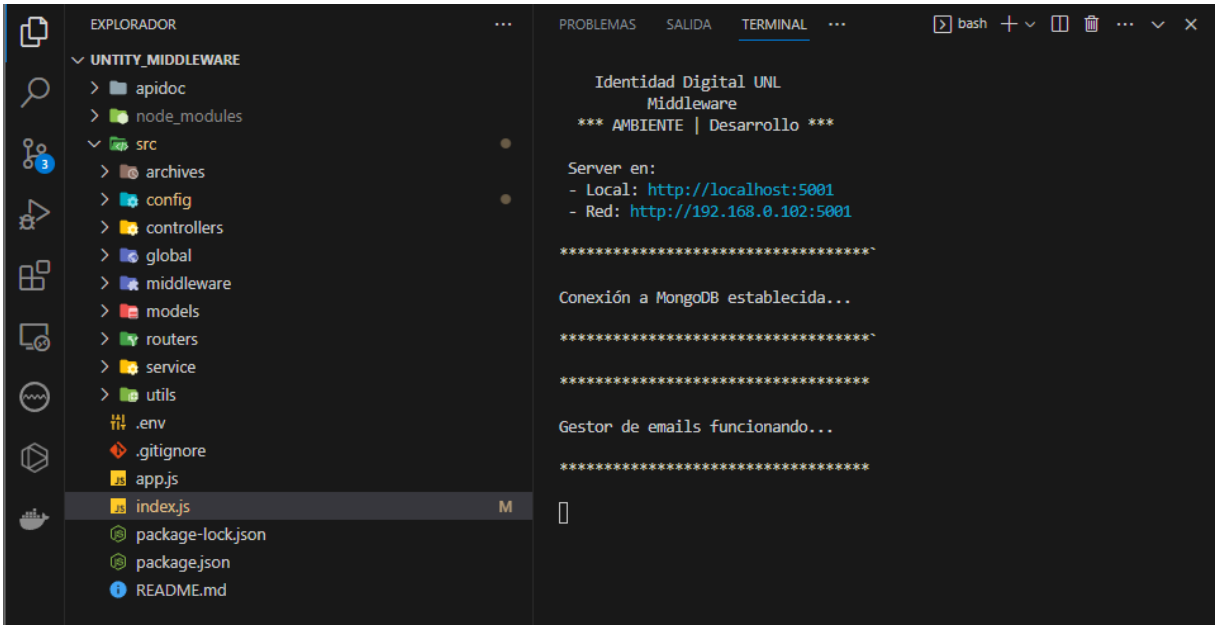


Figura 15: *Despliegue local de módulo "Middleware".*
Fuente: *Autor*

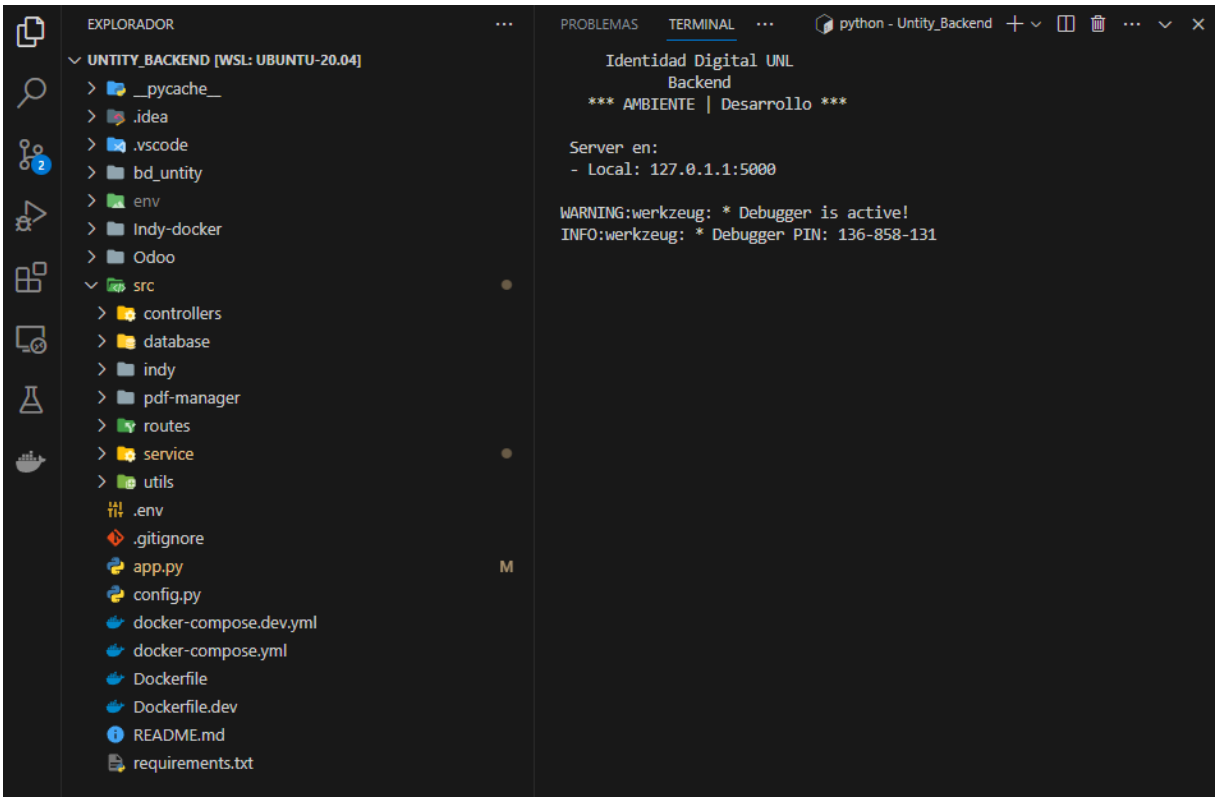


Figura 16: *Despliegue local de módulo "Back-end".*
Fuente: *Autor*

En la Figura 17 se muestran los siguientes contenedores de Docker: “blockchain”, “bd_untity” y “odoo”. El contendor de “blockchain” contiene la Blockchain de Hyperledger Indy y también sus nodos participantes, “bd_untity” contiene la base de datos utilizada por el

módulo “Back-end” y “odoo” es la simulación de la página de la Carrera de Computación donde se obtienen los usuarios y los cursos que hayan participado.

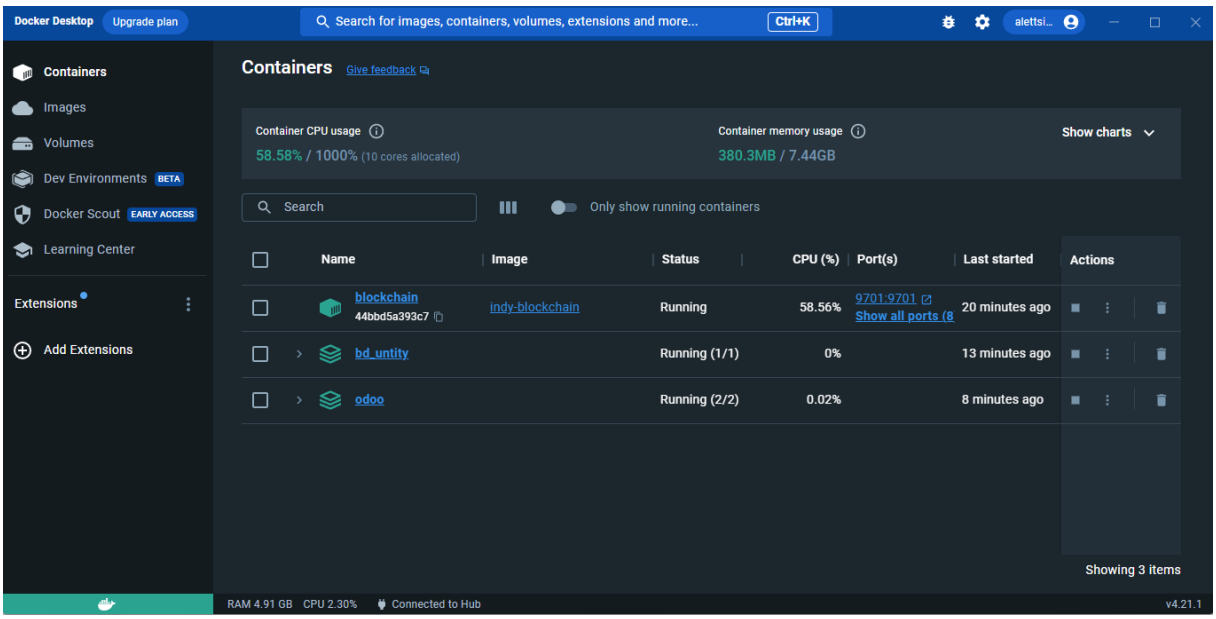


Figura 17: *Despliegue local de la red Blockchain.*
Fuente: *Autor*

En la Tabla 17 se muestra la configuración del Dockerfile para implementar la Blockchain de Hyperledger Indy junto con 4 nodos base. Dichos nodos son necesarios para la ejecución del protocolo de consenso que permite añadir nuevos bloques con la información de las transacciones recientes.

Tabla 17: *Codificación de Dockerfile para iniciar la Blockchain según Hyperledger Indy*

```
FROM ubuntu:16.04

ARG uid=1000

# Install environment
RUN apt-get update -y && apt-get install -y \
        git \
        wget \
        python3.5 \
        python3-pip \
        python-setuptools \
        python3-nacl \
        apt-transport-https \
        ca-certificates \
        supervisor

RUN pip3 install -U \
        pip==9.0.3 \
        setuptools

RUN apt-key adv --keyserver keyserver.ubuntu.com --recv-keys CE7709D068DB5E88 || \
        apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys CE7709D068DB5E88
ARG indy_stream=master
RUN echo "deb https://repo.sovrin.org/deb xenial $indy_stream" >> /etc/apt/sources.list
```

```
RUN useradd -ms /bin/bash -u $uid indy

ARG indy_plenum_ver=1.12.1~dev989
ARG indy_node_ver=1.12.1~dev1172
ARG python3_indy_crypto_ver=0.4.5
ARG indy_crypto_ver=0.4.5
ARG python3_pyzmq_ver=18.1.0
ARG python3_orderedset_ver=2.0
ARG python3_psutil_ver=5.4.3
ARG python3_pympler_ver=0.5

RUN apt-get update -y && apt-get install -y \
        python3-pyzmq=${python3_pyzmq_ver} \
        indy-plenum=${indy_plenum_ver} \
        indy-node=${indy_node_ver} \
        python3-indy-crypto=${python3_indy_crypto_ver} \
        libindy-crypto=${indy_crypto_ver} \
        python3-orderedset=${python3_orderedset_ver} \
        python3-psutil=${python3_psutil_ver} \
        python3-pympler=${python3_pympler_ver} \
        vim

RUN echo "[supervisord]\n\
logfile = /tmp/supervisord.log\n\
logfile_maxbytes = 50MB\n\
logfile_backups=10\n\
logLevel = error\n\
pidfile = /tmp/supervisord.pid\n\
nodaemon = true\n\
minfds = 1024\n\
minprocs = 200\n\
umask = 022\n\
user = indy\n\
identifier = supervisor\n\
directory = /tmp\n\
nocleanup = true\n\
childlogdir = /tmp\n\
strip_ansi = false\n\
\n\
[program:node1]\n\
command=start_indy_node Node1 0.0.0.0 9701 0.0.0.0 9702\n\
directory=/home/indy\n\
stdout_logfile=/tmp/node1.log\n\
stderr_logfile=/tmp/node1.log\n\
\n\
[program:node2]\n\
command=start_indy_node Node2 0.0.0.0 9703 0.0.0.0 9704\n\
directory=/home/indy\n\
stdout_logfile=/tmp/node2.log\n\
stderr_logfile=/tmp/node2.log\n\
\n\
[program:node3]\n\
command=start_indy_node Node3 0.0.0.0 9705 0.0.0.0 9706\n\
directory=/home/indy\n\
stdout_logfile=/tmp/node3.log\n\
stderr_logfile=/tmp/node3.log\n\
\n\
[program:node4]\n\
command=start_indy_node Node4 0.0.0.0 9707 0.0.0.0 9708\n\
directory=/home/indy\n\
```

```
stdout_logfile=/tmp/node4.log\n\
stderr_logfile=/tmp/node4.log\n"\
>> /etc/supervisord.conf

USER indy

RUN awk '{if (index($1, "NETWORK_NAME") != 0) {print("NETWORK_NAME = \"sandbox\"")} else
print($0)}' /etc/indy/indy_config.py> /tmp/indy_config.py
RUN mv /tmp/indy_config.py /etc/indy/indy_config.py

ARG pool_ip=127.0.0.1

RUN generate_indy_pool_transactions --nodes 4 --clients 5 --nodeNum 1 2 3 4 --
ips="$pool_ip,$pool_ip,$pool_ip,$pool_ip"

EXPOSE 9701 9702 9703 9704 9705 9706 9707 9708

CMD ["/usr/bin/supervisord"]
```

7. Discusión

El prototipo de sistema para este Proyecto de Integración Curricular denominado "Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja", otorga a los miembros de la Universidad una identidad digital única, válida, segura, fiable y transparente, estas características son garantizadas y aseguradas por la blockchain de Hyperledger Indy. Esto responde a la pregunta de investigación: **¿Cómo asegurar una identidad digital auto-gestionada para los usuarios de la Universidad Nacional de Loja?**

Al realizar la revisión de los trabajos relacionados [21], [28], [29], se observó que ninguna de las metodologías aplicadas sigue una metodología oficial que integre la blockchain junto a la aplicación. Por tal motivo se contrasta que la metodología de desarrollo de software ABCDE (ver en el **punto 4.10 de la sección Marco Teórico**) es la única para este tipo de proyectos, la cual integra los contratos inteligentes (chaincode) de la blockchain (de Hyperledger Indy) junto a una aplicación tradicional dando como resultado una DApp. Por lo tanto, se cambia a la metodología de desarrollo de software ABCDE, dejando a la metodología ICONIX (establecida en el **anteproyecto**) fuera del proyecto debido a su enfoque en aplicaciones tradicionales.

La metodología ABCDE sirve de guía en la construcción de este proyecto, y permite que el sistema tenga una base sólida para su correcto funcionamiento al aplicar fases de diseño, de codificación, de ejecución de pruebas y de integración a sus subsistemas (contratos inteligente y aplicación). Para el desarrollo de este proyecto se divide la totalidad en 3 fases, siendo estas los objetivos específicos.

7.2. Objetivo 1: Definir el sistema para la identidad digital académica auto-gestionada usando la Ingeniería de Requisitos.

En la primera fase de la metodología se indica el objetivo del sistema denominado "Desarrollar un sistema de identidad digital mediante tecnología Blockchain para la Universidad Nacional de Loja", teniendo como objetivo que los usuarios tengan el control total de su información digital. Las siguientes fases 2 y 3 se completan utilizando el documento Especificación de Requisitos de Software perteneciente al estándar IEEE 830, donde en la fase 3 se cambian las historias de usuario por requisitos funcionales y no funcionales porque es más viable y evita problemas en las siguientes fases. Por último, en la cuarta fase se divide el sistema en dos subsistemas (contratos inteligentes y aplicación), y además se define la arquitectura (ver en el **punto 6.1.4 de la sección Resultados**).

Al seguir la metodología ABCDE se consigue definir adecuadamente los actores, requisitos y arquitectura del sistema tomando en cuenta los subsistemas de contratos

inteligentes y aplicación tal cual lo realizan en los trabajos relacionados de [28], [30], a diferencia de [21], [29] que al no aplicar esta metodología su criterio para la definición de la DApp es empírica y no confiable.

7.2. Objetivo 2: Desarrollar el sistema para la identidad digital académica auto-gestionada mediante tecnología Blockchain.

Las tecnologías utilizadas para el desarrollo de los subsistemas son: Vuejs, Nodejs, MongoDB y Express.js para el subsistema de aplicación, mientras que Flask, PostgreSQL e Indy-Node para el subsistema de contratos inteligentes; aunque existen más tecnologías que se pueden utilizar, se decidió por las mencionadas por el conocimiento y experiencia del autor sobre ellas.

Para el desarrollo del subsistema de contratos inteligentes (ver el **punto 6.2.1 de la sección Resultados)** se investigó en la documentación oficial de Hyperledger Indy de cómo utilizar los métodos que proporciona para interactuar con la blockchain. A partir del aprendizaje adquirido, se configura y crea la Pool, Wallets, DIDs, Esquemas de Transcripción y Proceso de Transacción según los parámetros establecidos por Indy, esto se asemeja a lo realizado en [21] pero con las diferencias en que se controla la configuración al momento de crear wallets y DIDs mientras que en otros Frameworks de blockchain ya tienen sus wallets y contratos inteligentes definidos limitando sus opciones de uso, si por alguna razón las credenciales se repiten se generará un error imposibilitando su creación y uso. Para el proceso de transacciones se lo divide en 4 fases para otorgar un mayor control al usuario si continua o no con la transacción, y lo más importante es la implementación en la wallet de métodos para guardar información personal, estos son algunas de las características de este proyecto frente a los demás trabajos relacionados [21], [28], [29], [30] que tienen por enfoque las transacciones entre sus usuarios y no como asegurar su información. Cabe destacar que la blockchain de Hyperledger Indy no guarda los identificadores de los DIDs, ni los IDs de los Esquemas de Transcripción y tampoco las fases del Proceso de Transacción en un histórico, por ende, se utiliza una base de datos (en PostgreSQL) para administrar esa información.

Para lograr que el subsistema de aplicación (ver el **punto 6.2.3 de la sección Resultados)** sea seguro y confiable se aplican diversos métodos de seguridad, uno de ellos es el algoritmo de tokens dinámicos utilizando JWT, que consiste en que cada inicio de sesión del usuario genera un token diferente para comprobar y validar que el usuario tiene acceso a las funcionalidades de la DApp. Mientas que en [21], [29], [30] se desconoce qué tipo de seguridad es aplicada para el subsistema de aplicación, a excepción de [28] que utiliza una la librería de PassportJS para controlar las interacciones de los usuarios.

7.3. Objetivo 3: Probar el sistema de identidad digital académica auto-gestionada en un ambiente controlado.

En contraste con [30] que despliega su blockchain en un testnet de Rinkeby, para la blockchain de este proyecto no existe un testnet donde se pueda desplegarla y probarla, por tal motivo se hace el despliegue según lo especifica Hyperledger Indy, en el cual se configura la red para la cadena de bloques y sus nodos dentro de un mismo contenedor de Docker convirtiéndose en el servidor de la red Blockchain (ver el **punto 6.3.3 de la sección Resultados**), parecido al despliegue que se realiza en [21], [28], [29]. Para la ejecución del objetivo 3 se lo realiza en un ambiente controlado.

La ejecución de las pruebas de integración (ver el **punto 6.3.1 de la sección Resultados**) y pruebas de aceptación (ver el **punto 6.3.2 de la sección Resultados**) se realizaron mediante un servidor local (computadora del autor), donde las pruebas de integración permiten validar el cumplimiento exacto de los requisitos funcionales y no funcionales determinados en el primer objetivo, mientras que las pruebas de aceptación indican el nivel de aprobación de la DApp frente a una muestra de 50 estudiantes, dando un resultado positivo tomando en cuenta los criterios de evaluación establecidos.

Las transacciones realizadas utilizando los esquemas de transcripción no tienen ningún costo económico, esto se debe a que la blockchain de Hyperledger Indy es una red descentralizada permisionada, lo que significa que los usuarios podrán realizar transacciones de forma gratuita. Esta característica permite que los usuarios puedan intercambiar información sin necesidad de preocuparse por los costos económicos, únicamente deben respetar las normas del sistema que en caso de no hacerlo sus cuentas se desactivaran.

8. Conclusiones

Una vez culminado el Proyecto de Integración Curricular, se concluye lo siguiente:

- La implementación de la tecnología Blockchain permite asegurar la identidad digital por sus características de inmutabilidad, transparencia, fiabilidad y seguridad mediante los métodos de Hyperledger Indy (Wallet, DID, Ledger, entre otros), además otorga a sus usuarios el control total de su información digital cumpliendo con el objetivo del presente Proyecto de Integración Curricular.
- Las cuatro primeras fases de la metodología ABCDE junto con el estándar de requisitos IEEE 830 permiten definir de forma conveniente el sistema para asegurar la identidad digital académica de los miembros de la Universidad Nacional de Loja.
- La aplicación de la metodología ágil ABCDE garantiza la construcción de una DApp sólida, confiable y segura, al estar compuesta por algunas fases enfocadas al diseño, a la codificación y a las pruebas para los subsistemas, lo cual permite validar sus desempeños antes de realizar la integración.
- En la fase final de la metodología ABCDE se realizan las pruebas de Integración, Funcionales y Aceptación, que validan el correcto funcionamiento en conjunto de los subsistemas de la DApp, el cumplimiento y la satisfacción de los requisitos determinados, y también se obtiene, a partir de una muestra, el nivel de aceptación del sistema, siendo **positivo** para este proyecto**.**
- Los esquemas de transcripción son el contrato inteligente de Hyperledger Indy, el cual es la pieza clave para realizar las transacciones ya que en base a sus especificaciones los usuarios pueden intercambiar información.
- La blockchain de Hyperledger Indy no tiene costos por transferencia, lo cual facilita la adquisición de información entre sus usuarios, únicamente depende de la determinación de los usuarios al momento de iniciar y finalizar alguna transacción.
- La construcción de una DApp enfocada a la identidad digital auto-gestionada permite que sus usuarios administren y controlen su información sin preocuparse en que pueda ser manipulada o expuesta sin su consentimiento, algo que sucede en las aplicaciones tradicionales. Por tal motivo, el producto del presente Proyecto de Integración Curricular garantiza que la identidad digital de sus usuarios está segura y únicamente el usuario podrá administrar su información, además decidirá si la comparte.

9. Recomendaciones

Una vez culminado el Proyecto de Integración Curricular, se recomienda lo siguiente:

- Aplicar la metodología ABCDE para el desarrollo de aplicaciones de escritorio, móviles o web donde se vaya a implementar la tecnología Blockchain, debido a que esta metodología establece las fases y guía hacia la correcta integración entre los contratos inteligentes con una aplicación tradicional.
- Tener paciencia y determinación para entender el funcionamiento de Hyperledger Indy y sus métodos, para crear las Wallets en su configuración de credenciales es factible añadir algo aleatorio para evitar posibles errores, los identificadores que se generen deben ser guardados para poder tener acceso a la información que referencia, pero sobre todo asegurar y proteger el acceso a los DIDs ya que son la pieza fundamental para poder interactuar con la Blockchain.
- Tomar como referencia este Proyecto de Integración Curricular para construir sistemas o DApps enfocadas a la identidad digital utilizando tecnología Blockchain. Ya que a nivel nacional únicamente se encontró un trabajo relacionado a la temática pero que utiliza Hyperledger Fabric.

Se recomienda para trabajos futuros:

- Investigar a profundidad los demás métodos (blob_storage, cache, crypto, libindy, non_secrets, pairwise, payment, pool) de Hyperledger Indy no utilizados en este proyecto debida a su poca documentación e implementación, puesto que pueden proveer de mejores características a la identidad digital auto-gestionada y a la información que se intercambia en las transacciones.
- Ampliar el alcance de la identidad digital fuera del entorno de la Universidad Nacional de Loja, para lograr una mayor interoperabilidad con diferentes empresas u organizaciones en beneficio de los usuarios.
- Agregar nuevos enfoques a la identidad digital diferentes a lo académico, por ejemplo, de economía, de salud, de turismo, de noticias, de procesos administrativos, entre otros.
- Implementar el despliegue de la blockchain de Hyperledger Indy mediante la separación e instalación de nodos en diferentes dispositivos, para lograr una disponibilidad perfecta de la información.

10. Bibliografía

[1] "Un cuarto de siglo de spam en internet: así nació el correo basura." Accessed: Aug. 27, 2023. [Online]. Available: https://www.bbvaopenmind.com/tecnologia/mundo-digital/cuarto-siglo-spam-internet-asi-nacio-correo-basura/

[2] "Así es como las grandes empresas venden tus datos en internet." Accessed: Aug. 27, 2023. [Online]. Available: https://www.elconfidencial.com/tecnologia/2015-09-14/asi-es-como-venden-tus-datos-personales-en-internet_1011071/

[3] "Tras masiva filtración, advierten que venta de datos personales no es nueva en Ecuador | DPL News." Accessed: Aug. 27, 2023. [Online]. Available: https://dplnews.com/tras-masiva-filtracion-advierten-que-venta-de-datos-personales-no-es-nueva-en-ecuador/

[4] "identidad | Definición | Diccionario de la lengua española | RAE - ASALE." Accessed: Aug. 16, 2023. [Online]. Available: https://dle.rae.es/identidad

[5] Ariel, "Identidad Digital: El nuevo usuario en el mundo digital." Accessed: Aug. 16, 2023. [Online]. Available: https://www.ufasta.edu.ar/biblioteca/files/2017/02/identidad_digital.pdf

[6] M. Da Silva and A. Pardo Vegezzi, "Identidad digital auto-gestionada: El futuro de la identidad digital: Auto-gestión, billeteras digitales y blockchain," *Self-Sovereign Identity: The Future of Identity: Self-Sovereignity, Digital Wallets, and Blockchain*, Sep. 2020, doi: 10.18235/0002635.

[7] C. Dolader, R. Joan, B. Roig, J. Luís, and M. Tapia, "LA BLOCKCHAIN: FUNDAMENTOS, APLICACIONES Y RELACIÓN CON OTRAS TECNOLOGÍAS DISRUPTIVAS", Accessed: Aug. 25, 2023. [Online]. Available: https://www.mincotur.gob.es/Publicaciones/Publicacionesperiodicas/EconomiaIndustrial/RevistaEconomiaIndustrial/405/DOLADER,%20BEL%20Y%20MUÑOZ.pdf

[8] J. Guaña-Moya, H. N. Roa, F. Marcillo, L. Ayavaca-Vallejo, M. Chiluisa-Chiluisa, and B. Moya-Carrera, "Tecnología Blockchain, qué es y cómo funciona", Accessed: Aug. 25, 2023. [Online]. Available: https://media.proquest.com/media/hms/PFT/1/qcdZR?_s=7sCG%2BCsmB3LnevDKGuynMhBJCDI%3D

[9] "¿Qué es blockchain? - IBM Blockchain | IBM." Accessed: Aug. 25, 2023. [Online]. Available: https://www.ibm.com/mx-es/topics/blockchain

[10] M. José and F. Iglesias, "Introducción a Blockchain, Contratos Inteligentes y Aplicaciones Descentralizadas Versión n../ES-Mayo oooo".

[11] "Lenguaje de Programación - Concepto, tipos y ejemplos." Accessed: Aug. 26, 2023. [Online]. Available: https://concepto.de/lenguaje-de-programacion/

[12] "¿Qué es JavaScript? - Explicación de JavaScript (JS) - AWS." Accessed: Aug. 26, 2023. [Online]. Available: https://aws.amazon.com/es/what-is/javascript/

[13] "¿Qué es Python? - Explicación del lenguaje Python - AWS." Accessed: Aug. 26, 2023. [Online]. Available: https://aws.amazon.com/es/what-is/python/

[14] J. Díaz, M. D. Tugnarelli, M. F. Fornaroli, L. Barboza, F. Miño, and J. I. Carubia Grieco, "Protocolos de consenso".

[15] "Análisis comparativo de Métodos de Consenso sobre plataformas Blockchain Comparative analysis of Consensus Methods on Blockchain platforms", doi: 10.37815/rte.v33n2.828.

[16] Y. Marrero Travieso, "La Criptografía como elemento de la seguridad informática," *ACIMED*, vol. 11, no. 6, pp. 0–0, 2003, Accessed: Aug. 16, 2023. [Online]. Available:

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352003000600012&lng=es&nrm=iso&tlng=es
[17] Y. Nir and A. Langley, “ChaCha20 and Poly1305 for IETF Protocols,” Jun. 2018, doi: 10.17487/RFC8439.
[18] M. A. J. #1, B. R. Sawant, and A. Deshmukh, “Single Page Application using AngularJS”, Accessed: Aug. 23, 2023. [Online]. Available: http://mydomain.com/myseo#key=value
[19] “Framework: qué es, para qué sirve y algunos ejemplos.” Accessed: Aug. 26, 2023. [Online]. Available: https://www.edix.com/es/instituto/framework/
[20] “Why Hyperledger Indy Is Important For Digital Identities?” Accessed: Aug. 26, 2023. [Online]. Available: https://101blockchains.com/hyperledger-indy/
[21] S. Paola and J. Vásquez, “Identidad digital basada en blockchain en instituciones educativas,” *instname:Universidad de los Andes*, 2020, Accessed: Aug. 23, 2023. [Online]. Available: https://repositorio.uniandes.edu.co/handle/1992/48895
[22] “Introducción a Express/Node - Aprende desarrollo web | MDN.” Accessed: Aug. 26, 2023. [Online]. Available: https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction
[23] “Introducción — Vue.js.” Accessed: Aug. 26, 2023. [Online]. Available: https://es.vuejs.org/v2/guide/
[24] “¿Qué es Express.js? Todo lo que Debes Saber.” Accessed: Aug. 26, 2023. [Online]. Available: https://kinsta.com/es/base-de-conocimiento/que-es-express/
[25] “Qué es Flask y ventajas que ofrece | OpenWebinars.” Accessed: Jun. 18, 2023. [Online]. Available: https://openwebinars.net/blog/que-es-flask/
[26] A. Tobin and D. Reed, “The Inevitable Rise of Self-Sovereign Identity A white paper from the Sovrin Foundation,” 2017.
[27] L. Marchesi, M. Marchesi, and R. Tonelli, “ABCDE—agile block chain DApp engineering,” *Blockchain: Research and Applications*, vol. 1, no. 1–2, p. 100002, Dec. 2020, doi: 10.1016/J.BCRA.2020.100002.
[28] M. C. Cáceres Salamea and D. F. Peralta Velecela, “Propuesta de identidad digital para historial clínico unificado utilizando tecnología blockchain,” Nov. 2021, Accessed: Aug. 23, 2023. [Online]. Available: http://dspace.ucuenca.edu.ec/handle/123456789/37324
[29] G. Sanz González, “Diseño e implementación de un sistema de identidad digital descentralizada para ciudadanos de la Unión Europea en el ámbito sanitario,” 2023, Accessed: Aug. 23, 2023. [Online]. Available: https://oa.upm.es/72965/
[30] E. Patricio and S. Malla, “Implementación de la tecnología Blockchain para la validación de autenticidad de los certificados académicos digitales / Blockchain technology implementation for the authenticity validation of digital academic certificates.,” May 2022, Accessed: Aug. 26, 2023. [Online]. Available: https://dspace.unl.edu.ec//handle/123456789/24723
[31] “Método analítico: Qué es, para qué sirve y cómo realizarlo.” Accessed: Aug. 23, 2023. [Online]. Available: https://www.questionpro.com/blog/es/metodo-analitico/
[32] “Investigación-acción: Qué es, etapas y ejemplos.” Accessed: Aug. 23, 2023. [Online]. Available: https://www.questionpro.com/blog/es/investigacion-accion/
[33] “¿Qué es una encuesta? | QuestionPro.” Accessed: Aug. 23, 2023. [Online]. Available: https://www.questionpro.com/es/una-encuesta.html

11. Anexos

Anexo 1: Especificación de requisitos de la DApp.

Especificación de requisitos de software

Proyecto: "Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja".

Autor: Alexis Armijos
Correo electrónico: carlos.a.armijos@unl.edu.ec

Ficha del documento

Versión	Fecha de revisión	Cambios	Motivos del cambio
1.0	10-05-2023	Inicio del desarrollo del documento.	Inicio del desarrollo del documento.
1.1	23-05-2023	Correcciones	Correcciones en los requisitos y diagrama de casos de uso
1.2	16-06-2023	Correcciones	Correcciones en Requisitos No Funcionales

Documento validado por las partes en fecha:

Revisor Técnico	Tesista	Cliente
		
Ing. Cristian Narváez	Alexis Armijos	Ing. Pablo Ordoñez

URL:
https://drive.google.com/file/d/1EN8y01Fp6MJnZgnxmIrfNeW3Jsi5O9Va/view?usp=drive_link

1 Introducción

Este documento es una Especificación de Requisitos Software para el desarrollo de una propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja. Esta especificación de requisitos se ha estructurado basándose en las directrices propuestas por el estándar Especificaciones de Requisitos Software ANSI/IEEE 830-1998.

1.1 Propósito

El presente documento tiene como finalidad la definición de las especificaciones funcionales, no funcionales y obtener la información necesaria para poder desarrollar la propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja. Este documento va dirigido a la Carrera de Computación.

1.2 Alcance

Esta ERS va dirigido al personal involucrado en el desarrollo del presente Trabajo de Integración Curricular, donde se pretende darle seguimiento a un marco metodológico para el desarrollo basado en un modelo de calidad, aplicando metodologías ágiles para el desarrollo de la aplicación.

1.3 Personal involucrado

Tabla 1. Personal involucrado estudiante de CIC

Nombre	Carlos Alexis Armijos rios
Rol	Analista y Desarrollador del Software
Categoría profesional	Estudiante de la CIC
Responsabilidades	Análisis de información, diseño y desarrollo del software
Información de contacto	carlos.a.armijos@unl.edu.ec

Tabla 2. Personal involucrado docente de CIC/S

Nombre	Cristian Ramiro Narváez Guillen
Rol	Director del proyecto
Categoría profesional	Docente de la de la CIS/C
Responsabilidades	Supervisión del proyecto
Información de contacto	cristian.narvaez@unl.edu.ec

Tabla 4. Personal involucrado Gestor de la Carrera de Computación

Nombre	Pablo Fernando Ordoñez Ordoñez
Rol	Cliente de la aplicación
Categoría profesional	Gestor de la Carrera de Computación
Responsabilidades	Proporcionar información sobre los cursos de la carrera.

Información de contacto	pfordonez@unl.edu.ec

1.4 Definiciones, acrónimos y abreviaturas

Tabla 5. Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Untity	**Un**ique Iden**tity** => Nombre del sistema
Blockchain	Es un libro mayor compartido e inalterable compuesta por nodos que facilitan el proceso de registro de transacciones
Wallet	Es una cartera, billetera o monedero virtual en el que podemos gestionar nuestros activos.
Usuario	Persona que usará el sistema
Creador	Usuario que podrá gestionar los Esquemas de Transcripción
Administrador	Creador que podrá gestionar las cuentas de los usuarios y creadores
Esquemas de Transcripción	Marco de datos que permitirán la interoperabilidad entre usuarios por la Blockchain de Hyperledger Indy
RBFT	Es un protocolo basado en nodos replicados con la misma información, dónde un nodo maestro ejecuta las solicitudes.
Cifrado ChaCha20-Poly1305	Es un cifrado de autenticación, que combina el cifrado de flujo ChaCha20 con el código de autenticación de mensajes Poly1305.
Cifrado HMAC-256	Calcula un código de autenticación de mensajes basado en hash (HMAC) mediante la función hash SHA256.
Sistema	Producto que sirve como base para hacer funcionar elementos adicionales a una página de Internet que ofrece una solución específica para la necesidad del usuario
Auto-gestionada	El usuario posee y controla su identidad sin la intervención de las autoridades administrativas
ERS	Especificación de Requisitos de Software
RNF	Requisito no Funcional
RF	Requisito Funcional
UNL	Universidad Nacional de Loja
CIC	Carrera de Ingeniería en Computación

1.5 Referencias

Tabla 6. Referencias

Referencia	Título	Ruta	Fecha	Autor
IEEE Recommended Practice for Software Requirements Specifications	IEEE Std 830-1998	Enlace	1998	IEEE

1.6 Resumen

Este documento está compuesto por tres secciones.

En la primera sección, se realiza una introducción y se proporciona una visión general de la especificación de recursos de la DApp.

En la segunda sección, se realiza una descripción general, con todos los datos asociados y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, en la tercera sección se definen detalladamente los requisitos del sistema.

2 Descripción general

2.1 Perspectiva del producto

La propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja, será desarrollada en la web, para alcanzar una mayor accesibilidad a sus usuarios.

2.2 Funcionalidad del producto

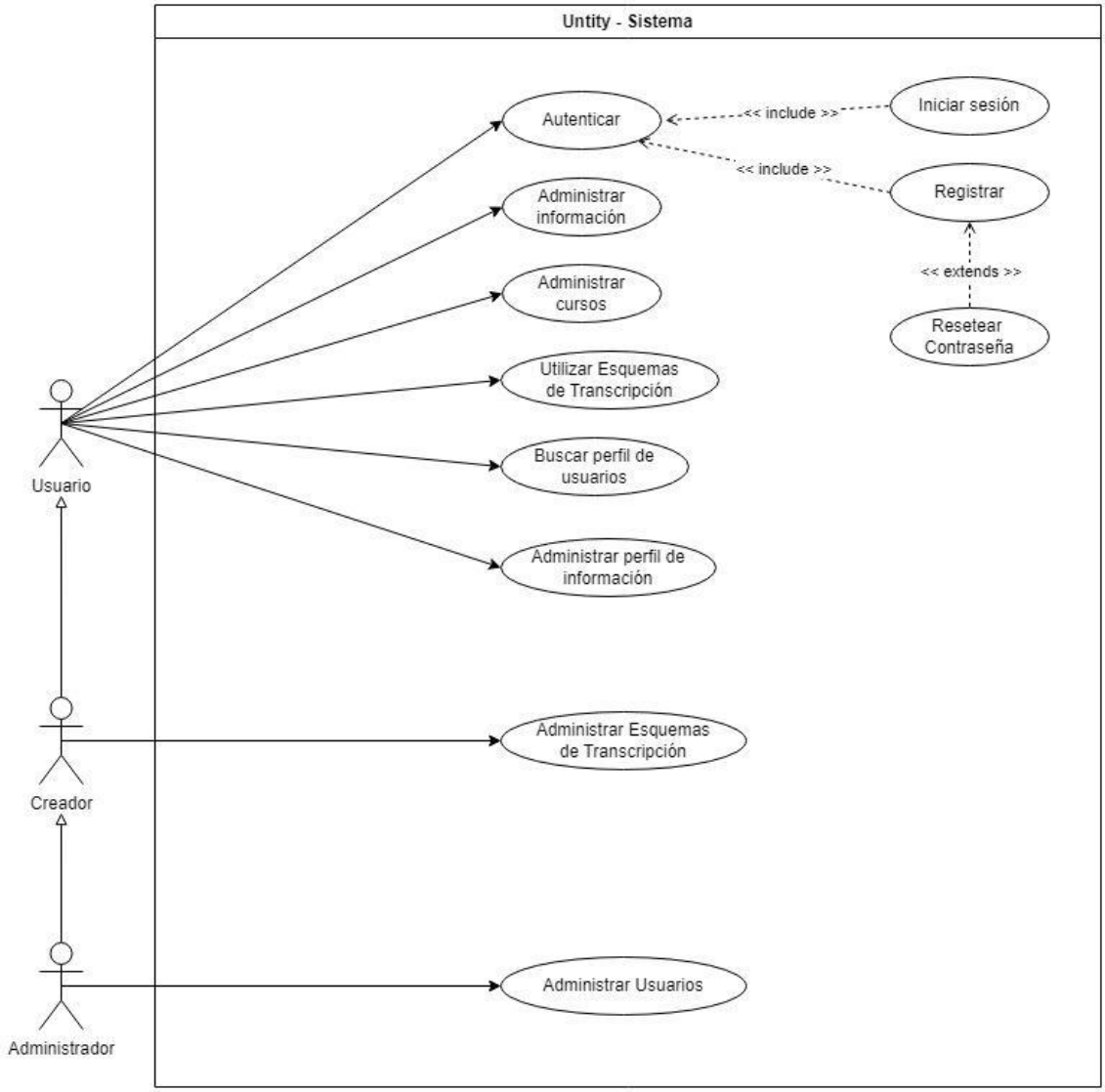


Figura 1. Roles y funcionalidades (Casos de usos).

2.3 Características de los usuarios

Tabla 7. Características del rol usuario

Usuario Nº 1	
Tipo de usuario	Usuario General
Formación	Estudiante, Docente o Personal Administrativo
Habilidades	Conocimientos básicos de informática
Actividades	Autenticar Administrar información Administrar cursos Utilizar Esquema de Transcripción Buscar usuarios Publicar perfil

Tabla 8. Características del rol creador

Usuario Nº 2	
Tipo de usuario	Creador
Formación	Docente o Personal Administrativo
Habilidades	Conocimiento sobre el framework Hyperledger Indy, además de conocer el funcionamiento de la Blockchain.
Actividades	Las mismas que el usuario Administrar Esquemas de Transcripción

Tabla 9. Características del rol administrador

Usuario Nº 3	
Tipo de usuario	Administrador
Formación	Docente o Personal Administrativo
Habilidades	Las mismas que el creador, además de saber manejo de usuarios
Actividades	Las mismas que el creador Administrar usuarios

2.4 Restricciones

Las restricciones que tendrá el sistema son las siguientes:

- Se utilizará Vue.js para el desarrollo del módulo “Frontend”, Node.js para el desarrollo del módulo “Middleware” y Flask para el desarrollo del módulo de “Backend”.
- Se utilizará el framework “Hyperledger Indy” para generar la identidad digital de los usuarios por medio de la Blockchain.
- El sistema solo podrá ser utilizado con conexión a internet.
- El sistema podrá ser usado en cualquier navegador, se recomienda utilizar Chrome, Mozilla o Edge.
- El sistema deberá tener una interfaz responsiva y amigable para los usuarios que interactúen con el sistema.
- Únicamente el rol “usuario” podrá modificar su información personal de manera segura, siendo el responsable de la misma.
- Únicamente los roles “creador” y “administrador” podrán administrar los Esquemas de Transcripción.
- El rol “usuario” podrá utilizar los Esquemas de Transcripción para enviar información por medio de la Blockchain hacia otros usuarios.

2.5 Suposiciones y dependencias

Las suposiciones y dependencias que se tendrá al desarrollar el sistema son las siguientes:

- Se asume que los requisitos descritos en el presente documento son estables.
- Para la interacción de los usuarios con los Esquemas de Transcripción, estos deberán estar registrados en el dominio de la wallet del gobierno.

3 Requisitos específicos

3.1 Requisitos comunes de las interfaces

3.1.1 Interfaces de usuario

Las interfaces de usuario del sistema tendrán un conjunto de ventanas, campos de texto, botones, menús de navegación (a excepción del login), listas, y un sinnúmero de componentes elementales para su interacción. Estas interfaces serán construidas específicamente para la Dapp que se propuso en el Proyecto de Integración Curricular, y podrá visualizarse desde un navegador web.

3.1.2 Interfaces de hardware

Será necesario disponer de equipos de cómputo en perfecto estado con las siguientes características:

- Dispositivo electrónico (computadora, teléfono inteligente, tablet, etc).
- Conexión a internet.

3.1.3 Interfaces de software

- Navegador Web: Chrome, Mozilla Firefox, Opera, Edge u otros.

3.1.4 Interfaces de comunicación

Los módulos del sistema se comunicarán entre sí, mediante el protocolo de comunicación HTTP. Mientras que las operaciones entre los nodos de la Blockchain de Hyperledger Indy, utilizarán el protocolo de consenso RBFT.

3.2 Requisitos funcionales

Tabla 10. Requisito funcional Autenticar

Identificación del requerimiento	RF 01
Nombre de requisito	Autenticar
Descripción	El sistema permitirá autenticarse al usuario
Características	El sistema le permitirá al usuario iniciar sesión, registrarse o recuperar la contraseña
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05 - RNF 07
Prioridad del requisito	Alta

Tabla 11. Requisito funcional Iniciar sesión

Identificación del requerimiento	RF 02
Nombre de requisito	Iniciar sesión
Descripción	El sistema permitirá al usuario iniciar sesión
Características	El sistema le permitirá al usuario iniciar sesión, mostrando mensajes de respuesta en cada interacción
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05 - RNF 07
Prioridad del requisito	Alta

Tabla 12. Requisito funcional Registrar

Identificación del requerimiento	RF 03
Nombre de requisito	Registrar
Descripción	El sistema permitirá al usuario registrarse
Características	El sistema le permitirá al usuario registrarse, para ello debe tener una cuenta en la página de la Carrera de Computación de la UNL. El usuario validará su correo y dependiendo si está registrado se emitirá un mensaje con el link para el registro a su correo.
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05 - RNF 07
Prioridad del requisito	Alta

Tabla 13. Requisito funcional Resetear contraseña

Identificación del requerimiento	RF 04
Nombre de requisito	Resetear contraseña
Descripción	El sistema permitirá al usuario resetear su contraseña
Características	El sistema le permitirá al usuario resetear su contraseña en caso de haberla olvidado, para ello el sistema le enviará al correo un link para el reseteo
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05 - RNF 07
Prioridad del requisito	Alta

Tabla 14. Requisito funcional Administrar información

Identificación del requerimiento	RF 05
Nombre de requisito	Administrar información
Descripción	El sistema permitirá al usuario administrar su información
Características	El sistema le permitirá al usuario administrar su información, ya sea añadiendo, modificando, eliminando o cambiando la visibilidad de su información, siendo almacenada en la Wallet.
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05 - RNF 07
Prioridad del requisito	Alta

Tabla 15. Requisito funcional Administrar cursos

Identificación del requerimiento	RF 06
Nombre de requisito	Administrar cursos
Descripción	El sistema permitirá al usuario administrar sus cursos
Características	El sistema le permitirá al usuario administrar sus cursos obtenidos de la página de la Carrera de Computación de la UNL. El usuario obtendrá un certificado, generado por el sistema, si ha cumplido con el 100% del curso, además únicamente podrá cambiar la visibilidad de sus cursos, siendo almacenados en su Wallet.

Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05 - RNF 07
Prioridad del requisito	Alta

Tabla 16. Requisito funcional Utilizar Esquemas de Transcripción

Identificación del requerimiento	RF 07
Nombre de requisito	Utilizar Esquemas de Transcripción
Descripción	El sistema permitirá al usuario utilizar Esquemas de Transcripción
Características	El sistema le permitirá al usuario utilizar Esquemas de Transcripción, estos Esquemas de Transcripción permitirán registrar en la Blockchain la información que comparte con otro usuario. Cada Esquema de Transcripción estará compuesto por campos que deberán ser llenados por el usuario destinatario hacia el solicitante.
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05 - RNF 06 - RNF 07
Prioridad del requisito	Alta

Tabla 17. Requisito funcional Buscar usuarios

Identificación del requerimiento	RF 08
Nombre de requisito	Buscar perfil de usuarios
Descripción	El sistema permitirá al usuario buscar información de otros usuarios, siendo expuesta si tienen dicho perfil de información público.
Características	El sistema le permitirá al usuario buscar otros perfiles de información de usuarios, la búsqueda dará como resultado el perfil del usuario objetivo si este tiene público su perfil, en caso de tener el perfil en privado este no aparecerá.
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05
Prioridad del requisito	Alta

Tabla 18. Requisito funcional Publicar perfil

Identificación del requerimiento	RF 09
Nombre de requisito	Administrador perfil de información

Descripción	El sistema permitirá al usuario administrador su perfil de información
Características	El sistema le permitirá al usuario hacer visible o no su perfil de información. El perfil de información estará compuesto por información personal (RF 05) y por los cursos (RF 06) que tengan su visibilidad activada. El perfil podrá ser público (otros usuarios podrán ver la información expuesta) y privado (ningún usuario podrá ver la información expuesta)
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05 - RNF 07
Prioridad del requisito	Alta

Tabla 19. Requisito funcional Administrar Esquemas de Transcripción

Identificación del requerimiento	RF 10
Nombre de requisito	Administrar Esquemas de Transcripción
Descripción	El sistema permitirá al rol creador administrar Esquemas de Transcripción
Características	El sistema le permitirá al rol creador administrar Esquemas de Transcripción, ya sea creando, actualizando, eliminando o cambiando su visibilidad.
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05 - RNF 06
Prioridad del requisito	Alta

Tabla 20. Requisito funcional Administrar usuarios

Identificación del requerimiento	RF 11
Nombre de requisito	Administrar usuarios
Descripción	El sistema permitirá al rol administrador administrar a los demás usuarios
Características	El sistema le permitirá al rol administrador activar o desactivar cuentas en caso de que la cuenta objetivo haya utilizado el sistema para perjudicar a otros usuarios o personas, y también asignará roles a los usuarios en caso de ser necesario.
Requerimiento no Funcional	RNF 01- RNF 02 - RNF 03 - RNF 04 - RNF 05
Prioridad del requisito	Alta

3.3 Requisitos no funcionales

Tabla 28. Usabilidad

Identificación del requerimiento	RNF 01
Nombre de requisito	Usabilidad
Descripción	El sistema tendrá una interfaz amigable, responsiva y sencilla para lograr una interacción amena con los usuarios.
Prioridad del requisito	Alta

Tabla 29. Seguridad

Identificación del requerimiento	RNF 02
Nombre de requisito	Seguridad
Descripción	El sistema utilizará tokens para el envío de información y la información sensible será almacenada en la wallet de cada usuario, dentro de la Blockchain, para mantener su confidencialidad e integridad.
Prioridad del requisito	Alta

Tabla 30. Fiabilidad

Identificación del requerimiento	RNF 03
Nombre de requisito	Fiabilidad
Descripción	La información de los usuarios será almacenada en la Blockchain, por tal motivo la información no podrá ser manipulada por usuarios ajenos a ella.
Prioridad del requisito	Alta

Tabla 31. Tiempo de respuesta

Identificación del requerimiento	RNF 04
Nombre de requisito	Tiempo de respuesta
Descripción	El sistema responderá a las solicitudes de los usuarios en un tiempo adecuado y eficiente, de 3 a 10 segundos máximo, siempre y cuando la conexión a internet sea estable (banda ancha de 10 Mbps). Este tiempo se prolonga mayormente por los servicios de la Blockchain, en caso de que la

	petición no ocupe dichos servicios se responderá en un tiempo mínimo.
Prioridad del requisito	Alta

Tabla 32. Disponibilidad

Identificación del requerimiento	RNF 05
Nombre de requisito	Disponibilidad
Descripción	El sistema funcionará en lo posible durante las 24 horas del día, durante los 7 días de la semana, siempre y cuando la infraestructura de los servidores y la red de Blockchain esté en condiciones óptimas. Para las actualizaciones del sistema, no habrá disponibilidad hasta que sean completadas.
Prioridad del requisito	Alta

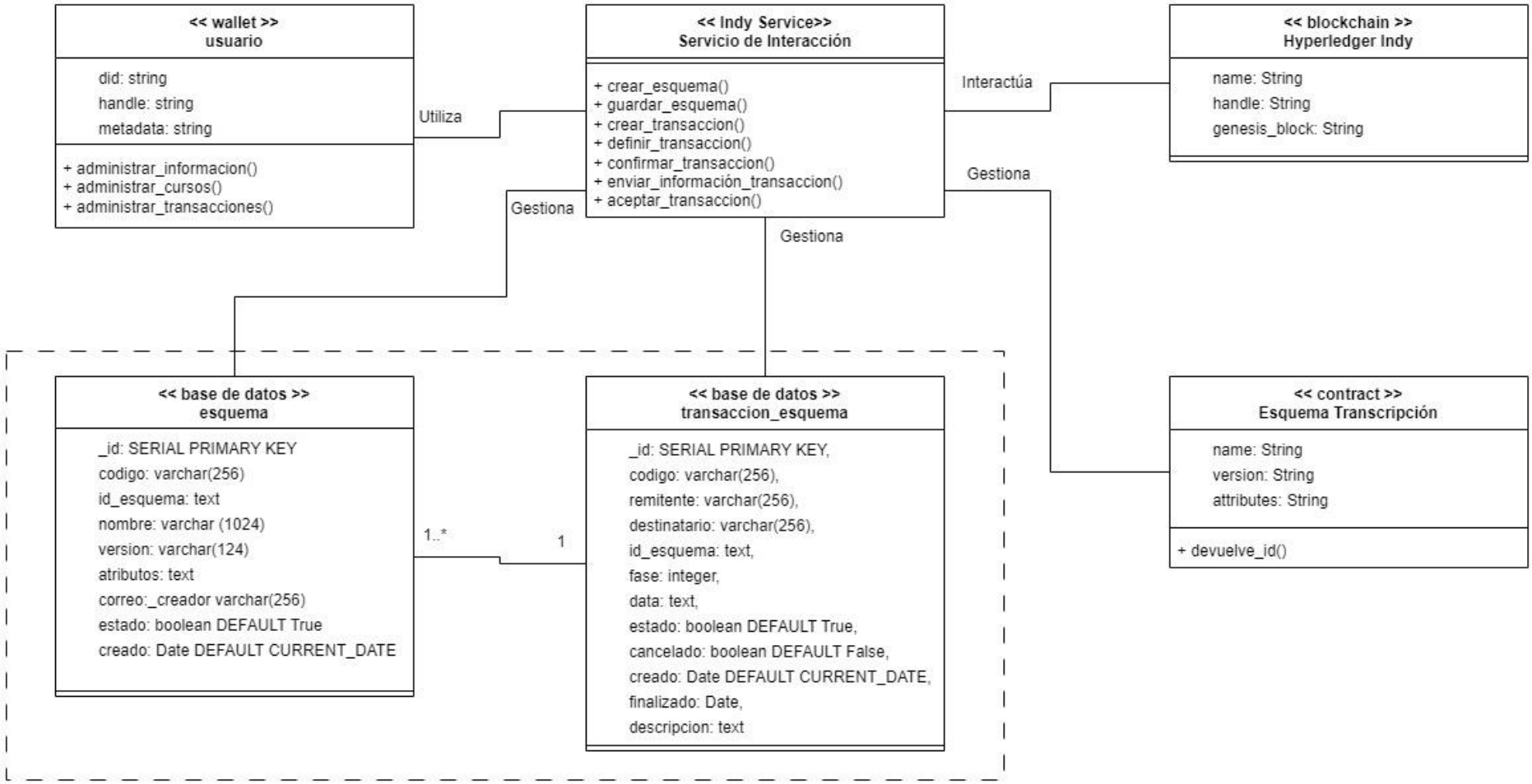
Tabla 33. Trazabilidad

Identificación del requerimiento	RNF 06
Nombre de requisito	Trazabilidad
Descripción	El sistema guardará la información pertinente de la utilización de los Esquemas de Transcripción en la base de datos y Blockchain, quedando como constancia de su uso.
Prioridad del requisito	Alta

Tabla 34. Confidencialidad

Identificación del requerimiento	RNF 07
Nombre de requisito	Confidencialidad
Descripción	El sistema al ser construido en base a la Blockchain de Hyperledger Indy, toda la información de los usuarios se mantendrá encriptada (algoritmo ChaCha20-Poly1305 con HMAC-256) y no podrá ser modificada si no es el mismo usuario.
Prioridad del requisito	Alta

Anexo 2: Diagrama de Clases del módulo “Back-end”.



Anexo 3: Documentación de Codificación del Subsistema de Contratos Inteligente.

Documentación de Codificación del Subsistema de Contratos Inteligentes

Proyecto: Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja

Autor: Alexis Armijos
Correo electrónico: carlos.a.armijos@unl.edu

1. Introducción

Este documento es el Documentación de Codificación del Subsistema de Contratos Inteligentes para el desarrollo del proyecto de una propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja. Este documento muestra lo más importante en la codificación del contrato inteligente denominado "**Esquema de Transcripción**". El cual permite, a los usuarios de la blockchain de Hyperledger Indy, el intercambio de información siguiendo un proceso de transacción en fases.

1.1. **Propósito**

El presente documento tiene como propósito mostrar la codificación de los contratos inteligentes para poder desarrollar la propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja.

1.2. **Alcance**

Este documento va dirigido al personal involucrado en el desarrollo del presente Proyecto de Integración Curricular y a lectores que pretendan recrear el contrato inteligente.

2. Codificación

2.1. Indicaciones generales

Para la codificación del contrato inteligente se siguió la guía proporcionada por Hyperledger Indy (ver https://github.com/hyperledger/indy-sdk/blob/main/docs/getting-started/indy-walkthrough.md). Entre los diferentes lenguajes de programación disponibles para su implementación, se decidió elegir la implementación mediante Python por la familiaridad que se tiene con el lenguaje.

2.2. Diseño del contrato inteligente

El esquema de transcripción tiene una estructura compuesta por su nombre, la versión (por defecto es “1.0”) y por los atributos que tendrá. Los atributos del esquema son la información que se deberá completar para que el esquema peda ser enviado hacia el usuario interesado en la información.

Un ejemplo sería un esquema de transcripción para la “Obtención de certificado por haber participado en algún curso”, dónde los atributos podrían ser la cédula del usuario interesado, el nombre del curso, entre otros atributos pertinentes. En la Figura 1 se muestra el diseño del contrato inteligente.

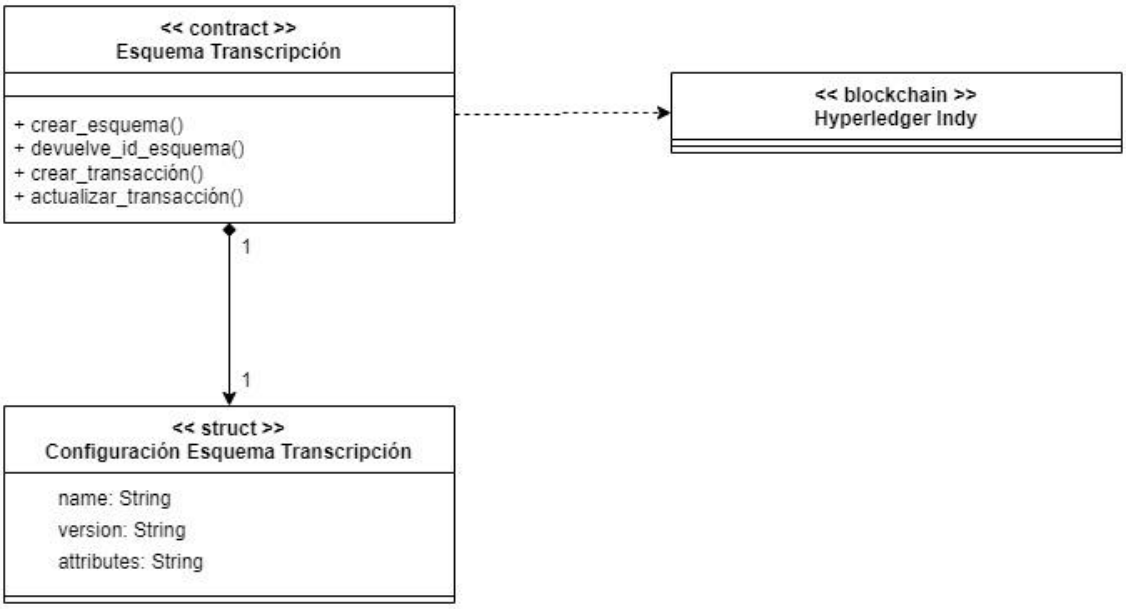


Figura 1: *Diseño del contrato inteligente.*

2.3. Codificación

En este punto se mostrarán la codificación para configurar la blockchain, wallets y esquema de transcripción en un ambiente sencillo fuera de la aplicación original.

En la Tabla 1, se muestra la configuración que tiene la blockchain, de Hyperledger Indy, denominada Blockchain-Computación, esta red descentralizada se levanta sobre el bloque génesis guardado en el archivo “pool.txn”. El protocolo que utiliza es el 2, ya que este es para las versiones mayores a la 1.4 de la librería indy-node

Una vez configurada la blockchain, se utiliza la librería “pool” para crearla. Si es la primera vez que se inicia no ocurrirá ningún error, en caso de no ser la primera vez dará error porque ya existe una blockchain con ese nombre. No obstante, al ser creada o haber aparecido el error, la “pool” abrirá la blockchain devolviendo el identificador que servirá para interactuar con ella.

Tabla 1: *configuración de la blockchain de Hyperledger Indy.*

```
from indy import anoncreds, did, ledger, pool, wallet
from indy.error import ErrorCode, IndyError

async def run():
    logger.info("Iniciando")

    pool_ = {
        'name': 'pool-UNL-prueba'
    }
    logger.info("Abre la Pool del Ledger: {}".format(pool_['name']))
    pool_['genesis_txn_path'] = "pool.txn"
    pool_['config'] = json.dumps(
        {'genesis_txn': str(pool_['genesis_txn_path'])})

    # Set protocol version 2 to work with Indy Node 1.4
    await pool.set_protocol_version(2)

    try:
        await pool.create_pool_ledger_config(pool_['name'], pool_['config'])
    except IndyError as ex:
        if ex.error_code == ErrorCode.PoolLedgerConfigAlreadyExistsError:
            pass
    pool_['handle'] = await pool.open_pool_ledger(pool_['name'], None)
```

Para crear las wallets dentro de la blockchain de Hyperledger Indy, primero se configuran con su respectivo nombre, la configuración y credenciales que tendrán (se recomienda utilizar claves no repetibles), el pool haciendo referencia al identificador de la blockchain abierta, la semilla que tendrá (únicamente lo requiere la wallet Gobierno) y el rol (lo tendrán todas las wallets menos el Gobierno), esta configuración se puede ver en la Tabla 2.

Tabla 2: *configuración y creación de las wallets para el Gobierno y Computación.*

```
# Configuración de la Wallet del Gobierno
    gobierno = {
        'name': "Gobierno",
        'wallet_config': json.dumps({'id': 'gobierno_wallet'}),
        'wallet_credentials': json.dumps({'key': 'gobierno_key'}),
        'pool': pool_['handle'],
        'seed': '000000000000000000000000Steward1'
    }

    # Gobierno -> Crea su wallet
    await create_wallet(gobierno)

    # Gobierno -> Crea y almacena en su Wallet su DID
    gobierno['did_info'] = json.dumps({'seed': gobierno['seed']})
    gobierno['did'], gobierno['key'] = await did.create_and_store_my_did(gobierno['wallet'],
gobierno['did_info'])

    # Configuración de la Wallet del Computación
    computacion = {
        'name': 'Computacion',
        'wallet_config': json.dumps({'id': 'computacion_wallet'}),
        'wallet_credentials': json.dumps({'key': 'computacion_wallet_key'}),
        'pool': pool_['handle'],
        'role': 'TRUST_ANCHOR'
    }

    # Computación -> Crea su wallet
    await create_wallet(computacion)

    # Computacipón -> Crea y almacena en su Wallet su DID
    (computacion['did'], computacion['key']) = await
did.create_and_store_my_did(computacion['wallet'], "{}")

    # Computación -> Se registra bajo el domino del gobierno
    await getting_verinym(gobierno, computacion)
```

Al ser configuradas las wallets, son creadas utilizando el método “create_wallet()”, las wallets con el rol de “TRUST_ANCHOR” son registradas bajo el dominio del Gobierno con el método “getting_verinym()”. Al estar en el dominio del Gobierno les permitirá interactuar con los esquemas de transcripción registrados en la blockchain.

En la Tabla 3 se muestran los métodos utilizados para la creación de las wallets del Gobierno y de Computación.

Tabla 3: *métodos utilizados para la configuración de las wallets.*

```
async def create_wallet(identity):
    logger.info("\"{}\" -> Create wallet".format(identity['name']))
    try:
        await wallet.create_wallet(identity['wallet_config'],
                                   identity['wallet_credentials'])
    except IndyError as ex:
        if ex.error_code == ErrorCode.PoolLedgerConfigAlreadyExistsError:
            pass
    identity['wallet'] = await wallet.open_wallet(identity['wallet_config'],
                                          identity['wallet_credentials'])

async def getting_verinym(from_, to):
    from_['info'] = {
        'did': to['did'],
        'verkey': to['key'],
        'role': to['role'] or None
    }

    await send_nym(from_['pool'], from_['wallet'], from_['did'], from_['info']['did'],
                   from_['info']['verkey'], from_['info']['role'])

async def send_nym(pool_handle, wallet_handle, _did, new_did, new_key, role):
    nym_request = await ledger.build_nym_request(_did, new_did, new_key, None, role)
    await ledger.sign_and_submit_request(pool_handle, wallet_handle, _did, nym_request)
```

Las wallets que tienen el rol “TRUST_ANCHOR” serán las únicas capaces de crear y registrar los esquemas de transcripción. La configuración del esquema de transcripción está compuesta por 3 valores:

- ‘**name’**: el nombre que tendrá el esquema de transcripción.
- ‘**version’**: se recomienda empezar en “1.0”.
- ‘**attributes’**: es una lista con todos los atributos que tiene el esquema de transcripción a crear.

Al tener la configuración del esquema de transcripción, Computación creará el esquema utilizando la librería “anoncreds” con su método “issuer_create_schema()” dando como resultado el identificador y estructura del esquema válidos. Por último, la estructura válida es firmada y enviada a la blockchain mediando la librería “ledger” utilizando el método “sign_and_submit_request()”. Este proceso se puede ver en la Tabla 4.

Tabla 4: *configuración y creación del esquema de transcripción.*

```
    # Computacion -> Configura el Esquema de Transcripción
    transcript = {
        'name': 'Esquema_Cursos',
        'version': '1.2',
        'attributes': ['cedula', 'nombre del curso', "tiempo de validez"]
    }

    # Computacion -> Crea Esquema de Transcripción
    (computacion['transcript_schema_id'], computacion['transcript_schema']) = \
        await anoncreds.issuer_create_schema(computacion['did'], transcript['name'],
transcript['version'],
                                             json.dumps(transcript['attributes']))
    transcript_schema_id = computacion['transcript_schema_id']

    # Computacion -> Envía el Esquema de Transcripción hacia el Ledger (Blockchain)
    await send_schema(computacion['pool'], computacion['wallet'], computacion['did'],
computacion['transcript_schema'])
```

Para ejemplificar el proceso de transacción se crearon dos wallets, Alexis siendo el usuario interesado y Esther siendo el usuario objetivo, estos dos participantes muestran el proceso de transacción utilizando el esquema de transcripción creado en la Figura 5.

El proceso de transacción estará compuesto por 4 fases fundamentales, descritas a continuación:

- La primera fase se denomina “**Definición**”, el objetivo es que Esther cree la definición de credenciales de transcripción para Alexis utilizando la librería “anoncreds” con el método “issuer_create_and_store_credential_def()”. Al haber creado la definición, también se creará la oferta de credencial de transcripción con ayuda de la librería “anoncreds” mediante el método “issuer_create_credential_offer()” siendo importante para la siguiente fase. Esta definición es el contrato que permitirá el envío de información entre estos dos usuarios, ver la Tabla 5.

Tabla 5: *configuración y ejecución de la fase 1, "Definición".*

```
    # Agentes en el Proceso de Transacción
    # Esther (Usuario Objetivo)
    # Alexis (Usuario Interesado)

    # Paso 1: Esther crea la Definición de las Credenciales de Transcripción para la petición
de Alexis

    logger.info("--- Creaciòn de la Definiciòn de Credenciales para Alexis de Esther ---")

    logger.info("Esther -> Obtiene Esquema de Trancripción desde el Ledger")
    (esther['transcript_schema_id'], esther['transcript_schema']) = await get_schema(
        esther['pool'], esther['did'], transcript_schema_id)

    datos_esquema = json.loads(esther['transcript_schema'])

    logger.info("Esther -> Crea la Definición de las Credenciales de Transcripción")
    transcript_cred_def = {
        'tag': datos_esquema['name'] + alexis['name'],
        'type': 'CL',
        'config': {"support_revocation": False}
    }
    (esther['transcript_cred_def_id'],
     esther['transcript_cred_def']) = await anoncreds.issuer_create_and_store_credential_def(
        esther['wallet'], esther['did'], esther['transcript_schema'], transcript_cred_def['tag'],
        transcript_cred_def['type'], json.dumps(transcript_cred_def['config']))

    logger.info("Esther -> Envía  la Definición de las Credenciales de Transcripción al Ledger")
    await send_cred_def(esther['pool'], esther['wallet'], esther['did'],
esther['transcript_cred_def'])

    logger.info("Esther -> Crea la Oferta de Credencial de Trasncripción para Alexis")
    esther['transcript_cred_offer'] = \
        await anoncreds.issuer_create_credential_offer(esther['wallet'],
esther['transcript_cred_def_id'])

    logger.info("Esther -> Envía la Oferta de Credencial de Trasncripción a Alexis")
    alexis['transcript_cred_offer'] = esther['transcript_cred_offer']
    transcript_cred_offer_object = json.loads(alexis['transcript_cred_offer'])
```

- La segunda fase denominada "**Confirmación**", el objetivo es que Alexis confirme el contrato definido por Esther. Para ello se crea la solicitud de credencial de transcripción, en base a la oferta de la fase anterior, mediante el método "prover_create_credential_req()" de la librería "anoncreds". Ver en la Tabla 6 la ejecución de esta fase.

Tabla 6: *ejecución de la fase 2, "Confirmación".*

```
    # Paso 2: Alexis Crea la Solicitud de Credencial de Transcripción para Esther

    alexis['transcript_schema_id'] = transcript_cred_offer_object['schema_id']
    alexis['transcript_cred_def_id'] = transcript_cred_offer_object['cred_def_id']

    logger.info("Alexis -> Crea y almacena la Master Secret en la Wallet")
    alexis['master_secret_id'] = await anoncreds.prover_create_master_secret(alexis['wallet'],
None)

    logger.info("Alexis -> Obtiene la Definición de Credencial de Transcripción (Computación)
del Ledger")
    (alexis['esther_transcript_cred_def_id'], alexis['esther_transcript_cred_def']) = await
get_cred_def(
        alexis['pool'], alexis['did'], alexis['transcript_cred_def_id'])

    logger.info("Alexis -> Crea la Solicitud de Credencial de Transcripción para Esther")
    (alexis['transcript_cred_request'],
     alexis['transcript_cred_request_metadata']) = await
anoncreds.prover_create_credential_req(
        alexis['wallet'], alexis['did'], alexis['transcript_cred_offer'],
alexis['esther_transcript_cred_def'],
        alexis['master_secret_id'])

    logger.info("Alexis -> Envía la Solicitud de Credencial de Transcripción para Esther")
    esther['transcript_cred_request'] = alexis['transcript_cred_request']
```

- La tercera fase denominada "**Envío**", el objetivo es que Esther complete los atributos del esquema de transcripción con la información adecuada. Para completar esta fase primero se configura la estructura de los atributos del esquema con sus respectivos valores de información, y con el método "issuer_create_credential()" de la librería "anoncreds" se crea la credencial de transcripción con la información proporcionada de Esther. Esta credencial es envida a Alexis, ver la Tabla 7.

Tabla 7: *configuración y ejecución de la fase 3, "Envío".*

```
    # Paso 3: Esther completa los datos datos del esquema de transcripción y lo envía a
Alexis

    logger.info(
        "Esther -> Crea la Credencial de Transcripción (Comple el esquema con sus datos)
para Alexis")

    esther['alexis_transcript_cred_values'] = json.dumps({
        "cedula": {"raw": "1900142311", "encoded":
"113948171645748869017221791627810 3335"},
        "nombre del curso": {"raw": "Matemáticas avanzadas", "encoded":
"113948171645748869017221791627810 3335"},
        "tiempo de validez": {"raw": "2 dias", "encoded":
"113948171645748869017221791627810 3335"},
    })

    esther['transcript_cred'], _, _ = await anoncreds.issuer_create_credential(
        esther['wallet'], esther['transcript_cred_offer'],
        esther['transcript_cred_request'],
        esther['alexis_transcript_cred_values'], None, None)

    logger.info("Esther -> Envía la Credencial de Transcripción para Alexis")
    alexis['transcript_cred'] = esther['transcript_cred']
```

- La cuarta fase denominada "**Aceptación**", el objetivo es que Alexis almacene la información proporcionada por Esther en su wallet. Con ayuda de la librería "anoncreds" y su método "prover_store_credential()" se consigue registrar en la wallet la información obtenida, ver la Tabla 8.

Tabla 8: *ejecución de la fase 4, "Aceptación".*

```
    # Paso 4: Alexis guarda la información de Esther

    logger.info("Alexis -> Almacena la Credencial de Transcripción de Esther")
    _, alexis['transcript_cred_def'] = await get_cred_def(alexis['pool'], alexis['did'],
                                            alexis['transcript_cred_def_id'])

    await anoncreds.prover_store_credential(alexis['wallet'], None,
alexis['transcript_cred_request_metadata'],
                                    alexis['transcript_cred'], alexis['transcript_cred_def'], None)
```

A continuación, en la Tabla 9 se muestran los métodos personalizados utilizados durante este proceso de transacción.

Tabla 9: *métodos personalizados del proceso de transacción.*

```
async def send_cred_def(pool_handle, wallet_handle, _did, cred_def_json):
    cred_def_request = await ledger.build_cred_def_request(_did, cred_def_json)
    await ledger.sign_and_submit_request(pool_handle, wallet_handle, _did,
cred_def_request)

async def get_cred_def(pool_handle, _did, cred_def_id):
    get_cred_def_request = await ledger.build_get_cred_def_request(_did, cred_def_id)
    get_cred_def_response = await ensure_previous_request_applied(
        pool_handle, get_cred_def_request, lambda response: response['result']['data'] is not
None)
    return await ledger.parse_get_cred_def_response(get_cred_def_response)

async def ensure_previous_request_applied(pool_handle, checker_request, checker):
    for _ in range(3):
        response = json.loads(await ledger.submit_request(pool_handle, checker_request))
        try:
            if checker(response):
                return json.dumps(response)
        except TypeError:
            pass
        time.sleep(5)
```

Anexo 4: Plan de Pruebas Unitarias para el Subsistema de Contratos Inteligentes.

Plan de Pruebas Unitarias para el Subsistema de Contratos Inteligentes

Proyecto: Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja

Versión: 2.0

Fecha: 1/8/2023

Autor: Alexis Armijos
Correo electrónico: carlos.a.armijos@unl.edu

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja		
Entregable	Planes de Pruebas Unitarias para Subsistema de Contratos Inteligentes		
Autor	Alexis Armijos		
Versión/Edición	2.0	**Fecha Versión**	31/07/2023
Aprobado por	Cristian Ramiro Narváez Guillen, Mg. Sc.	**Fecha Aprobación**	1/8/2023
		N.º Total de Páginas	8

Registro de cambios

Versión doc.	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas Unitarias subsistema de Contratos Inteligentes	Carlos Alexis Armijos Rios	31/07/2023
2.0	Versión final del Plan de Pruebas Unitarias subsistema de Contratos Inteligentes	Carlos Alexis Armijos Rios	1/8/2023

Control de distribución

Nombre y Apellidos
Carlos Alexis Armijos Rios
Cristian Ramiro Narváez Guillen, Mg. Sc

Introducción

Objetivo

El objetivo de este documento es verificar la funcionalidad correcta del subsistema de Contratos Inteligentes aislando cada parte del código principal en bloques, según su propósito y mostrando que dichos bloques son correctos. Esto respecto al código principal que se ejecuta en el módulo “Back-end”.

Propósito

Comprobar el correcto funcionamiento del subsistema de Contratos Inteligentes, validando los bloques de código obtenidos al separar el código principal del módulo, para asegurar que cada uno funcione correctamente y eficientemente por separado.

Definición de los casos de pruebas

En esta parte se describen cada uno de los casos de prueba necesarios para verificar la funcionalidad completa de los contratos inteligentes.

Tabla resumen de todos los casos de prueba:

Nro del Caso de Prueba	Componente	Descripción de lo que se probará	Prerrequisitos
CP-01	Wallet	Crear la wallet del usuario, y verificar que se obtiene el identificador de la wallet	N/A
CP-02	Wallet	Crear y almacenar el DID para la wallet del usuario	Wallet creada
CP-03	Wallet	Registrar la wallet del usuario bajo el dominio del Gobierno	Wallet creada
CP-04	Chain code	Crear el Esquema de Transcripción	Wallet creada, bajo domino del Gobierno
CP-05	Chain code	Firmar y enviar el esquema de transcripción a la blockchain	Esquema configurado
CP-06	Chain code	Crear y enviar la Oferta de Credencial de Transcripción	Esquema creado correctamente
CP-07	Chain code	Crear y enviar la Solicitud de Credencial de Transcripción	Oferta de Credencial de Transcripción creada correctamente
CP-08	Chain code	Crear y enviar la Credencial de Transcripción	Solicitud de Credencial de Transcripción creada correctamente
CP-09	Chain code	Almacenar la Credencial de Transcripción en la wallet del usuario	Credencial de Transcripción creada correctamente

Casos de prueba a detalle:

CP-01					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Crear la wallet del usuario, y verificar que se obtiene el identificador de la wallet	create_wallet()	configuracion_wallet	✓	N/A

CP-02					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
2	Crear y almacenar el DID para la wallet del usuario	create_and_store_my_did()	identificador_wallet, configuracion_did	✓	N/A

CP-03					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
3	Registrar la wallet del usuario bajo el dominio del Gobierno	Getting_verinym()	wallet_gobierno, wallet_usuario	✓	N/A

CP-04					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
4	Crear el Esquema de Transcripción	issuer_create_schema()	did_usuario_wallet, nombre_esquema, version_esquema, atributos_esquema	✓	N/A

CP-05					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
5	Firmar y enviar el esquema a la blockchain	send_schema()	identificador_blockchain, identificador_usuario_wallet, did_usuario_wallet, esquema	✓	N/A

CP-06					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
6	**Transacción, fase (1):** Crear y enviar la Oferta de Credencial de Transcripción	issuer_create_credential_offer()	identificador_usuario_wallet, transcript_cred_def_id	✓	N/A

CP-07					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
7	**Transacción, fase (2):** Crear y enviar la Solicitud de Credencial de Transcripción	prover_create_credential_req()	identificador_usuario_wallet, did_usuario_wallet, transcript_cred_offer, transcript_cred_def	✓	N/A

CP-08

N.º	Descripción	Método	Datos Entrada	¿OK?	Observaciones
8	**Transacción, fase (3):** Crear y enviar la Credencial de Transcripción	issuer_create_credential()	identificador_usuario_wallet, transcript_cred_offer, transcript_cred_request, transcript_cred_def_values	✓	N/A

CP-09					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
9	**Transacción, fase (4):** Almacenar la Credencial de Transcripción en la wallet del usuario	prover_store_credential()"	identificador_usuario_wallet, None, transcript_cred_request_metadata transcript_cred, transcript_cred_def, None	✓	N/A

Glosario

A continuación, se muestra la definición de todos los términos utilizados en el presente documento.

Término	Descripción
Esquema de Transcripción	Es el contrato inteligente que permite a los usuarios intercambiar información a través de la blockchain de Hyperledger Indy.
Wallet	Es la billetera virtual que tiene el usuario de la blockchain de Hyperledger Indy, dentro de ella podrá manipular su propia información.
DID	Identificadores descentralizados asignados a cada wallet.
Chain Code	Es la codificación que permite interactuar con la blockchain, en este caso mediante las librerías y métodos proporcionados por Hyperledger Indy

Anexo 5: Documentación de Codificación del Subsistema de Aplicación.

Documentación de Codificación del Subsistema de Aplicación

Proyecto: Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja

Autor: Alexis Armijos
Correo electrónico: carlos.a.armijos@unl.edu

1. Introducción

Este documento es el Documentación de Codificación del Subsistema de Aplicación para el desarrollo del proyecto de una propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja. Se muestra lo más importante en la codificación de la aplicación, siendo esto las funcionalidades de: tokens dinámicos para la interacción de los usuarios con el sistema, envío de correos (para registrarse y resetear contraseña), la obtención de los usuarios de la página de la Carrera de Computación, el usuario actualizando su información personal, de cursos o transacciones, y, por último, las fases del proceso de transacción.

1.3. **Propósito**

El presente documento tiene como propósito mostrar la codificación de las principales funcionalidades del subsistema de aplicación para el desarrollo de la propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja.

1.4. **Alcance**

Este documento va dirigido al personal involucrado en el desarrollo del presente Proyecto de Integración Curricular y a lectores que pretendan recrear el subsistema de aplicación.

2. Codificación

En este punto se mostrarán la codificación de las principales funcionales del subsistema de aplicación, las cuales son fundamentales para lograr un desempeño eficiente y eficaz, y también para conseguir una seguridad alta en la interacción con el sistema.

2.1. Tokens Dinámicos

En la Tabla 1 se muestra la codificación para crear los tokens dinámicos en cada inicio de sesión en el sistema. Esto permite que la última sesión autenticada correctamente en el sistema tendrá el token válido para realizar cualquier acción, mientras que las otras sesiones más antiguas no podrán realizar ninguna acción porque su token no es válido.

Tabla 1: *codificación del método que genera tokens dinámicos.*

```
const jwt = require("jsonwebtoken");

exports.g_token_handle = function (_id) {
  // Crea tokens dinámicos para que el usuario pueda interactuar en el sistema
  let password =
    Math.random() + "-" + process.env.SECRET_KEY_PASSWORD + "-" + Math.random();
  const token = jwt.sign(_id, password);
  return { token, password };
};
```

2.2. Envío de correos

Para enviar correos se hace uso de la librería nodemailer, con la siguiente configuración base, ver Tabla 2.

Tabla 2: *configuración base para enviar correos.*

```
const nodemailer = require("nodemailer");
const config = require("../environment");

const transporter = nodemailer.createTransport({
  service: "gmail",
  auth: {
    user: config.email.user,
    pass: config.email.clave,
  },
});
```

a. Registrarse en el sistema:

En la Tabla 3 se muestra la estructura del correo, compuesta por el token de registro, lo cual permite al usuario acceder a una vista para poder registrarse en el sistema.

Tabla 3: *estructura del correo para registrarse en el sistema.*

```
exports.enviar_email_registro = async function (correo, token) {
  try {
    let mailOptions = {
      from: "Registo en Untity <" + config.email.user + ">",
      to: "<" + correo + ">",
      subject: "Verificación para registro",
      html: `
      <div>
        <center>
          <h1>Untity</h1>
          <p>
            Bienvenido al sistema de identidad digital auto-gestionado de la
            Carrera de Computación de la Universidad Nacional de Loja, Ecuador.
          </p>
          <p>
            Utilizamos tecnología Blockchain para que tú mismo puedas
            controlar tu información sin que otros puedan manipularla.
          </p>
          <a href="${config.frotend.path}/auth/register/${token}" target="_blank">Abrir enlace de
registro</a>
        </center>
        <p><strong>Nota:</strong> Este enlace quedará anulado a las 00:00am. UTC</p>
        <p><strong>*Importante:</strong> Cualquier mal uso del sistema, será tu
responsabilidad.</p>
    </div>`,
    };

    await transporter.sendMail(mailOptions, (error, info) => {
      if (error) throw error;
    });

    return true;
  } catch (error) {
    console.log(
      "\n========== Email_service | ERROR | enviar_email ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
    return false;
  }
```

```
};
```

b. Resetear la contraseña:

En la Tabla 4 se muestra la estructura del correo, compuesta por el token de reseteo, lo cual permite al usuario acceder a una vista para poder resetear su contraseña.

Tabla 4: *estructura del correo para resetear la contraseña.*

```
exports.enviar_email_reset = async function (correo, token) {
  try {
    let mailOptions = {
      from: "Resetear Contraseña en Untity <" + config.email.user + ">",
      to: "<" + correo + ">",
      subject: "Verificación para resetear contraseña",
      html: `
      <div>
        <center>
          <h1>Untity</h1>
          <p>
            Bienvenido al sistema de identidad digital auto-gestionado de la
            Carrera de Computación de la Universidad Nacional de Loja, Ecuador.
          </p>
          <p>
            Vaya, al parecer has olvidado tu contraseña, dale click al siguiente
            enlace para poder cambiar tu contraseña.
          </p>
          <a href="${config.frotend.path}/auth/reset/${token}" target="_blank">Abrir enlace de
registro</a>
        </center>
        <p><strong>Nota:</strong> Este enlace quedará anulado a las 00:00am. UTC</p>
        <p><strong>*Importante:</strong> Cualquier mal uso del sistema, será tu
responsabilidad.</p>
    </div>`,
    };

    await transporter.sendMail(mailOptions, (error, info) => {
      if (error) throw error;
    });

    return true;
  } catch (error) {
    console.log(
      "\n========== Email_service | ERROR | enviar_email ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
```

```
    );
    return false;
  }
};
```

2.3. Obtener usuarios del sistema de la Carrera de Computación

En la Tabla 5 se muestran los métodos utilizados para obtener los usuarios del sistema de la Carrera de Computación. Primero se hace una petición a dicho sistema, devolviendo una respuesta de éxito o de error, dependiendo cual sea la respuesta se podrán obtener los usuarios. Como segundo paso se validan los correos filtrándolos mediante expresiones regulares. Por último, se verifica si los correos obtenidos ya están registrados en el sistema, en caso de no estarlo se crea un nuevo usuario con ese correo.

Tabla 5: *métodos para obtener los usuarios de la Carrera de Computación.*

```
exports.obtener_usuarios_carrera = async function () {
  try {
    let response = await axios.get(path.listar_usuarios);
    if (response.data.tipo == "success") {
      let correos = response.data.data;
      await spy_service.validar_correos_odoo(correos);
    } else {
      console.log(
        "\n ============================== Listar Usuarios Odoo :: Info :: Inicio
================================\n"
      );
      console.log(response.data.mensaje);
      console.log("No hay usuarios");
      console.log(
        "\n ============================== Listar Usuarios Odoo :: Info :: Fin
================================\n"
      );
    }
  } catch (error) {
    if (error.code == "ECONNREFUSED") {
      error = "Sin Conexión a la BD del Back-end...";
    }
    console.log(
      "\n========== Spy_controller | ERROR | listar_usuarios() ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
  }
};
```

2.4. Cookies de usuario

a. Establecer cookies de sesión

En la Tabla 6 se muestra como se establecen las cookies de sesión del usuario cuando inicia sesión en el sistema. Estas cookies permiten la interacción con las funcionalidades, el tiempo de duración es enviando por el módulo “Middleware” al módulo “Front-end”.

Tabla 6: *codificación para establecer las cookies de sesión.*

```
async set_cookie_sesion(token, expiration) {
  try {
    let d_cookie = await helpers.obtener_data_cookie(token);
    let time = this.calc_expiration(expiration);
    await Cookies.set("iu", d_cookie._id, {
      expires: time,
    });
    await Cookies.set("th", d_cookie.token, {
      expires: time,
    });
  } catch (error) {
    console.log();
  }
},
```

b. Borrar cookies de sesión

En la Tabla 7 se muestra la codificación para borrar las cookies de sesión del usuario cuando cierre su sesión en el sistema.

Tabla 7: *codificación para borrar las cookies de sesión.*

```
async cerrar_sesion() {
  await Cookies.remove("iu");
  await Cookies.remove("th");
},
```

2.5. Protección de rutas en las vistas principales

En la Tabla 8 se muestra la codificación proteger el acceso a las vistas sin antes haber iniciado sesión, para ello primero se valida si hay una sesión activa, en caso de no estarlo se lo redirecciona a la vista de iniciar sesión mientas que si está activa irá a la vista de inicio.

Tabla 8: *codificación para proteger el acceso a las vistas principales.*

```
router.beforeEach(async (to, from, next) => {
  let check_user = await cookie.sesion_activa();
  if (to.meta.auth && !check_user) {
    next("/auth/login");
  }

  if (to.meta.pass && check_user) {
    next("/");
  }
  next();
});
```

2.6. Validar el rol del usuario antes de realizar alguna acción

En la Tabla 9 se muestra la codificación para validar el rol del usuario antes de realizar alguna acción en el sistema. Teniendo en cuenta que los 3 roles (usuarios, creador y administrador) tienen un valor numérico asignado, se comparan los valores para validar si el usuario tiene permisos para realizar la acción

Tabla 9: *Codificación para validar el rol del usuario.*

```
exports.validar_permiso_rol = async function (rol_usuario, rol_necesario) {
  const roles = new Map();
  roles.set("usuario", 1);
  roles.set("creador", 2);
  roles.set("administrador", 3);
  try {
    return roles.get(rol_usuario) >= roles.get(rol_necesario);
  } catch (error) {
    console.log(
      "\n========== Usuario_service | ERROR | validar_permiso_rol ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
    return false;
  }
};
```

2.7. Actualizar información del usuario

a. Actualizar información personal

En la Tabla 10 se muestra codificación para actualizar la información personal en la wallet del usuario. Primero se valida si el _id existe en la petición, luego se valida el token asignado para

ese momento. En segunda instancia se valida si la información personal del usuario existe en la petición, y también si el _id enviado le pertenece a algún usuario en la base de datos. Una vez obtenido al usuario, se consigue la configuración de su wallet para realizar una petición al módulo “Back-end” para que actualice la información personal del usuario en su wallet en la blockchain. Por último, se verifica el estado de la respuesta, al haber actualizado en la wallet, para mostrar mensaje de error o éxito al usuario.

Tabla 10: *Codificación para actualizar la información personal del usuario.*

```
exports.actualizar_info_usuario = async function (req, res) {
  try {
    let _id = req.body._id;
    if (!_id) throw { mensaje: mensaje_api.cuenta.id_no };

    let token = await req.headers.authorization;
    let check_token = await usuario_service.validar_usuario_token_handle(
      _id,
      token
    );
    if (!check_token) throw { mensaje: mensaje_api.keyToken.token_no };

    let informacion = req.body.informacion;

    let check_campos = await api_service.validar_campos({ informacion });
    if (check_campos) throw { mensaje: check_campos };

    let usuario = await usuario_service.validar_usuario_id(_id);

    let config_wallet = JSON.parse(usuario.config_wallet);

    let data = {
      informacion,
      config_wallet,
    };

    let response = await axios.post(path.actualizar_info_user_wallet, data);

    let check_response = response.data.tipo === "success" ? true : false;
    if (!check_response) throw { mensaje: mensaje_api.cuenta.actualizar_no };

    await usuario_service.actualizar_visibilidad(_id, false, null);

    await validar_api.succeesServer(
      req,
      res,
      null,
      mensaje_api.cuenta.actualizar_ok
```

```
    );
  } catch (error) {
    console.log(
      "\n========== Usuario_controller | ERROR | actualizar_info_usuario()  ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
    await validar_api.errorServer(req, res, error);
  }
};
```

b. Actualizar información de cursos

En la Tabla 11 se muestra codificación para actualizar la información de los cursos en la wallet del usuario. Primero se valida si el _id existe en la petición, luego se valida el token asignado para ese momento. En segunda instancia se valida si la información de los cursos del usuario existe en la petición, y también si el _id enviado le pertenece a algún usuario en la base de datos. Una vez obtenido al usuario, se consigue la configuración de su wallet para realizar una petición al módulo “Back-end” para que actualice la información de los cursos del usuario en su wallet en la blockchain. Por último, se verifica el estado de la respuesta, al haber actualizado en la wallet, para mostrar mensaje de error o éxito al usuario.

Tabla 11: *Codificación para actualizar la información de los cursos del usuario.*

```
exports.actualizar_cursos_usuario = async function (req, res) {
  try {
    let { _id, cursos } = req.body;
    if (!_id) throw { mensaje: mensaje_api.cuenta.id_no };

    let token = await req.headers.authorization;
    let check_token = await usuario_service.validar_usuario_token_handle(
      _id,
      token
    );
    if (!check_token) throw { mensaje: mensaje_api.keyToken.token_no };

    let check_campos = await api_service.validar_campos({ cursos });
    if (check_campos) throw { mensaje: check_campos };

    let usuario = await usuario_service.validar_usuario_id(_id);

    let config_wallet = JSON.parse(usuario.config_wallet);

    let data = {
      cursos,
```

```
      config_wallet,
    };

    let response = await axios.post(path.actualizar_cursos_user_wallet, data);
    let check_response = response.data.tipo === "success" ? true : false;
    if (!check_response) throw { mensaje: mensaje_api.cuenta.actualizar_no };

    await usuario_service.actualizar_visibilidad(_id, false, null);

    await validar_api.succeesServer(
      req,
      res,
      null,
      mensaje_api.cuenta.actualizar_ok
    );
  } catch (error) {
    console.log(
      "\n========== Usuario_controller | ERROR |
actualizar_cursos_usuario()  ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
    await validar_api.errorServer(req, res, error);
  }
};
```

c. Actualizar información de transacciones

En la Tabla 12 se muestra codificación para actualizar la información de las transacciones en la wallet del usuario. Primero se valida si el _id existe en la petición, luego se valida el token asignado para ese momento. En segunda instancia se valida si el código y la visibilidad existe en la petición, y también si el _id enviado le pertenece a algún usuario en la base de datos. Una vez obtenido al usuario, se consigue la configuración de su wallet para realizar una petición al módulo “Back-end” para que actualice la información de la transacción del usuario en su wallet en la blockchain. Por último, se verifica el estado de la respuesta, al haber actualizado en la wallet, para mostrar mensaje de error o éxito al usuario.

Tabla 12: *Codificación para actualizar la información de las transacciones del usuario.*

```
exports.visibilidad_transaccion = async function (req, res) {
  try {
    let _id = req.body._id;
    if (!_id) throw { mensaje: mensaje_api.cuenta.id_no };
    let token = await req.headers.authorization;
```

```
    let check_token = await usuario_service.validar_usuario_token_handle(
      _id,
      token
    );
    if (!check_token) throw { mensaje: mensaje_api.keyToken.token_no };

    let { codigo, visible } = req.body;

    let check_campos = await api_service.validar_campos({ codigo, visible });
    if (check_campos) throw { mensaje: check_campos };

    let usuario = await usuario_service.validar_usuario_id(_id);
    if (!usuario) throw { mensaje: mensaje_api.cuenta.id_no };

    let data = {
      config_wallet: usuario.config_wallet,
      codigo,
      visible,
    };

    let response = await axios.post(path.visibilidad_transaccion, data);

    let check_response = response.data.tipo === "success" ? true : false;

    if (!check_response) throw { mensaje: mensaje_api.transaccion.visible_no };

    let data_response = response.data.data;

    await validar_api.succeesServer(
      req,
      res,
      data_response,
      mensaje_api.transaccion.visible_ok
    );
  } catch (error) {
    console.log(
      "\n========== Usuario_controller | ERROR | visibilidad_transaccion()  ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
    await validar_api.errorServer(req, res, error);
  }
};
```

2.8. Ejecutar fases de la transacción

a. Definir la transacción

En la Tabla 13 se muestra codificación para definir la transacción. Primero se valida si el _id existe en la petición, luego se valida el token asignado para ese momento. En segunda instancia se valida si el código de la transacción existe en la petición, y también si el _id enviado le pertenece a algún usuario en la base de datos. Una vez obtenido al usuario, se consigue la configuración de su wallet para realizar una petición al módulo “Back-end” para que defina la transacción en la blockchain. Por último, se verifica el estado de la respuesta, al haber definido la transacción, para mostrar mensaje de error o éxito al usuario.

Tabla 13: *Codificación para definir la transacción.*

```
exports.definir_transaccion = async function (req, res) {
  try {
    let _id = req.body._id;
    if (!_id) throw { mensaje: mensaje_api.cuenta.id_no };
    let token = await req.headers.authorization;

    let check_token = await usuario_service.validar_usuario_token_handle(
      _id,
      token
    );
    if (!check_token) throw { mensaje: mensaje_api.keyToken.token_no };

    let codigo = req.body.codigo;

    let check_campos = await api_service.validar_campos({ codigo });
    if (check_campos) throw { mensaje: check_campos };

    let usuario = await usuario_service.validar_usuario_id(_id);
    if (!usuario) throw { mensaje: mensaje_api.cuenta.id_no };

    let data = {
      destinatario: usuario.correo,
      codigo,
      config_wallet: usuario.config_wallet,
    };

    let response = await axios.post(path.definir_transaccion, data);

    let check_response = rresponse.data.tipo === "success";

    if (!check_response) throw { mensaje: mensaje_api.transaccion.definir_no };

    await validar_api.succeesServer(
```

```
      req,
      res,
      null,
      mensaje_api.transaccion.definir_ok
    );
  } catch (error) {
    console.log(
      "\n========== Usuario_controller | ERROR | definir_transaccion()  ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
    await validar_api.errorServer(req, res, error);
  }
};
```

b. Confirmar la transacción

En la Tabla 14 se muestra codificación para confirmar la transacción. Primero se valida si el _id existe en la petición, luego se valida el token asignado para ese momento. En segunda instancia se valida si el código de la transacción existe en la petición, y también si el _id enviado le pertenece a algún usuario en la base de datos. Una vez obtenido al usuario, se consigue la configuración de su wallet para realizar una petición al módulo “Back-end” para que confirme la transacción en la blockchain. Por último, se verifica el estado de la respuesta, al haber confirmado la transacción, para mostrar mensaje de error o éxito al usuario.

Tabla 14: *Codificación para confirmar la transacción.*

```
exports.confirmar_transaccion = async function (req, res) {
  try {
    let _id = req.body._id;
    if (!_id) throw { mensaje: mensaje_api.cuenta.id_no };
    let token = await req.headers.authorization;

    let check_token = await usuario_service.validar_usuario_token_handle(
      _id,
      token
    );
    if (!check_token) throw { mensaje: mensaje_api.keyToken.token_no };

    let codigo = req.body.codigo;

    let check_campos = await api_service.validar_campos({ codigo });
    if (check_campos) throw { mensaje: check_campos };

    let usuario = await usuario_service.validar_usuario_id(_id);
```

```
    if (!usuario) throw { mensaje: mensaje_api.cuenta.id_no };

    let data = {
      remitente: usuario.correo,
      codigo,
      config_wallet: usuario.config_wallet,
    };

    let response = await axios.post(path.confirmar_transaccion, data);

    let check_response = response.data.tipo === "success";

    if (!check_response)
      throw { mensaje: mensaje_api.transaccion.confirmar_no };

    await validar_api.succeesServer(
      req,
      res,
      null,
      mensaje_api.transaccion.confirmar_ok
    );
  } catch (error) {
    console.log(
      "\n========== Usuario_controller | ERROR | confirmar_transaccion()  ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
    await validar_api.errorServer(req, res, error);
  }
};
```

c. Enviar información solicitada de la transacción

En la Tabla 15 se muestra codificación para enviar información solicitado de la transacción. Primero se valida si el _id existe en la petición, luego se valida el token asignado para ese momento. En segunda instancia se valida si el código de la transacción y los atributos (contienen la información) existen en la petición, la información de los atributos y también si el _id enviado le pertenece a algún usuario en la base de datos. Una vez obtenido al usuario, se consigue la configuración de su wallet para realizar una petición al módulo “Back-end” para que confirme la transacción en la blockchain. Por último, se verifica el estado de la respuesta, al haber confirmado la transacción, para mostrar mensaje de error o éxito al usuario.

Tabla 15: *Codificación para enviar información solicitada de la transacción.*

```
exports.enviar_informacion_transaccion = async function (req, res) {
  try {
    let _id = req.body._id;
    if (!_id) throw { mensaje: mensaje_api.cuenta.id_no };
    let token = await req.headers.authorization;

    let check_token = await usuario_service.validar_usuario_token_handle(
      _id,
      token
    );
    if (!check_token) throw { mensaje: mensaje_api.keyToken.token_no };

    let { codigo, atributos } = req.body;

    let check_campos = await api_service.validar_campos({ codigo, atributos });
    if (check_campos) throw { mensaje: check_campos };

    let usuario = await usuario_service.validar_usuario_id(_id);
    if (!usuario) throw { mensaje: mensaje_api.cuenta.id_no };

    atributos = await helpers.sanitizar_atributos(atributos);

    let check_atributos = await helpers.v_atributos(atributos);
    if (check_atributos) throw { mensaje: check_atributos };

    let data = {
      destinatario: usuario.correo,
      codigo,
      config_wallet: usuario.config_wallet,
      atributos,
    };

    let response = await axios.post(path.enviar_informacion_transaccion, data);

    let check_response = response.data.tipo === "success";

    if (!check_response) throw { mensaje: mensaje_api.transaccion.enviar_no };

    await validar_api.succeesServer(
      req,
      res,
      null,
      mensaje_api.transaccion.enviar_ok
    );
  } catch (error) {
    console.log(
```

```
      "\n========== Usuario_controller | ERROR |
enviar_informacion_transaccion()  ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
    await validar_api.errorServer(req, res, error);
  }
};
```

d. Aceptar información de la transacción

En la Tabla 16 se muestra codificación para aceptar información de la transacción. Primero se valida si el _id existe en la petición, luego se valida el token asignado para ese momento. En segunda instancia se valida si el código de la transacción existe en la petición, y también si el _id enviado le pertenece a algún usuario en la base de datos. Una vez obtenido al usuario, se consigue la configuración de su wallet para realizar una petición al módulo “Back-end” para que acepte la información de la transacción en la blockchain. Por último, se verifica el estado de la respuesta, al haber aceptado la información, para mostrar mensaje de error o éxito al usuario.

Tabla 16: *Codificación para aceptar información de la transacción.*

```
exports.aceptar_informacion_transaccion = async function (req, res) {
  try {
    let _id = req.body._id;
    if (!_id) throw { mensaje: mensaje_api.cuenta.id_no };
    let token = await req.headers.authorization;

    let check_token = await usuario_service.validar_usuario_token_handle(
      _id,
      token
    );
    if (!check_token) throw { mensaje: mensaje_api.keyToken.token_no };

    let codigo = req.body.codigo;

    let check_campos = await api_service.validar_campos({ codigo });
    if (check_campos) throw { mensaje: check_campos };

    let usuario = await usuario_service.validar_usuario_id(_id);
    if (!usuario) throw { mensaje: mensaje_api.cuenta.id_no };

    let data = {
      remitente: usuario.correo,
      codigo,
```

```
      config_wallet: usuario.config_wallet,
    };

    let response = await axios.post(path.aceptar_informacion_transaccion, data);

    let check_response = response.data.tipo === "success";

    if (!check_response) throw { mensaje: mensaje_api.transaccion.aceptar_no };

    await validar_api.succeesServer(
      req,
      res,
      null,
      mensaje_api.transaccion.aceptar_ok
    );
  } catch (error) {
    console.log(
      "\n========== Usuario_controller | ERROR |
aceptar_informacion_transaccion()  ==========\n"
    );
    console.log(error);
    console.log(
      "\n====================== Fin | ERROR | ======================\n"
    );
    await validar_api.errorServer(req, res, error);
  }
};
```

Anexo 6: Plan de Pruebas Unitarias para el Subsistema de Aplicación.

Plan de Pruebas Unitarias para el Subsistema de Aplicación

Proyecto: Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja

Versión: 2.0

Fecha: 18/8/2023

Autor: Alexis Armijos
Correo electrónico: carlos.a.armijos@unl.edu

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja		
Entregable	Planes de Pruebas Unitarias para Subsistema de Aplicación		
Autor	Alexis Armijos		
Versión/Edición	2.0	**Fecha Versión**	17/08/2023
Aprobado por	Cristian Ramiro Narváez Guillen, Mg. Sc.	**Fecha Aprobación**	18/8/2023
		N.º Total de Páginas	8

Registro de cambios

Versión doc.	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas Unitarias subsistema de Aplicaciones	Carlos Alexis Armijos Rios	15/08/2023
2.0	Versión final del Plan de Pruebas Unitarias subsistema de Aplicaciones	Carlos Alexis Armijos Rios	17/8/2023

Control de distribución

Nombre y Apellidos
Carlos Alexis Armijos Rios
Cristian Ramiro Narváez Guillen, Mg. Sc

1 Introducción

1.1 Objetivo

El objetivo de este documento es verificar el correcto desempeño de las principales funcionalidades del subsistema de Aplicación.

1.2 Propósito

Comprobar el correcto funcionamiento del subsistema de Aplicación validando los bloques de código de las principales funcionalidades, para asegurar que cada uno funcione correctamente y eficientemente por separado.

2 Definición de los casos de pruebas

En este apartado se describen a detalle cada uno de los casos de pruebas que se han identificado como necesarios para verificar la funcionalidad completa de la aplicación.

Tabla resumen de todos los casos de prueba:

Número del Caso de Prueba	Módulo	Descripción de lo que se probará	Prerrequisitos
CP-01	Middleware	Crear el token dinámico al iniciar sesión el usuario	Usuario registrado
CP-02	Middleware	Enviar el correo con el token de registro al usuario	Correo del usuario válido
CP-03	Middleware	Enviar el correo con el token de resetear contraseña al usuario	Usuario registrado
CP-04	Middleware	Validar únicamente usuarios de la Universidad Nacional de Loja	N/A
CP-05	Chain code	Establecer la cookie de sesión con su debido tiempo de expiración	Usuario registrado ha iniciado sesión
CP-06	Front-end	Borrar la cookie de sesión del usuario	Usuario con sesión activa
CP-07	Front-end	Comprobar que las rutas de las vistas principales estén protegidas	N/A
CP-08	Middleware	Validar el rol del usuario antes de realizar alguna acción	Usuario con sesión activa
CP-09	Middleware	Actualizar la información personal de usuario en su wallet	Usuario con sesión activa
CP-10	Middleware	Actualizar la información de los cursos del usuario en su wallet	Usuario con sesión activa y haber realizado algún curso en la Carrera
CP-11	Middleware	Actualizar la información de la transacción del usuario en su wallet	Usuario con sesión activa y haber realizado alguna transacción
CP-12	Middleware	Definir la transacción en la blockchain	Usuario con sesión activa y esquemas de transcripción disponibles
CP-13	Middleware	Confirmar la transacción en la blockchain	Usuario con sesión activa y transacción definida
CP-14	Middleware	Enviar la información solicitada de la transacción en la blockchain	Usuario con sesión activa y transacción confirmada
CP-15	Middleware	Aceptar la información de la transacción en la blockchain	Usuario con sesión activa y transacción con información enviada

Casos de prueba a detalle:

CP-01					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Crear el token dinámico al iniciar sesión el usuario	g_token_handle ()	_id	✓	N/A

CP-02					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
2	Enviar el correo con el token de registro al usuario	enviar_email_registro()	correo, token	✓	N/A

CP-03					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
3	Enviar el correo con el token de resetear contraseña al usuario	enviar_email_reset()	correo, token	✓	N/A

CP-04					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
4	Validar únicamente usuarios de la Universidad Nacional de Loja	validar_correo()	correo	✓	N/A

CP-05					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
5	Establecer la cookie de sesión con su debido tiempo de expiración	set_cookie_sesion()	token, expiration	✓	N/A

CP-06					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
6	Borrar la cookie de sesión del usuario	cerrar_sesion()	N/A	✓	N/A

CP-07					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
7	Comprobar que as rutas de las vistas principales estén protegidas	router.beforeEach()	to, from, next	✓	N/A

CP-08					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
8	Validar el rol del usuario antes de realizar alguna acción	validar_permiso_rol()	rol_usuario, rol_necesario	✓	N/A

CP-09					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
9	Actualizar la información personal de usuario en su wallet	actualizar_info_usuario()	_id, token, Información	✓	N/A

CP-10					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
10	Actualizar la información de los cursos del usuario en su wallet	actualizar_cursos_usuario()	_id, token, cursos	✓	N/A

CP-11					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
11	Actualizar la información de la transacción del usuario en su wallet	visibilidad_transacción()	_id, token, codigo, visible	✓	N/A

CP-12					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
12	Definir la transacción en la blockchain	definir_transacción()	_id, token, codigo	✓	N/A

CP-13					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
13	Confirmar la transacción en la blockchain	confirmar_transacción()	_id, token, codigo	✓	N/A

CP-14					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
14	Enviar la información solicitada de la transacción en la blockchain	enviar_informacion_transaccion()	_id, token, codigo, atributos	✓	N/A

CP-15					
N.º	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
15	Aceptar la información de la transacción en la blockchain	aceptar_informacion_transaccion()	_id, token, codigo	✓	N/A

Anexo 7: Plan de Pruebas de Integración.

Plan de Pruebas de Integración

Proyecto: Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja

Versión: 2.0

Fecha: 23/08/2023

Autor: Alexis Armijos
Correo electrónico: carlos.a.armijos@unl.edu

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja		
Entregable	Plan de Pruebas de Integración		
Autor	Alexis Armijos		
Versión/Edición	2.0	**Fecha Versión**	23/08/2023
Aprobado por	Cristian Ramiro Narváez Guillen, Mg. Sc.	**Fecha Aprobación**	23/08/2023
		N.º Total de Páginas	11

Registro de cambios

Versión doc.	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas de Integración	Carlos Alexis Armijos Rios	21/08/2023
2.0	Versión final del Plan de Pruebas de Integración	Carlos Alexis Armijos Rios	23/08/2023

Control de distribución

Nombre y Apellidos
Carlos Alexis Armijos Rios
Cristian Ramiro Narváez Guillen, Mg. Sc

1 Introducción

1.1 Objetivo

El objetivo de este documento es elaborar, ejecutar y validar los casos de prueba para verificar que el sistema tiene un ensamblaje correcto de todos sus módulos y, además, que todas sus funcionalidades se desempeñen adecuadamente. Los casos de prueba de integración son generados a partir de los principales requerimientos funcionales, aplicados después de comprobar que todos los módulos funcionan correctamente por separado.

1.2 Propósito

Comprobar el correcto funcionamiento del sistema para el proyecto “Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja”, validando la correcta integración de los 3 módulos: “Front-end”, “Middleware” y “Back-end” mediante los principales requerimientos funcionales establecidos al inicio del proyecto.

2 Definición de los casos de pruebas

En este apartado se describen a detalle cada uno de los casos de pruebas que se han identificado para validar la integración de los módulos del sistema.

Tabla resumen de todos los casos de prueba de integración:

Número del Caso de Prueba	Componente	Descripción de lo que se probará	Prerrequisitos
CP-01	Subsistema de contratos inteligentes – Subsistema de aplicación	Se registra correctamente el usuario en el sistema	• Ingresar a la página • El correo del usuario debe estar registrado en la página de la Carrera de Computación • El token debe ser válido • La información ingresada debe respetar los formatos
CP-02	Subsistema de contratos inteligentes – Subsistema de aplicación	Se inicia sesión correctamente en el sistema	• Ingresar a la página • El usuario debe estar registrado en el sistema • La información ingresada debe respetar los formatos
CP-03	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario administra su información personal	• Ingresar a la página • Iniciar sesión • Ir a la vista “Wallet - Información” • Tener información personal para administrar
CP-04	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario administra sus cursos de la página de la Carrera de Computación	• Ingresar a la página • Iniciar sesión • Ir a la vista “Wallet - Cursos” • Haber participado en los cursos de la página de la Carrera de Computación
CP-05	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario administra sus transacciones	• Ingresar a la página • Iniciar sesión • Ir a la vista “Wallet - Transacciones” • Haber utilizado los esquemas para obtener información de otros usuarios
CP-06	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario utiliza los esquemas	• Ingresar a la página • Iniciar sesión • Ir a la vista “Utilizar Esquema” • Debe haber esquemas disponibles • Debe haber usuarios disponibles para realizar la transacción
CP-07	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario visualiza su historial de transacciones	• Ingresar a la página • Iniciar sesión • Ir a la vista “Historial Transacciones” • Debe haber realizado transacciones
CP-08	Subsistema de contratos inteligentes – Subsistema de aplicación	El usuario realiza el proceso de transacción correctamente	• Ingresar a la página • Iniciar sesión • Ir a la vista “Ver Transacción” • Debe haber utilizado algún esquema con algún usuario • La información de los atributos del esquema debe respetar los formatos

Casos de prueba de integración a detalle:

CP-01					
Paso	**Descripción de los pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Ingresar a la página del sistema	Link de la página en el navegador	Página web del sistema	✓	N/A
2	Hacer clic en el enlace denominado "¿No tienes cuenta?"	Clic	Vista para verificar si el correo es válido en el sistema	✓	El correo del usuario debe estar registrado en la página de la Carrera de Computación
3	Escribir el correo	String	N/A	✓	El correo sebe ser válido en su formato
4	Clic en el botón denominado "Validar correo"	Clic	Muestra mensaje de validez e instrucciones a seguir	✓	El correo debe estar registrado en el sistema
5	Ingresar a la vista de registro	Link enviado al correo	Vista para registrarse y mensaje sobre el token	✓	El token debe estar en vigencia y ser válido
6	Escribir los nombres, apellidos, cédula (DNI), teléfono y contraseña	String	Ningún mensaje de alerta en las cajas de textos	✓	La información ingresada debe respetar los formatos
7	Clic en el botón denominado "Registrarse"	Clic	Mensaje de éxito y redirección a la vista de inicio	✓	N/A

CP-02					
Paso	**Descripción de los pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Ingresar a la página del sistema	Link de la página en el navegador	Página web del sistema	✓	N/A
2	Escribir el correo y contraseña	String	Ningún mensaje de alerta en las cajas de textos	✓	El correo sebe ser válido en su formato
3	Clic en el botón denominado "Iniciar sesión"	Clic	Mensaje de bienvenido y redirección a la vista de inicio	✓	El usuario debe estar registrado en el sistema

CP-03					
Paso	**Descripción de los pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Hacer clic en la opción del menú de la izquierda denominado "Wallet - Información"	Clic	Vista para administrar información personal	✓	N/A
2	Hacer clic en el botón con el símbolo + a la derecha del encabezado de la tabla	Clic	Agrega un nuevo campo de información en la tabla	✓	Si hay un nuevo campo de información sin editar, no se podrán añadir más campos

3	Editar campos de información en la tabla	String	N/A	✓	Si el campo de información es editable se podrá editar
4	Hacer clic en el botón con forma de ojo, de la columna Visible según el campo de información	Clic	Cambia de color e icono según el estado de visibilidad	✓	N/A
5	Hacer clic en el botón con forma de basurero, de la columna Acciones según el campo de información	Clic	Borra el campo de información en la tabla	✓	Si el campo de información es editable se podrá eliminar
6	Hacer clic en el botón al final de la tabla denominado “Actualizar”	Clic	Muestra un mensaje de que se actualizó la información	✓	Todos los campos de información deben estar completos

CP-04					
Paso	**Descripción de los pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Hacer clic en la opción del menú de la izquierda denominado “Wallet - Cursos”	Clic	Vista para administrar cursos	✓	N/A
2	Hacer clic en el botón con el símbolo de una flecha girando a la derecha del encabezado de la tabla	Clic	Obtiene los cursos participados en la página de la Carrera de Computación	✓	N/A
3	Escribir el nombre de algún curso en la caja de texto denominada “Buscar”	String	Busca y muestra el curso o cursos similares en la tabla	✓	Debe haber cursos para poder buscar
4	Hacer clic en el botón con el símbolo de PDF, de la columna Certificado según el curso	Clic	Se abre una nueva ventana mostrando el certificado obtenido	✓	No se generará el certificado sin haber completado el curso
5	Hacer clic en el botón con forma de ojo, de la columna Visible según el curso	Clic	Cambia de color e icono según el estado de visibilidad	✓	Debe haber cursos para poder cambiar la visibilidad
6	Hacer clic en el botón al final de la tabla denominado “Actualizar”	Clic	Muestra un mensaje de que se actualizó la información	✓	Debe haber cursos para poder actualizar

CP-05					
Paso	**Descripción de los pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Hacer clic en la opción del menú de la izquierda denominado “Wallet - Transacciones”	Clic	Vista para administrar transacciones	✓	N/A
2	Escribir el nombre de alguna transacción en la caja de texto denominada “Buscar”	String	Busca y muestra la transacción o transacciones similares en la tabla	✓	Debe haber transacciones para poder buscar

3	Hacer clic en el botón con el símbolo de una flecha con dirección hacia abajo, a la izquierda de la columna Esquema según la transacción	Clic	Se expande una ventana hacia abajo mostrando los valores de la transacción	✓	N/A
4	Hacer clic en el botón con el símbolo de PDF en algún valor dentro de la ventada que muestra los valores de la transacción	Clic	Se abre una nueva ventana mostrando el contenido del PDF	✓	El valor debe ser tipo Archivo
5	Hacer clic en el botón con forma de ojo, de la columna Visible según el curso	Clic	Cambia de color e icono según el estado de visibilidad	✓	N/A

CP-06					
Paso	**Descripción de los pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Hacer clic en la opción del menú de la izquierda denominado “Utilizar Esquema”	Clic	Vista para utilizar los esquemas	✓	N/A
2	Hacer clic en el botón que está al lado del encabezado “Solicitudes permitidas”	Clic	Cambia de color e icono según el estado para permitir solicitudes	✓	N/A
3	Escribir el nombre de algún esquema en la caja de texto denominada “Buscar”	String	Busca y muestra el esquema o los esquemas similares en la tabla	✓	Debe haber esquemas para poder buscar
4	Hacer clic en el botón con forma de archivo con un lápiz, de la columna Acciones según el esquema	Clic	Abre un modal que muestra información sobre el esquema a utilizar	✓	Debe haber esquemas
5	Hacer clic, escribir y seleccionar el usuario en la caja de selección denominada “Seleccione al usuario”	Clic y String	Muestra y fija al usuario requerido	✓	Debe haber usuarios y estos deben permitir las solicitudes
6	Hacer clic en la caja de selección denominada “Tiempo”	Clic	Muestra y fija el tiempo requerido	✓	N/A
7	Escribir en la caja de texto denominada “¿Cuánto tiempo?”	String	Muestra la cantidad de tiempo requerido	✓	El tiempo selecciona debe ser diferente a Indefinido
8	Hacer clic en el botón al final del modal, denominada “Utilizar”	Clic	Se cambia a la vista de ver transacción y muestra un mensaje de éxito	✓	Todos los campos solicitados deben estar completos

CP-07					
Paso	**Descripción de los pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Hacer clic en la opción del menú de la izquierda denominado	Clic	Vista para ver el historial de las	✓	N/A

	“Historial Transacciones”		transacciones del usuario		
3	Escribir el nombre de alguna transacción en la caja de texto denominada “Buscar”	String	Busca y muestra la transacción o transacciones similares en la tabla	✓	Debe haber transacciones para poder buscar
4	Hacer clic en el botón con forma de ojo, de la columna Acciones según la transacción	Clic	Se cambia a la vista de ver transacción	✓	N/A

CP-08					
Paso	**Descripción de los pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Ingresar a la vista ver transacción	Link en el navegador	Vista para ver el estado de transacción	✓	Haber utilizado algún esquema
3	Hacer clic en el botón denominado “Aceptar”	Clic	Mensaje de éxito al haber definido la transacción	✓	El usuario debe ser el destinatario
4	Hacer clic en el botón denominado “Aceptar”	Clic	Mensaje de éxito al haber confirmado la transacción	✓	El usuario debe ser el remitente
5	Escribir el valor de los atributos solicitados, pudiendo ser texto o archivo	String y PDF	N/A	✓	El usuario debe ser el destinatario, los archivos no puede exceder los 5MB
6	Hacer clic en el botón denominado “Aceptar”	Clic	Mensaje de éxito al haber enviado la información de la transacción	✓	El usuario debe ser el destinatario y se valida los valores de los atributos
7	Hacer clic en el botón denominado “Aceptar”	Clic	Mensaje de éxito al haber aceptado la información enviada por la transacción	✓	El usuario debe ser el remitente

3 Glosario

A continuación, se muestra la definición de todos los términos utilizados en el presente documento.

Término	Descripción
Esquema	Es el Esquema de Transcripción (también conocido como contrato inteligente) que proporciona Hyperledger Indy para que los usuarios puedan intercambiar información válida por medio de su blockchain
PDF	Formato de Documento Portátil (Portable Document Format)

4 Bibliografía y referencias

Referencia	Título
Anexo 1 (Del Documento del Proyecto de Integración Curricular)	Especificación de requisitos de software.

Anexo 8: Plan de Pruebas Funcionales.

Plan de Pruebas Funcionales

Proyecto: Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja

Versión: 2.0

Fecha: 24/08/2023

Autor: Alexis Armijos
Correo electrónico: carlos.a.armijos@unl.edu

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja		
Entregable	Plan de Pruebas Funcionales		
Autor	Alexis Armijos		
Versión/Edición	2.0	**Fecha Versión**	24/08/2023
Aprobado por	Cristian Ramiro Narváez Guillen, Mg. Sc.	**Fecha Aprobación**	24/08/2023
		N.º Total de Páginas	34

Registro de cambios

Versión doc.	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas Funcionales	Carlos Alexis Armijos Rios	22/08/2023
2.0	Versión final del Plan de Pruebas Funcionales	Carlos Alexis Armijos Rios	24/08/2023

Control de distribución

Nombre y Apellidos
Carlos Alexis Armijos Rios
Cristian Ramiro Narváez Guillen, Mg. Sc

1 Introducción

1.1 Objetivo

El objetivo de este documento es elaborar, ejecutar y validar los casos de prueba para verificar que el sistema cumple con los requisitos funcionales y no funcionales especificados. Se definirán los casos de prueba, la matriz de trazabilidad entre los casos de pruebas y requisitos funcionales y no funcionales, y también la estrategia a seguir durante la ejecución de las pruebas.

1.2 Propósito

Comprobar el correcto funcionamiento del sistema para el proyecto “Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja”, validando que el sistema ejecuta correctamente los requisitos funcionales y no funcionales.

2 Trazabilidad de casos de prueba

En las siguientes Matrices se muestra la relación entre los casos de pruebas definidos y los requisitos funcionales y no funcionales por separado. Dónde las filas representan cada uno de los de pruebas definidos mientras que las columnas los requisitos funcionales y no funcionales. La “x” representan la relación que tienen.

a. Casos de pruebas – Requisitos Funcionales

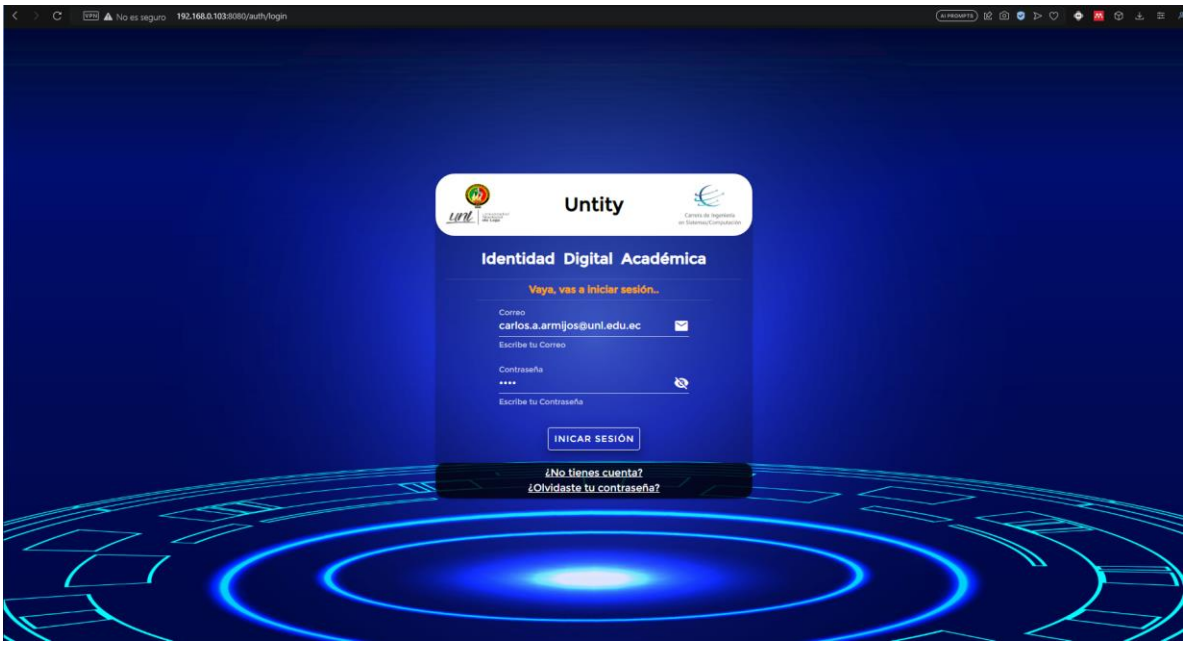
	RF01	RF02	RF03	RF04	RF05	RF06	RF07	RF08	RF09	RF10	RF11
CP01	x	x									
CP02	x		x								
CP03	x		x	x							
CP04	x	x			x				x		
CP05	x	x				x			x		
CP06	x	x							x		
CP07	x	x					x				
CP08	x	x						x	x		
CP09	x	x			x	x			x		
CP10	x	x								x	
CP11	x	x									x

b. Casos de pruebas – Requisitos No Funcionales

	RNF01	RNF02	RNF03	RNF04	RNF05	RNF06	RNF07
CP01	x	x	x	x	x		x
CP02	x	x	x	x	x		x
CP03	x	x	x	x	x		x
CP04	x	x	x	x	x		x
CP05	x	x	x	x	x		x
CP06	x	x	x	x	x		x
CP07	x	x	x	x	x	x	x
CP08	x	x	x	x	x		x
CP09	x	x	x	x	x		x
CP10	x	x	x	x	x	x	x
CP11	x	x	x	x	x		x

3 Definición de los casos de pruebas

En esta parte se describen cada uno de los casos de prueba necesarios para verificar las funcionalidades del sistema a partir de los requisitos funcionales.

<table>
<tr><td rowspan="2">Iniciar sesión en el sistema</td><td colspan="2">CPO1</td></tr>
<tr><td>¿Prueba de despliegue?</td><td>Sí</td></tr>
<tr><td colspan="3">Descripción:
Se probará la respuesta del sistema cuando el usuario vaya a iniciar sesión</td></tr>
<tr><td colspan="3">Prerrequisitos:
• Ingresar a la página.
• Estar registrado en el sistema
• Saber el correo y contraseña del usuario</td></tr>
<tr><td colspan="3">Pasos:
1. Escribir el enlace de la página web en el navegar
2. Completar las cajas de texto de Correo y Contraseña
3. Hacer clic en el botón “Iniciar sesión”
4. Esperar el mensaje informativo</td></tr>
<tr><td colspan="3">Resultado esperado:
• Las cajas de texto de Correo y Contraseña sin mensajes de error
• Se redirecciona a la vista de inicio junto a un mensaje de bienvenida</td></tr>
<tr><td colspan="3">Resultado obtenido:
• Las cajas de texto de Correo y Contraseña sin mensajes de errores
 

• Se redirecciona a la vista de inicio junto a un mensaje de bienvenida</td></tr>
</table>

Registrarse en el sistema	**CPO2**	
	¿Prueba de despliegue?	Sí

Descripción:

Se probará la respuesta del sistema cuando el usuario vaya a registrarse

Prerrequisitos:

- Ingresar a la página.
- Estar registrado en la página de la Carrera de Computación
- Saber el correo del usuario

Pasos:

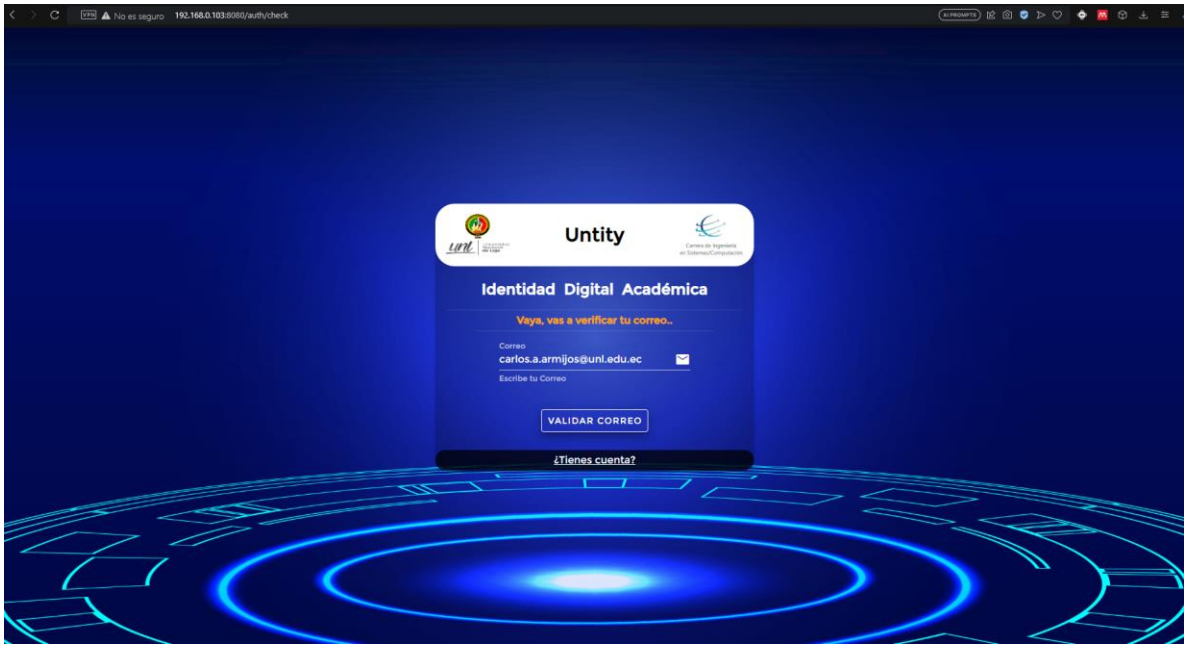
1. Escribir el enlace de la página web en el navegar
2. Clic en el enlace “¿No tienes cuenta?”
3. Completar la caja de texto de Correo
4. Hacer clic en el botón “Validar correo”
5. Esperar el mensaje informativo
6. Ir a la bandeja de entrada del correo validado
7. Clic en el enlace del mensaje en el correo validado
8. Completar las cajas de texto de Nombres, Apellidos, Cédula, Teléfono y Contraseña
9. Hacer clic en el botón “Registrarse”
10. Esperar el mensaje informativo

Resultado esperado:

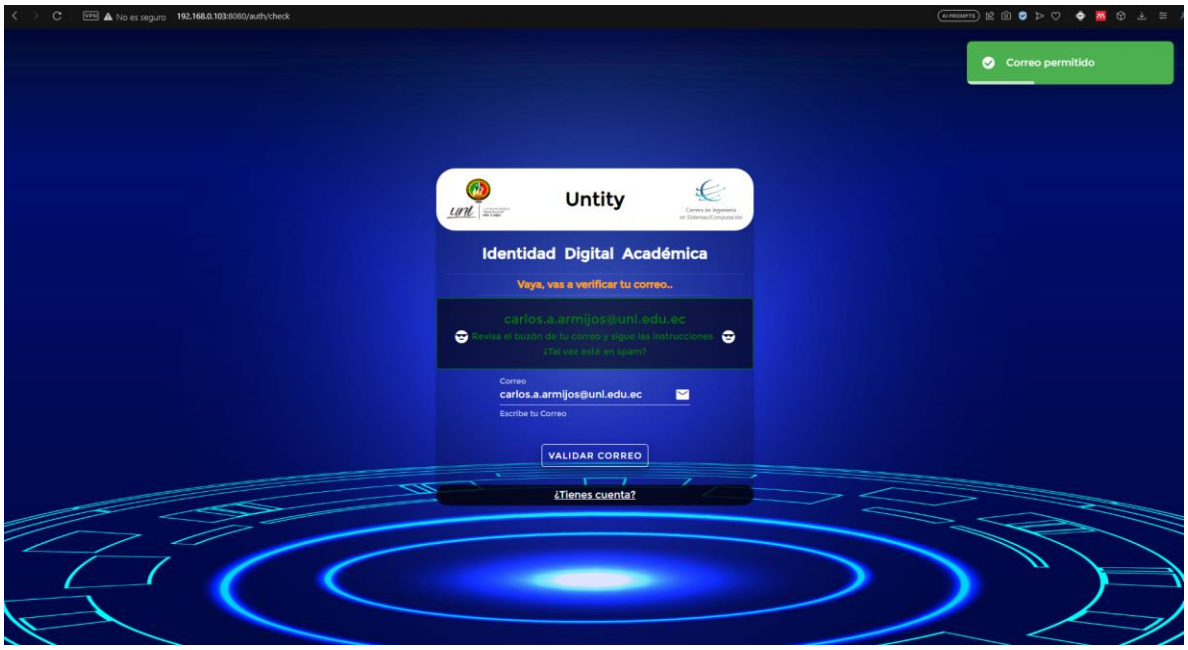
- La caja de texto de Correo sin mensaje de error
- Mensaje informativo sobre la validación del correo
- Mensaje con el enlace en la bandeja de entrada del correo
- Vista para registrarse junto al mensaje de validación del token
- Las cajas de texto de Nombres, Apellidos sin mensajes de error
- Se redirecciona a la vista de inicio junto al mensaje informativo sobre el registro del usuario

Resultado obtenido:

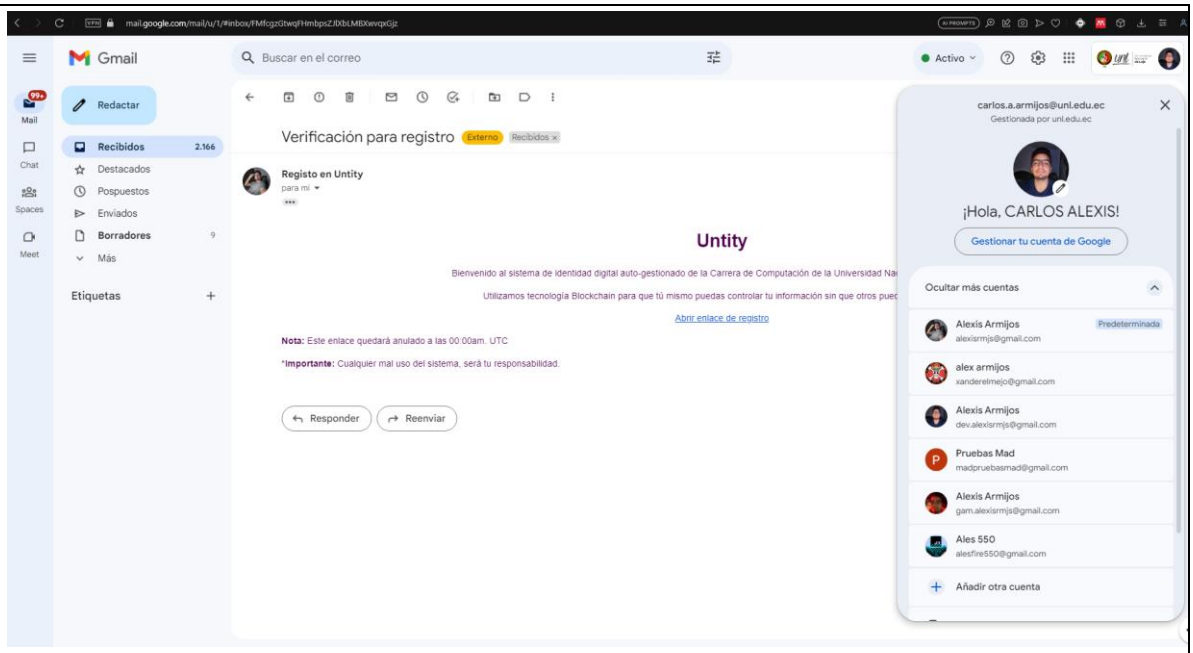
- La caja de texto de Correo sin mensaje de error



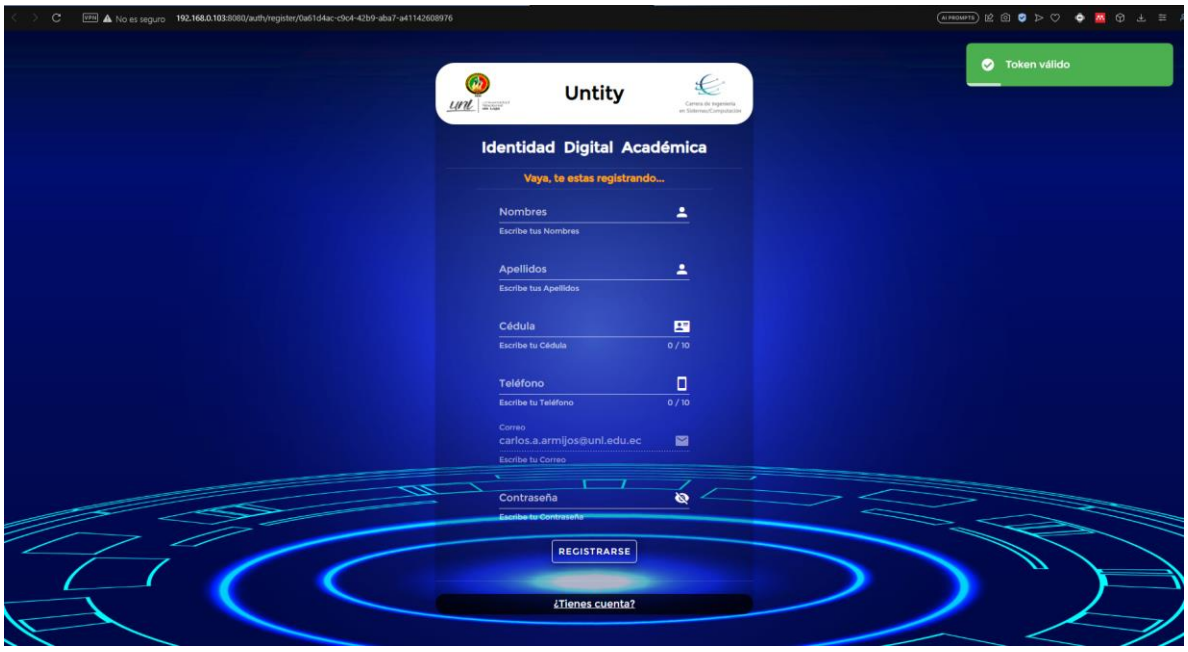
- Mensaje informativo sobre la validación del correo



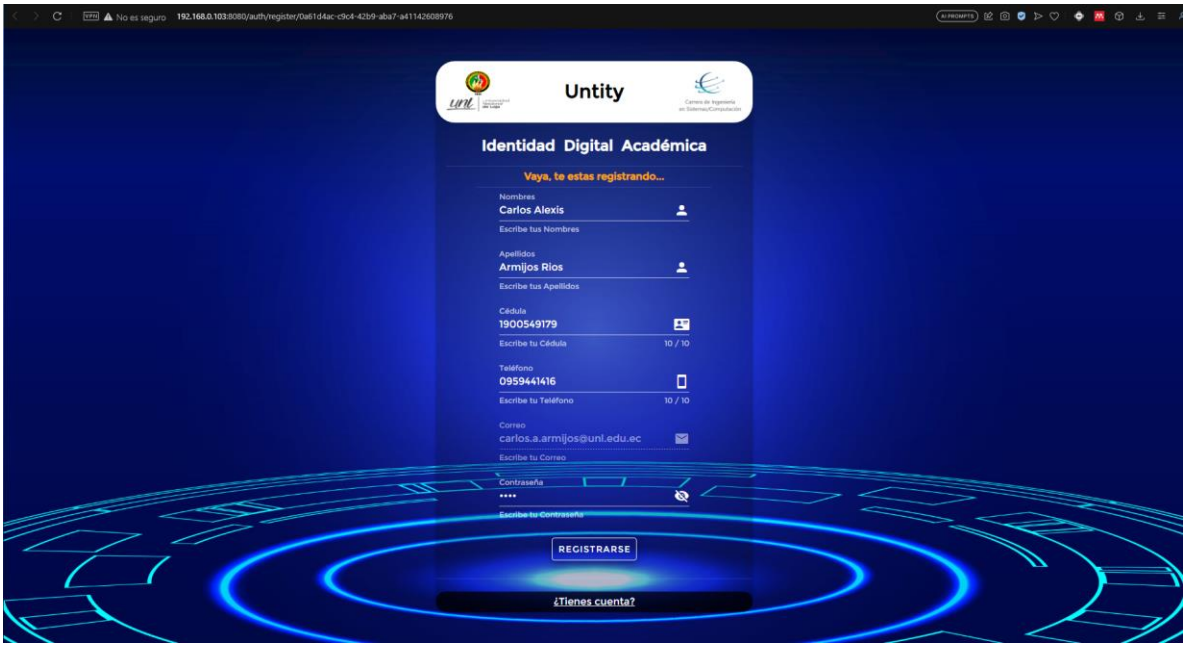
- Mensaje con el enlace en la bandeja de entrada del correo



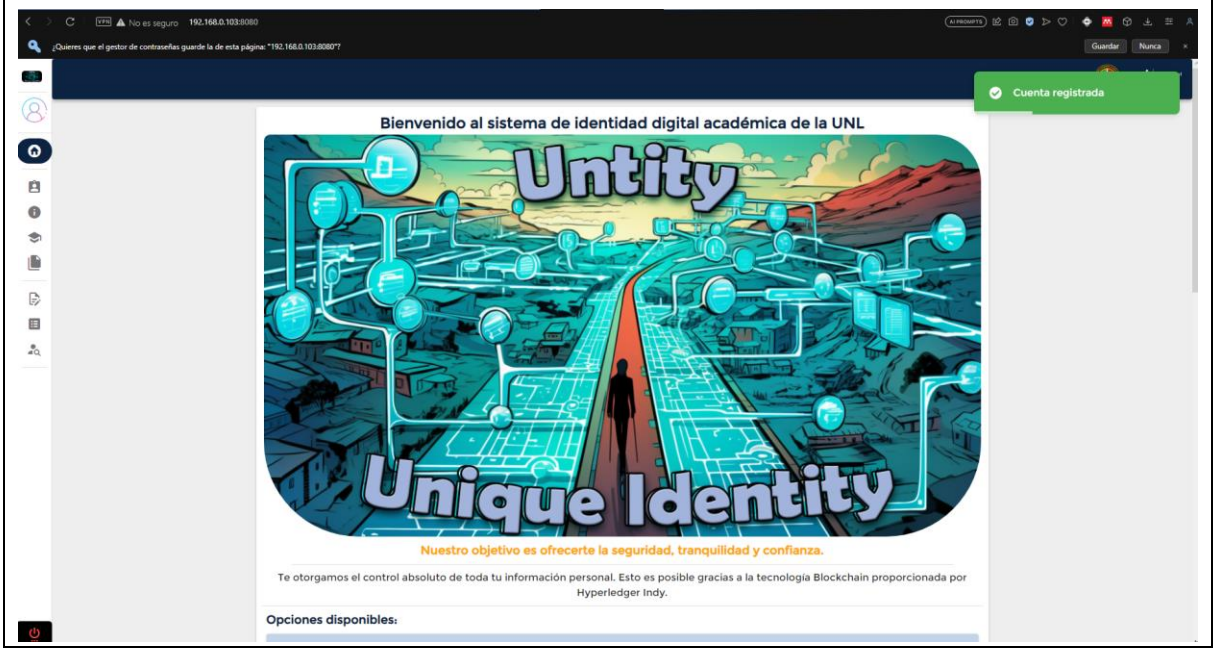
- Vista para registrarse junto al mensaje de validación del token



- Las cajas de texto de Nombres, Apellidos, Cédula, Teléfono y Contraseña sin mensajes de error

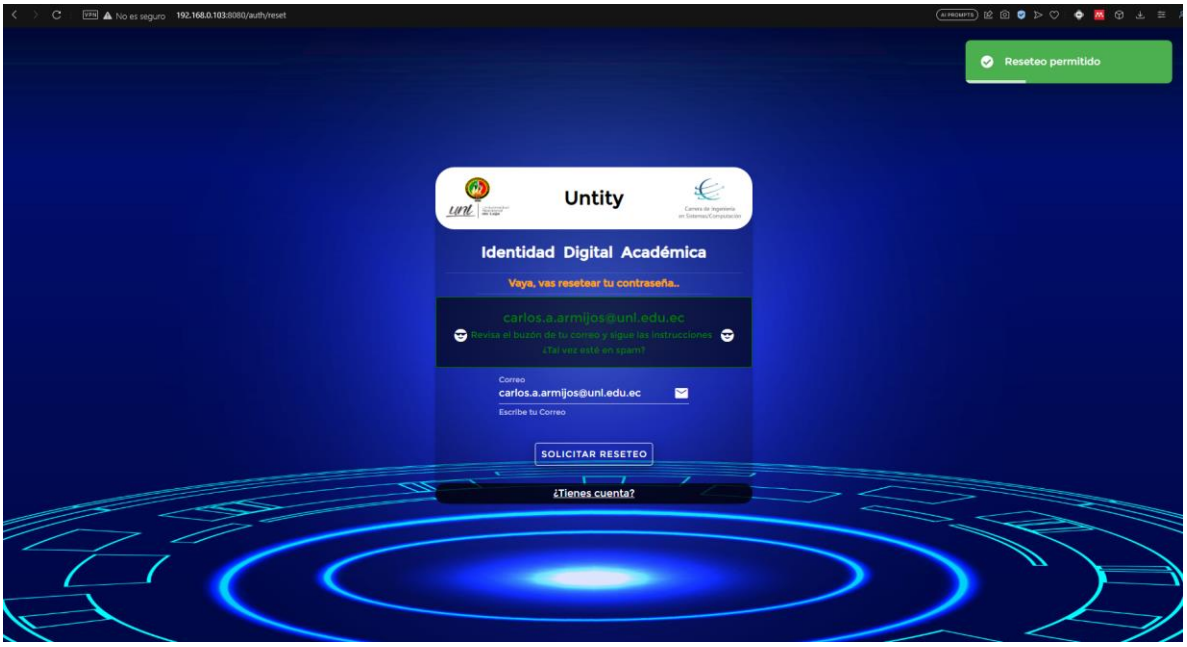


- Se redirecciona a la vista de inicio junto al mensaje informativo sobre el registro del usuario

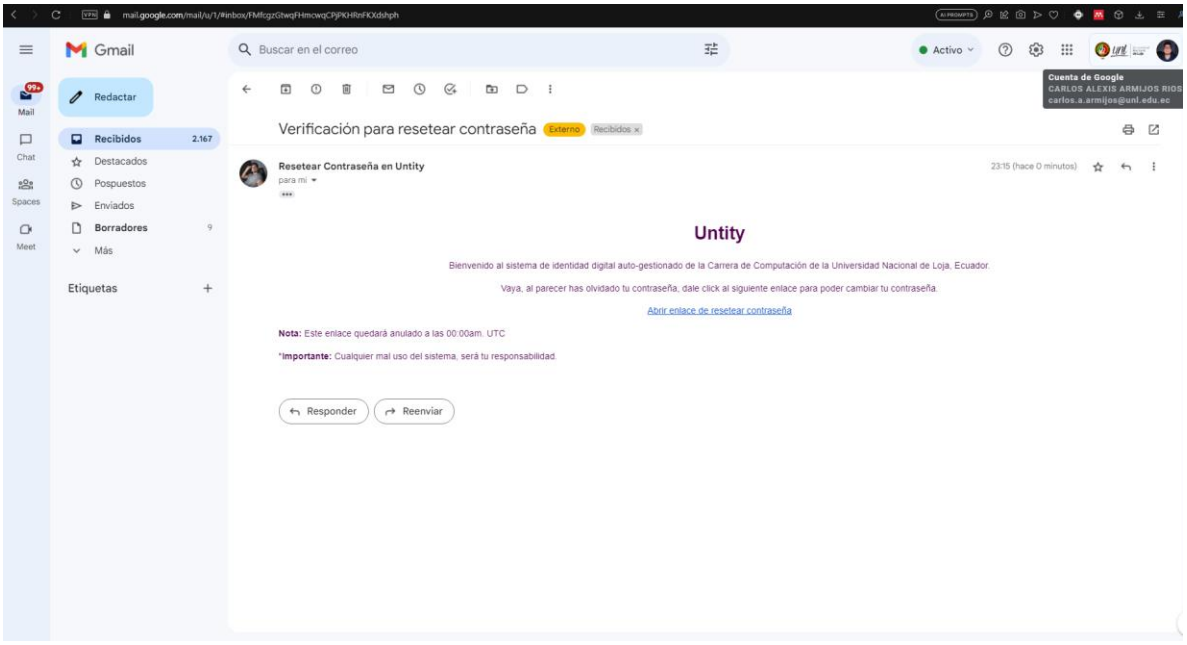


<table>
<tr><td rowspan="2">Resetear contraseña del usuario</td><td colspan="2">CPO3</td></tr>
<tr><td>¿Prueba de despliegue?</td><td>Sí</td></tr>
<tr><td colspan="3">Descripción:
Se probará la respuesta del sistema cuando el usuario vaya a resetear su contraseña</td></tr>
<tr><td colspan="3">Prerrequisitos:
• Ingresar a la página.
• Estar registrado en el sistema
• Saber el correo del usuario</td></tr>
<tr><td colspan="3">Pasos:
1. Escribir el enlace de la página web en el navegar
2. Clic en el enlace “¿Olvidaste tu contraseña?”
3. Completar la caja de texto de Correo
4. Hacer clic en el botón “Solicitar reseteo”
5. Esperar el mensaje informativo
6. Ir a la bandeja de entrada del correo validado
7. Clic en el enlace del mensaje en la bandeja de entrada del correo validado
8. Completar la caja de texto de Contraseña
9. Hacer clic en el botón “Resetear contraseña”
10. Esperar el mensaje informativo</td></tr>
<tr><td colspan="3">Resultado esperado:
• La caja de texto de Correo sin mensaje de error
• Mensaje informativo sobre la validación del correo
• Mensaje con el enlace en la bandeja de entrada del correo validado
• Vista para resetear contraseña junto al mensaje de validación del token
• La caja de texto de Contraseña sin mensaje de error
• Mensaje informativo sobre el reseteo de contraseña</td></tr>
<tr><td colspan="3">Resultado obtenido:
• La caja de texto de Correo sin mensaje de error
 

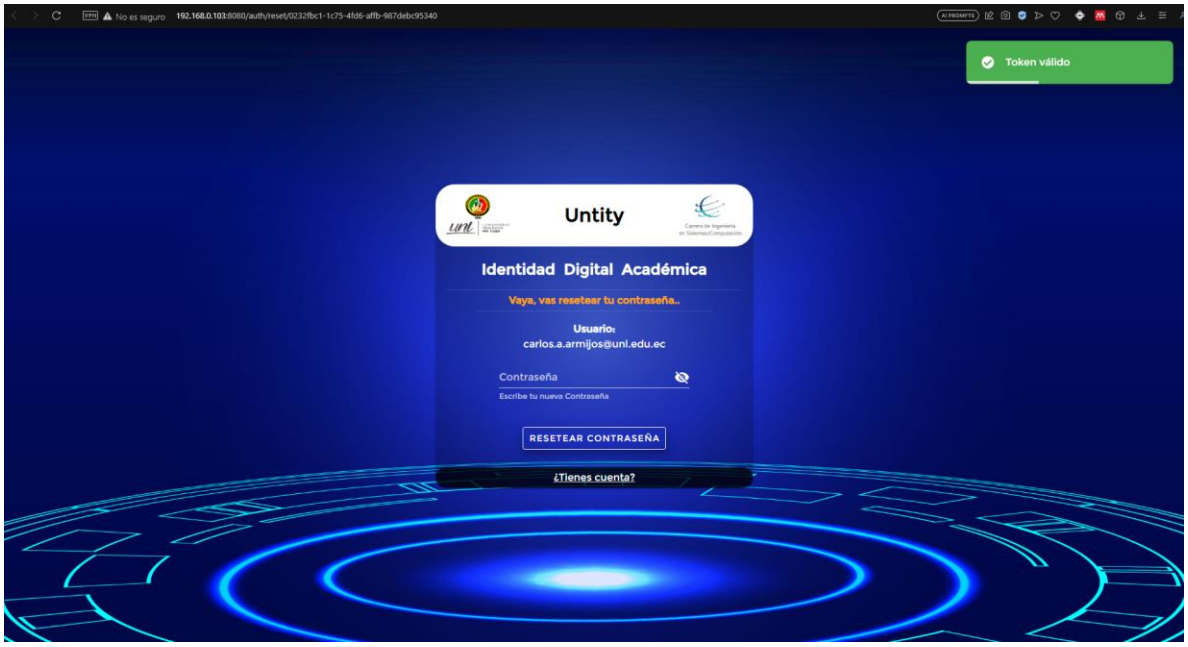
• Mensaje informativo sobre la validación del correo</td></tr>
</table>



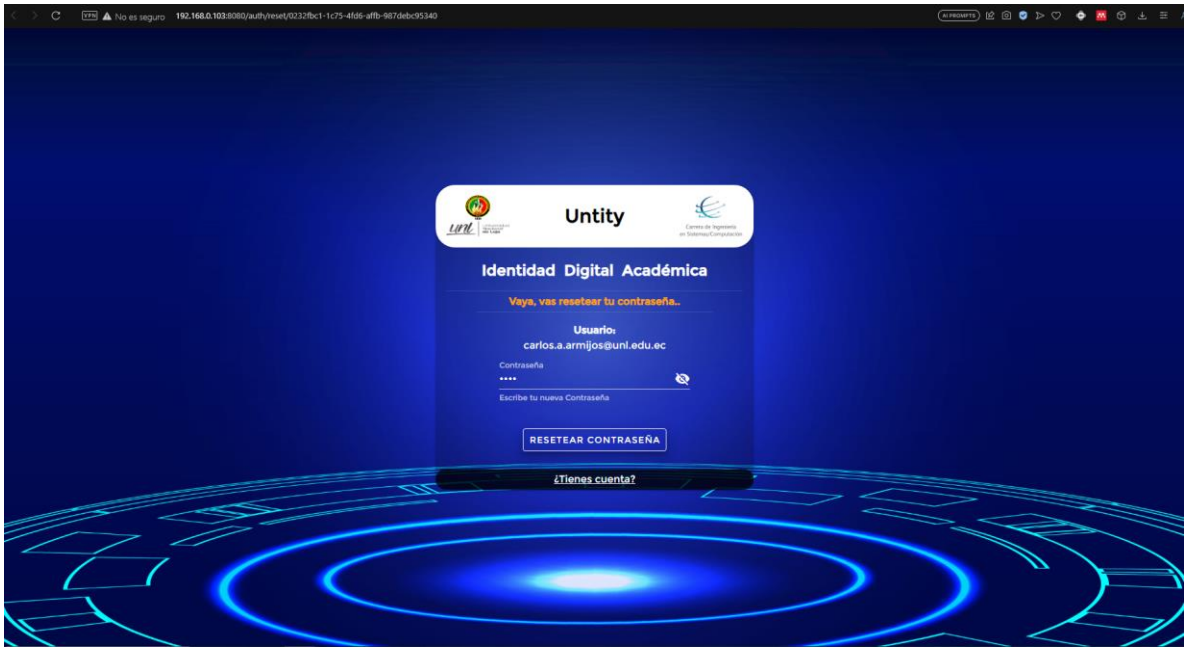
- Mensaje con el enlace en la bandeja de entrada del correo validado



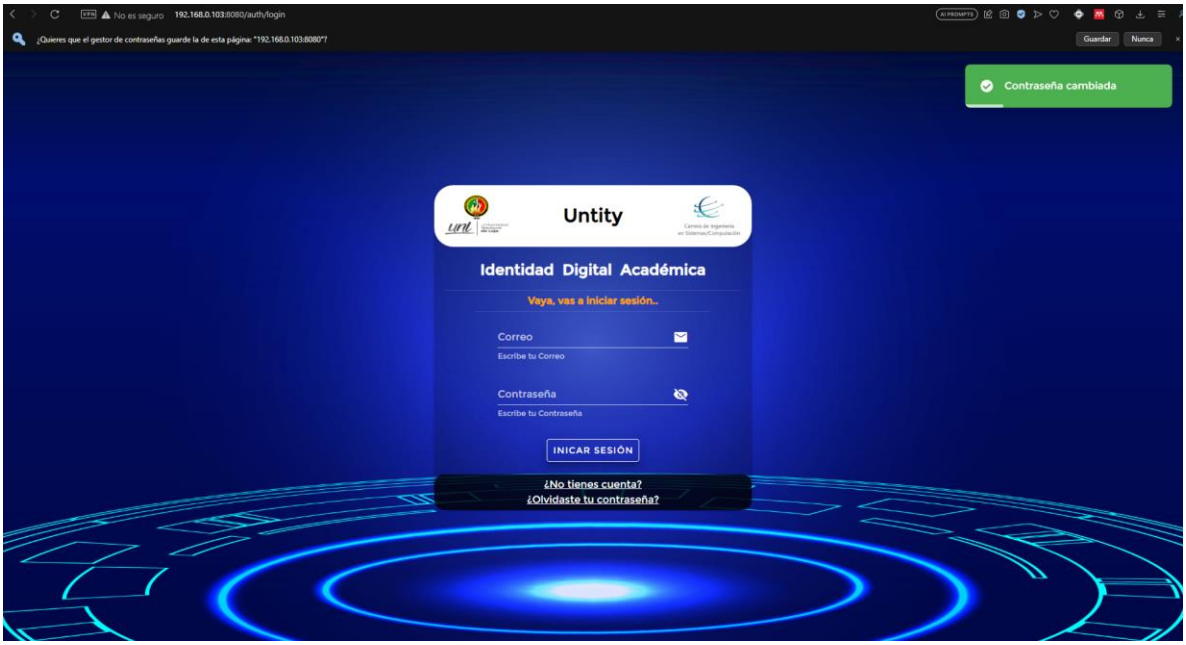
- Vista para resetear contraseña junto al mensaje de validación del token



- La caja de texto de Contraseña sin mensaje de error



- Mensaje informativo sobre el reseteo de contraseña



Administrar información personal del usuario	**CPO4**	
	¿Prueba de despliegue?	Sí

Descripción:

Se probará la respuesta del sistema cuando el usuario vaya a administrar su información personal

Prerrequisitos:

- Ingresar a la página.
- Tener una sesión activa
- Saber qué información se va a administrar

Pasos:

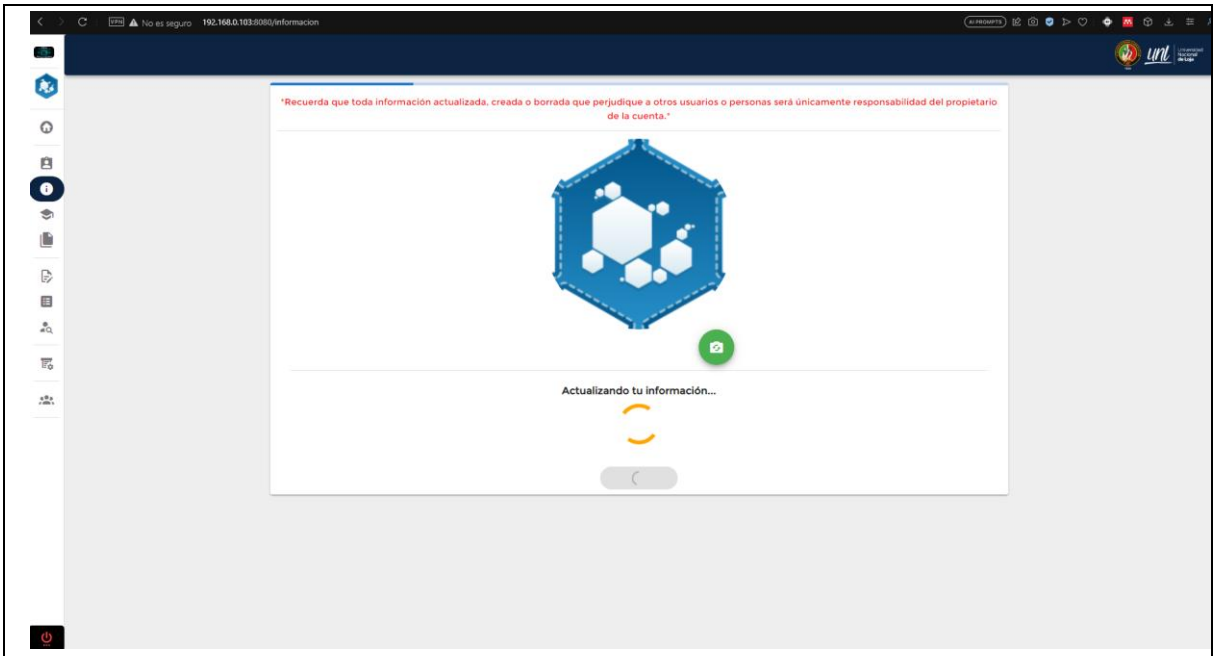
1. Escribir el enlace de la página web en el navegar
2. Iniciar sesión en el sistema
3. Hacer clic en la opción “Wallet - Información” del menú de la izquierda
4. Esperar que cargue la información
5. Clic en el botón con símbolo de + que está a la derecha del encabezado de la tabla, para agregar un nuevo campo de información
6. Completar el nuevo campo de información
7. Modificar los valores cualquier campo de información
8. Cambiar la visibilidad de cualquier campo de información al hacer clic en el símbolo con forma de ojo
9. Eliminar el campo de información al hacer clic en el símbolo con forma de basurero
10. Hacer clic en el botón “Actualizar”, al final de la tabla
11. Esperar el mensaje informativo

Resultado esperado:

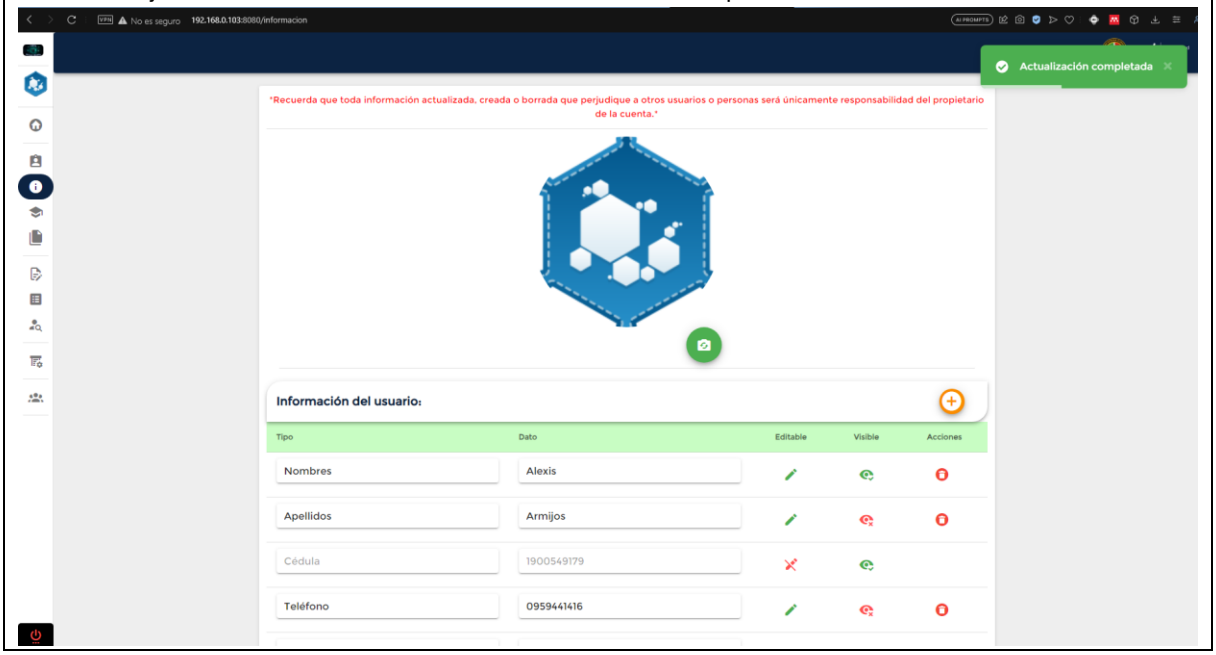
- Ningún mensaje de error al hacer clic en el botón “Actualizar”
- Mensaje informativo sobre la actualización de la información personal del usuario

Resultado obtenido:

- Ningún mensaje de error al hacer clic en el botón “Actualizar”



- Mensaje informativo sobre la actualización de la información personal del usuario



Administrar cursos del usuario	CPO5	
	¿Prueba de despliegue?	Sí
Descripción: Se probará la respuesta del sistema cuando el usuario vaya a administrar sus cursos		
Prerrequisitos: • Ingresar a la página. • Tener una sesión activa • Saber qué cursos se va a administrar		
Pasos: 1. Escribir el enlace de la página web en el navegar 2. Iniciar sesión en el sistema		

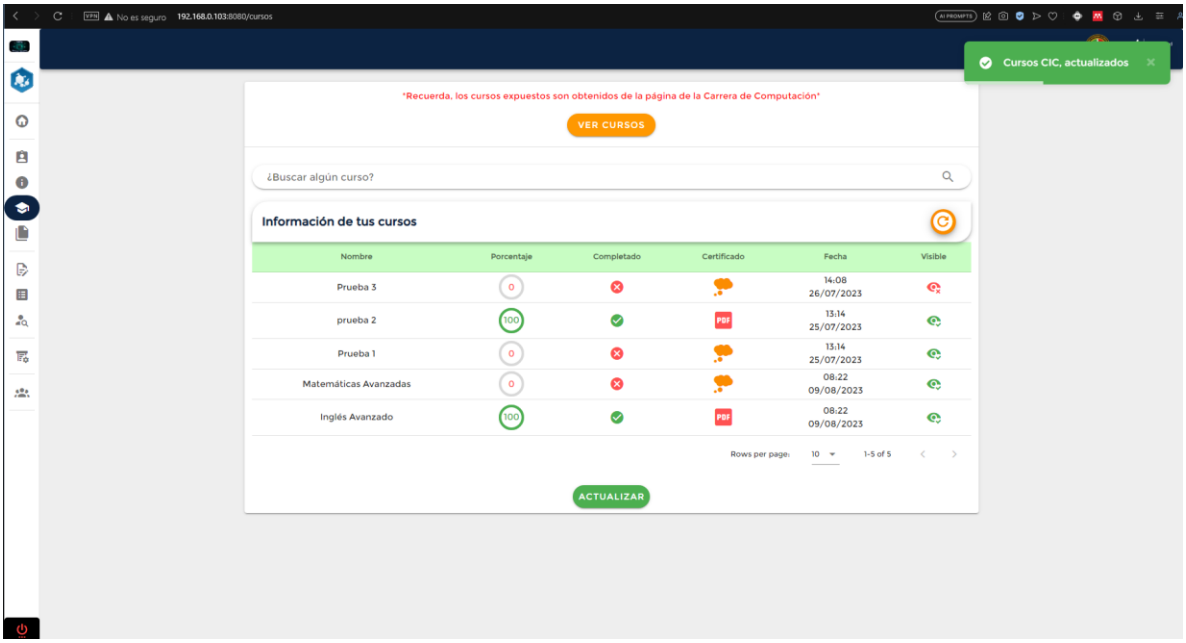
3. Hacer clic en la opción “Wallet - Cursos” del menú de la izquierda
4. Esperar que cargue la información
5. Clic en el botón con el símbolo de una flecha girando que está a la derecha del encabezado de la tabla, para obtener los cursos en los que hayas participado en la página de la Carrera de Computación
6. Esperar el mensaje informativo
7. Hacer clic en el símbolo de PDF del curso (debe estar completado para que aparezca)
8. Cambiar la visibilidad de cualquier curso al hacer clic en el símbolo con forma de ojo
9. Hacer clic en el botón “Actualizar”, al final de la tabla
10. Esperar el mensaje informativo

Resultado esperado:

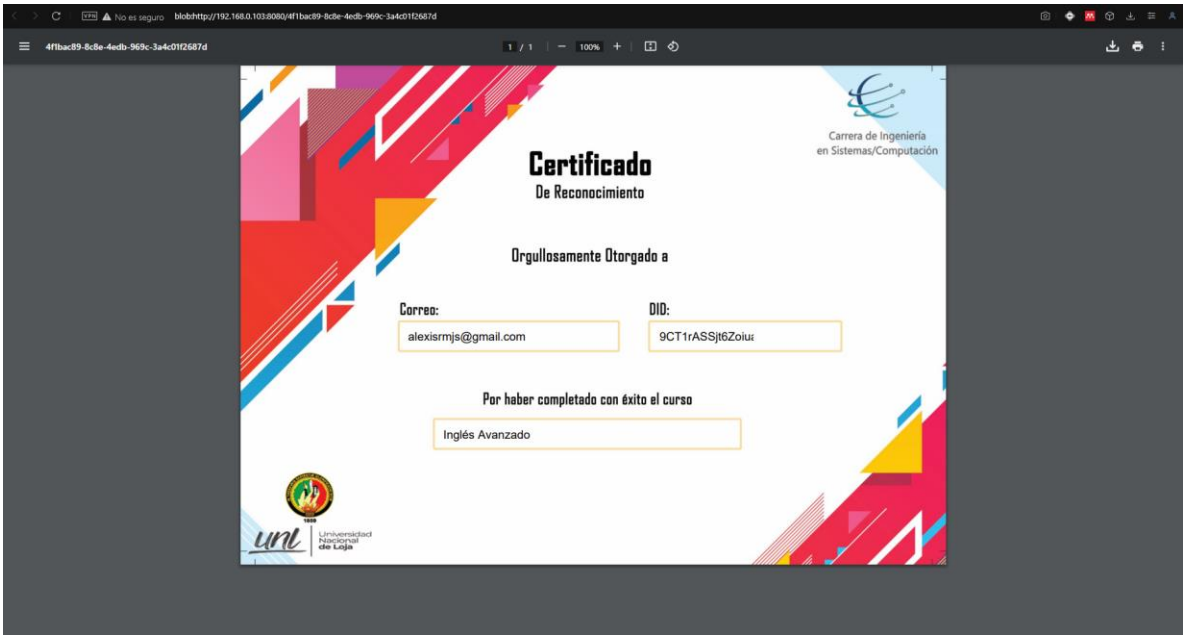
- Mensaje informativo sobre la obtención de los cursos de la página de la Carrera de Computación
- Vista del certificado generado del curso completado
- Mensaje informativo sobre la actualización de la información de los cursos del usuario

Resultado obtenido:

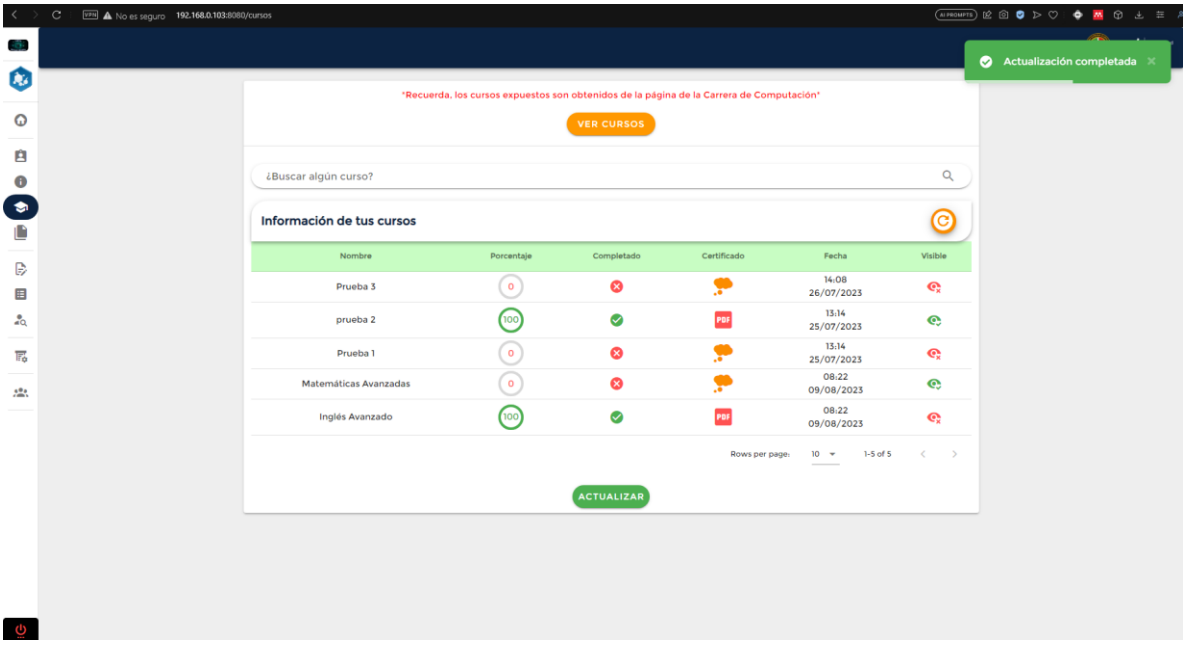
- Mensaje informativo sobre la obtención de los cursos de la página de la Carrera de Computación



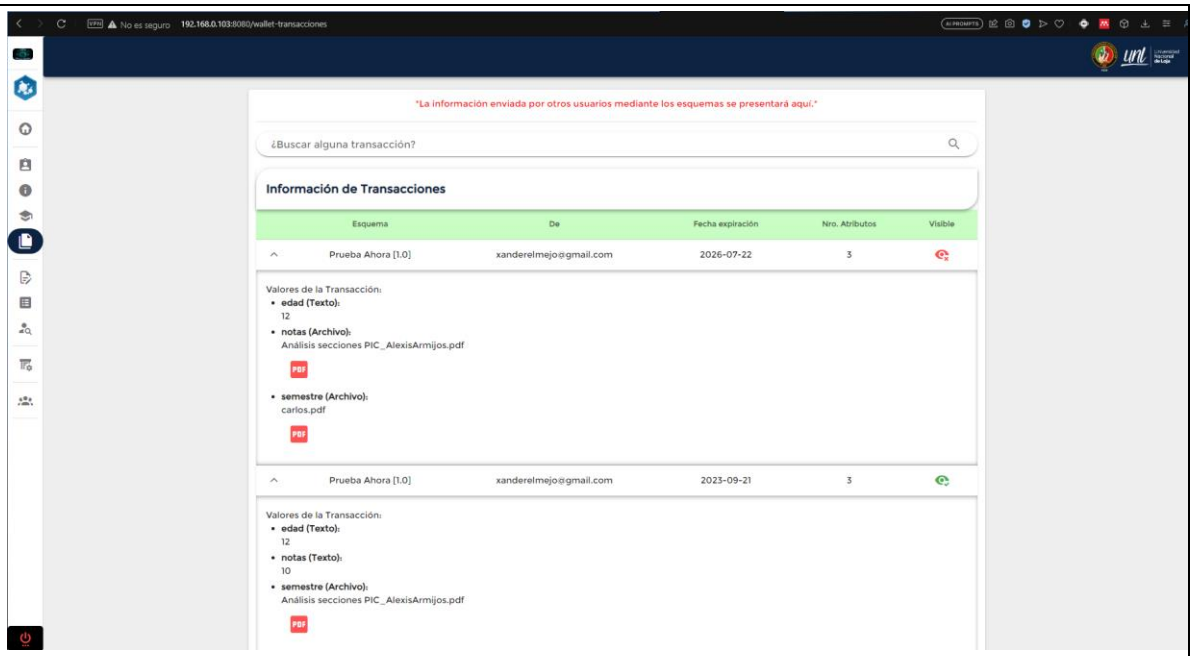
- Vista del certificado generado del curso completado



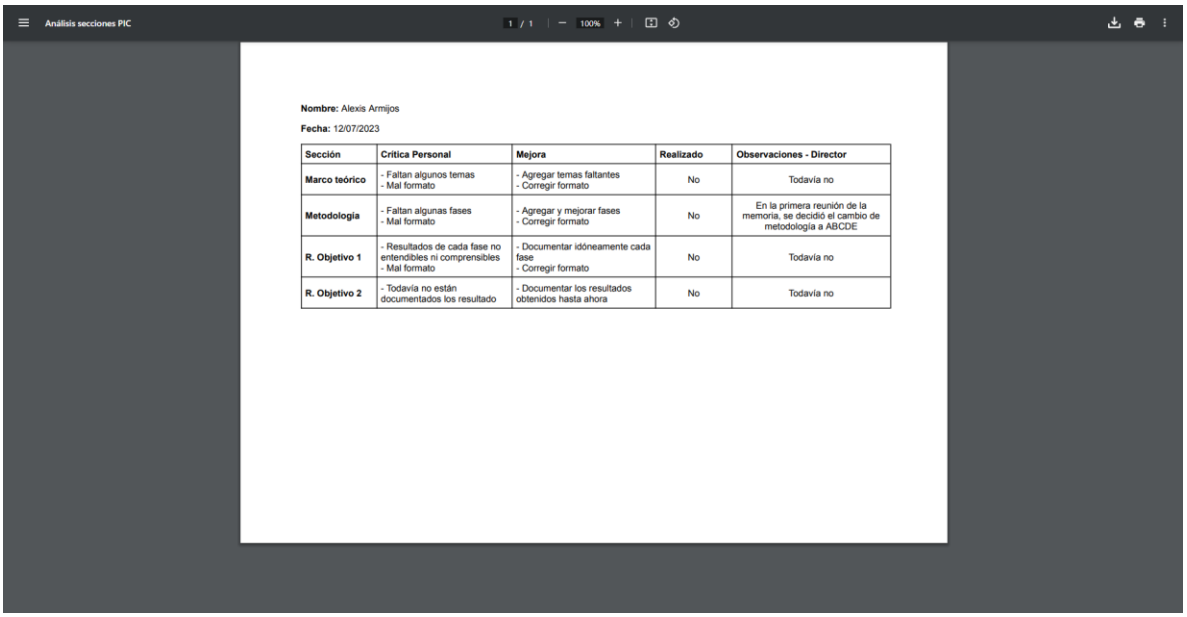
- Mensaje informativo sobre la actualización de la información de los cursos del usuario



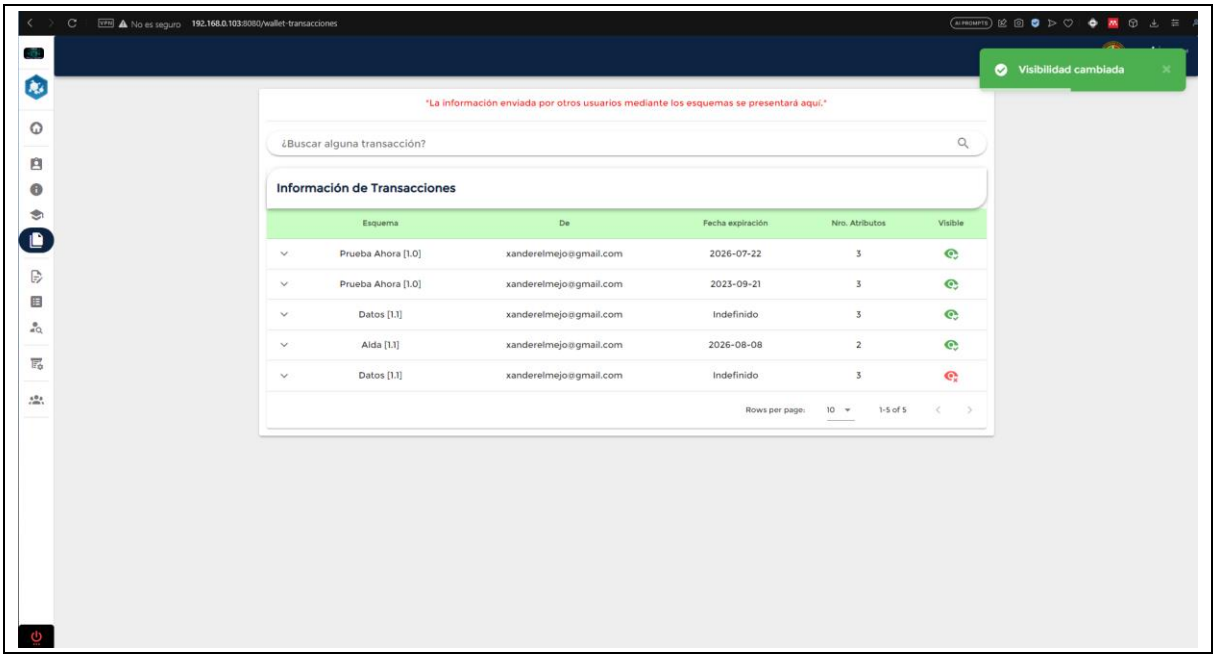
Administrar transacciones del usuario	**CPO6**	
	¿Prueba de despliegue?	Sí
Descripción: Se probará la respuesta del sistema cuando el usuario vaya a administrar sus transacciones		
Prerrequisitos: • Ingresar a la página. • Tener una sesión activa • Haber realizado transacciones utilizando los esquemas • Saber qué transacciones se va a administrar		
Pasos: 1. Escribir el enlace de la página web en el navegar 2. Iniciar sesión en el sistema 3. Hacer clic en la opción “Wallet - Transacciones” del menú de la izquierda 4. Esperar que cargue la información 5. Clic en el botón con el símbolo de una flecha hacia abajo que está a la izquierda de la columna en la tabla 6. Hacer clic en el símbolo de PDF de algún valor de la transacción de tipo Archivo 7. Cambiar la visibilidad de cualquier transacción al hacer clic en el símbolo con forma de ojo 8. Esperar el mensaje informativo		
Resultado esperado: • Se muestra la información sobre los valores de la transacción • Vista del PDF de algún valor de la transacción • Mensaje informativo sobre el cambio visibilidad de la transacción		
Resultado obtenido: • Se muestra la información sobre los valores de la transacción		



- Vista del PDF de algún valor de la transacción



- Mensaje informativo sobre el cambio visibilidad de la transacción



Utilizar los esquemas	**CPO7**	
	¿Prueba de despliegue?	Sí

Descripción:

Se probará la respuesta del sistema cuando el usuario vaya a utilizar algún esquema

Prerrequisitos:

- Ingresar a la página.
- Tener una sesión activa
- Debe haber esquemas disponibles
- Saber qué esquema utilizar

Pasos:

1. Escribir el enlace de la página web en el navegar
2. Iniciar sesión en el sistema
3. Hacer clic en la opción “Utilizar Esquema” del menú de la izquierda
4. Esperar que cargue la información
5. Cambiar el estado de solicitudes permitidas, al hacer clic en botón que está a la derecha
6. Hacer clic en el símbolo de un archivo con un lápiz para abrir el modal de utilizar esquema, en la columna Opciones según el esquema a utilizar
7. Seleccionar el usuario que proporcionará la información
8. Seleccionar y completar el tiempo de validez de la información
9. Esperar el mensaje informativo y redirección a la vista ver transacción
10. Clic en aceptar si eres el usuario destinatario (se define la transacción)
11. Clic en aceptar si eres el usuario remitente (se confirma la transacción)
12. Clic en aceptar si eres el usuario destinatario (se envía la información solicitada a la transacción)
13. Clic en aceptar si eres el usuario remitente (se acepta la información enviada a la transacción)
14. Esperar el mensaje informativo

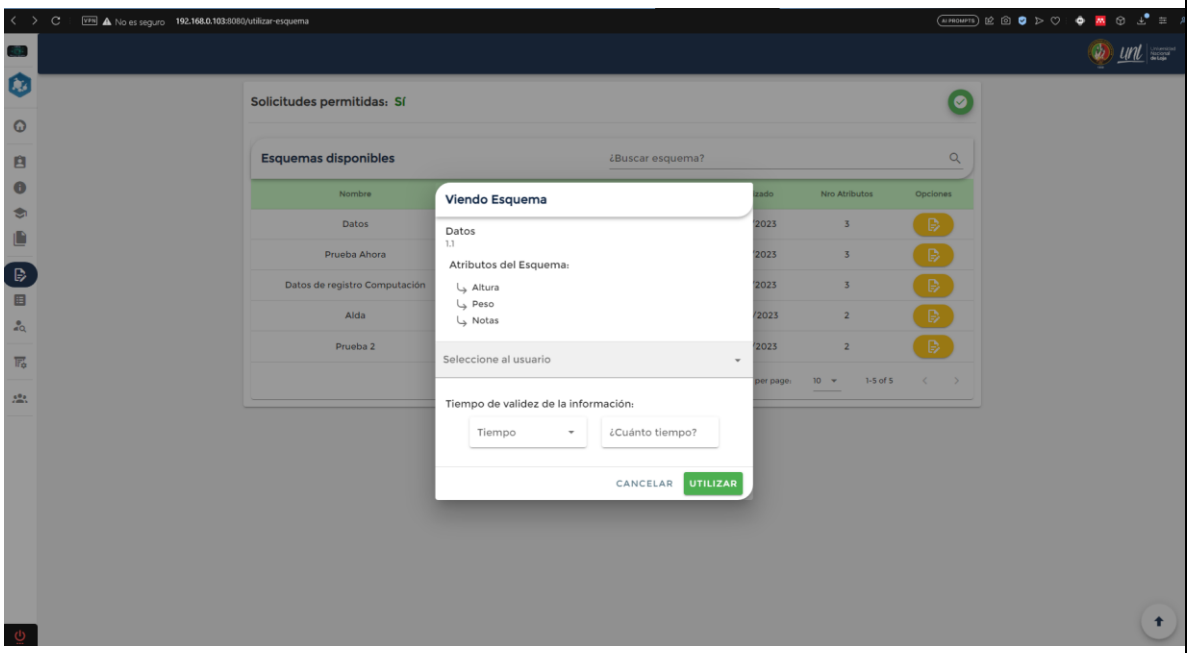
Resultado esperado:

- Ningún mensaje de error al abrir el modal para utilizar el esquema
- Ningún mensaje de error al utilizar el esquema y redirección a la vista ver transacción
- Mensaje informativo al aceptar la primera fase (siendo destinatario)
- Mensaje informativo al aceptar la segunda fase (siendo remitente)
- Ningún mensaje de error en los valores de envío de información
- Mensaje informativo al aceptar la tercera fase (siendo destinatario)

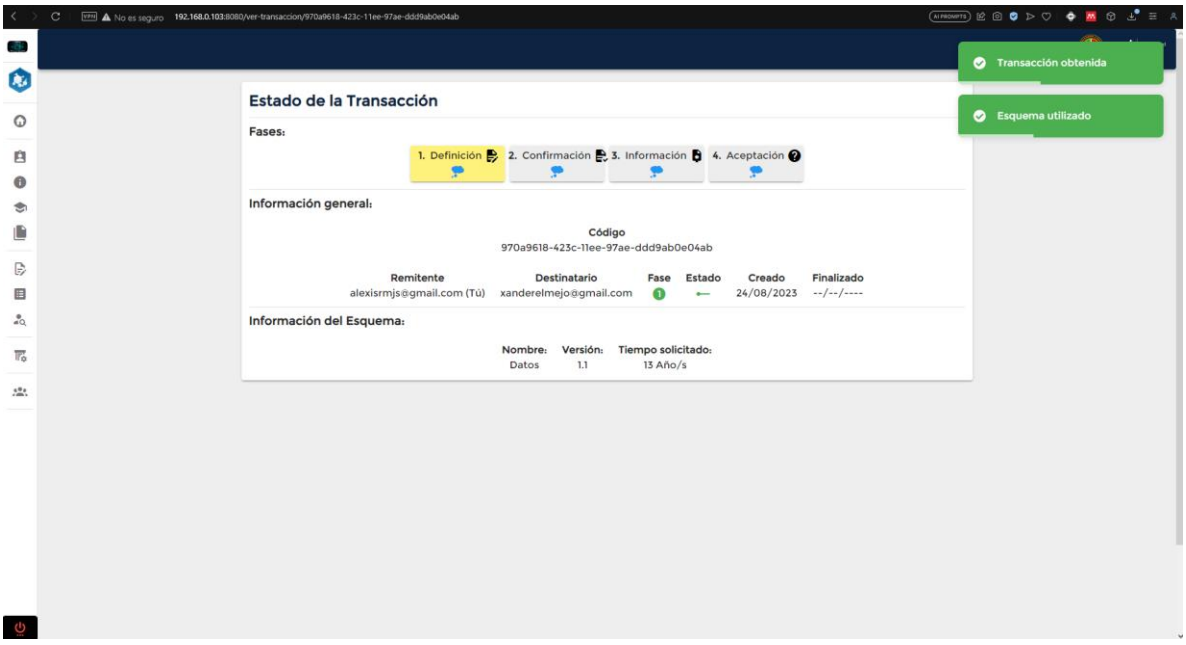
- Mensaje informativo al aceptar la cuarta fase (siendo remitente)

Resultado obtenido:

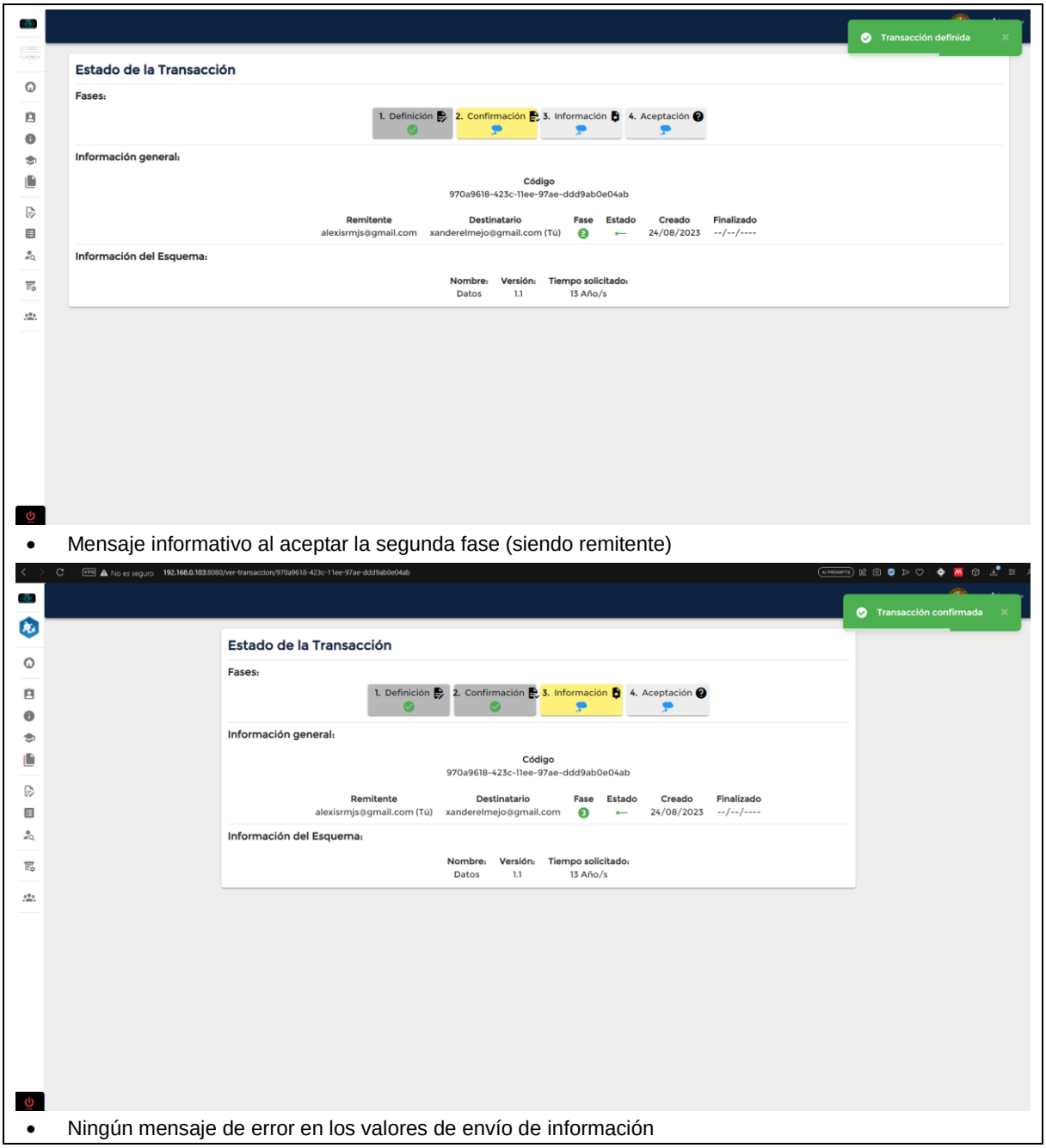
- Ningún mensaje de error al abrir el modal para utilizar el esquema



- Ningún mensaje de error al utilizar el esquema y redirección a la vista ver transacción

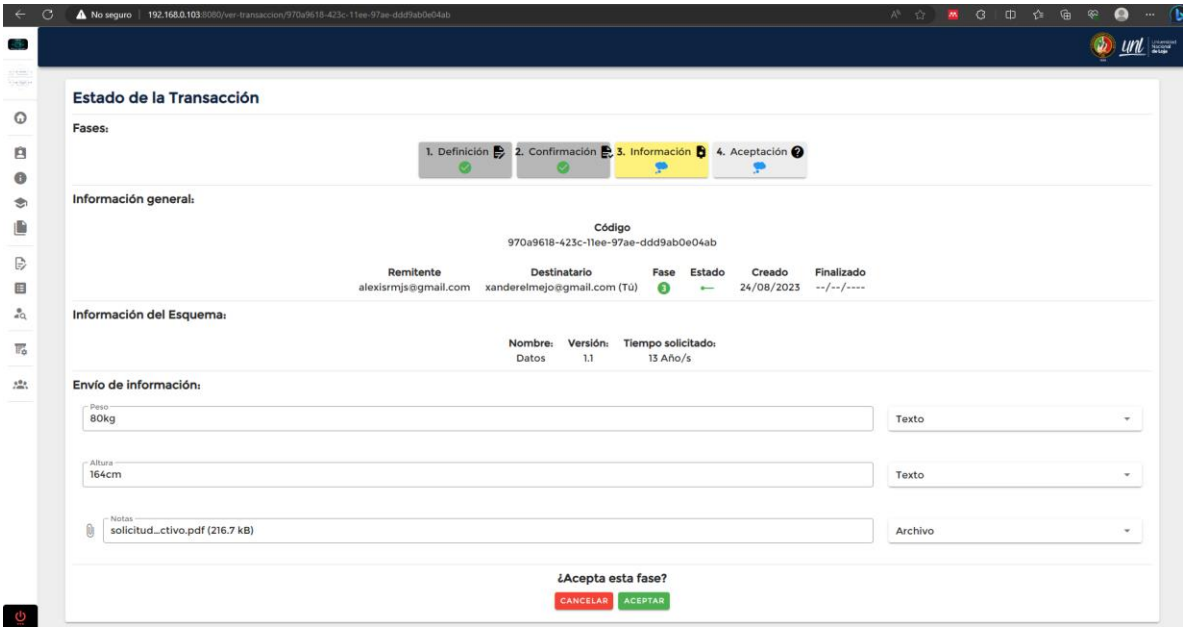


- Mensaje informativo al aceptar la primera fase (siendo destinatario)

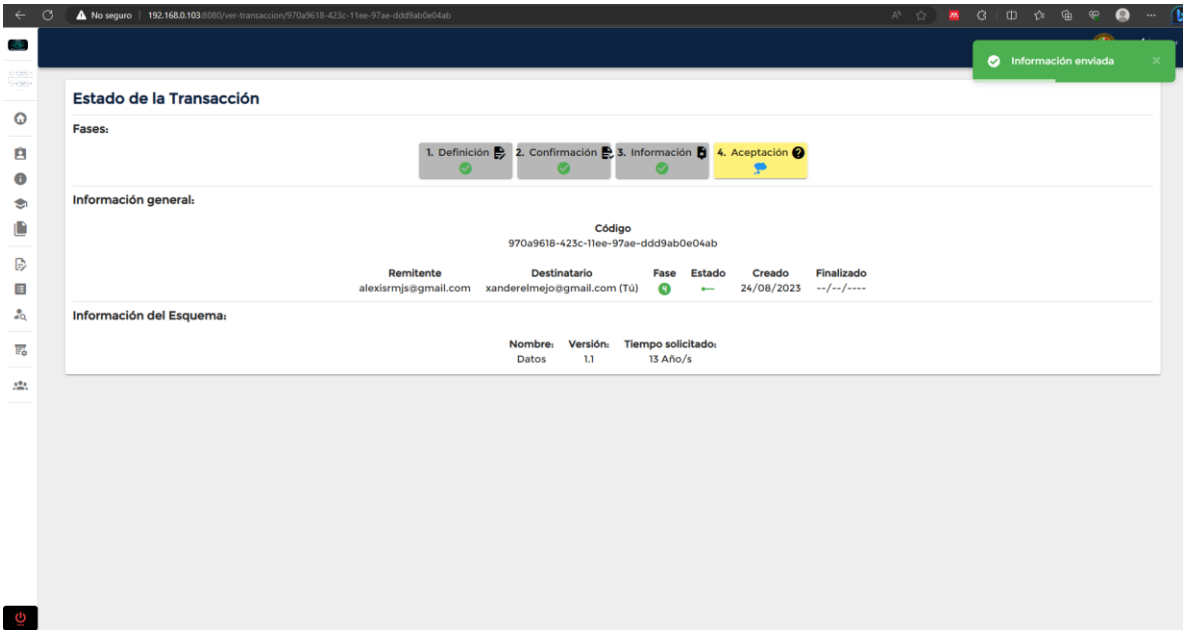


- Mensaje informativo al aceptar la segunda fase (siendo remitente)

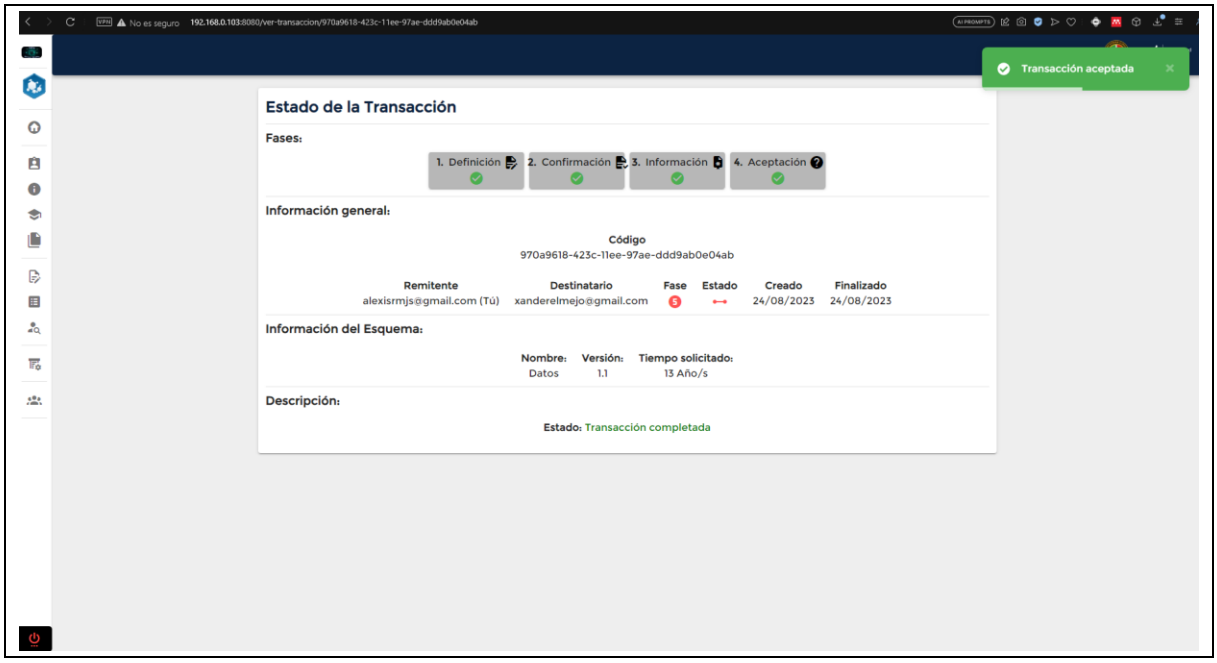
- Ningún mensaje de error en los valores de envío de información



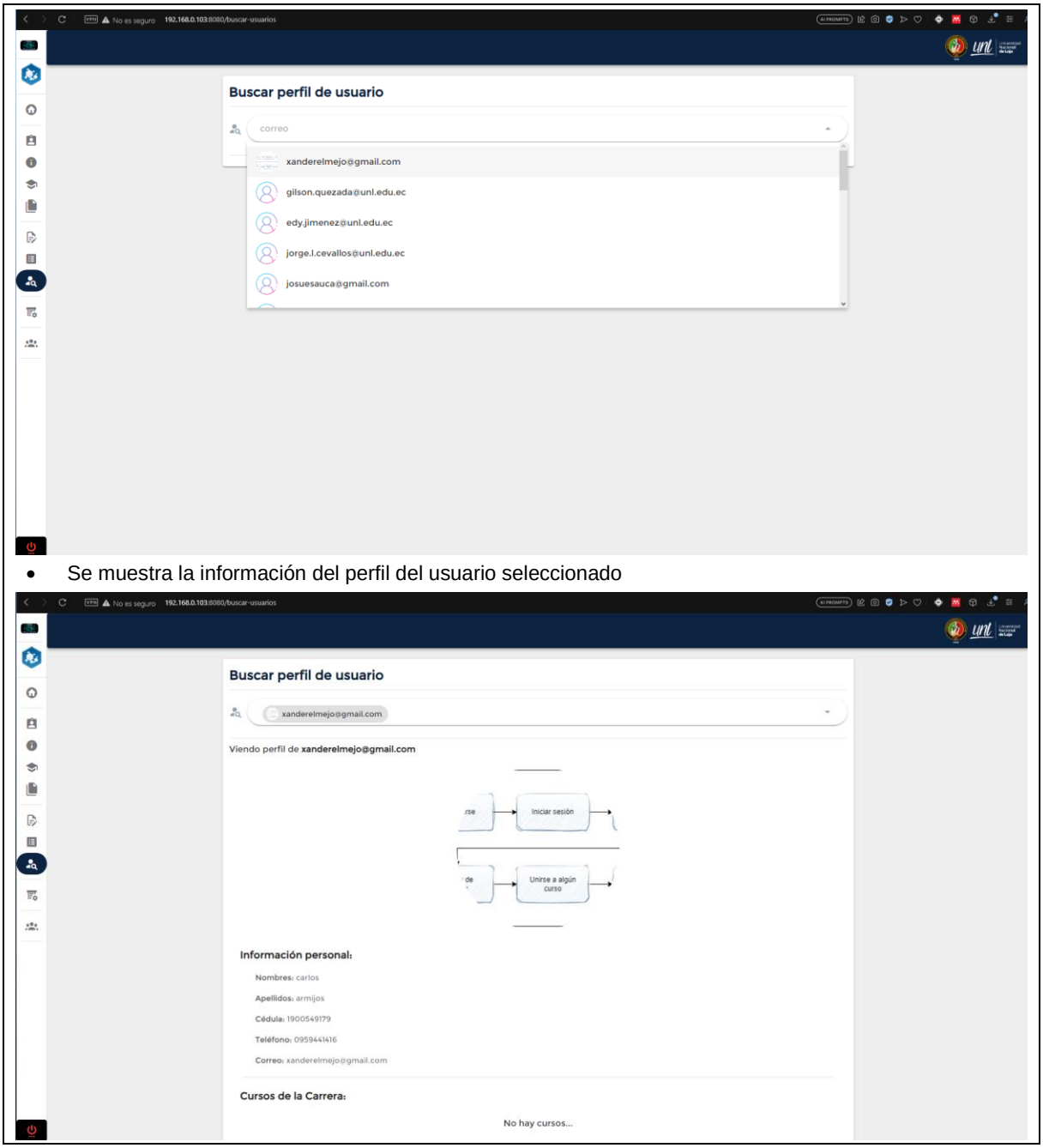
- Mensaje informativo al aceptar la tercera fase (siendo destinatario)



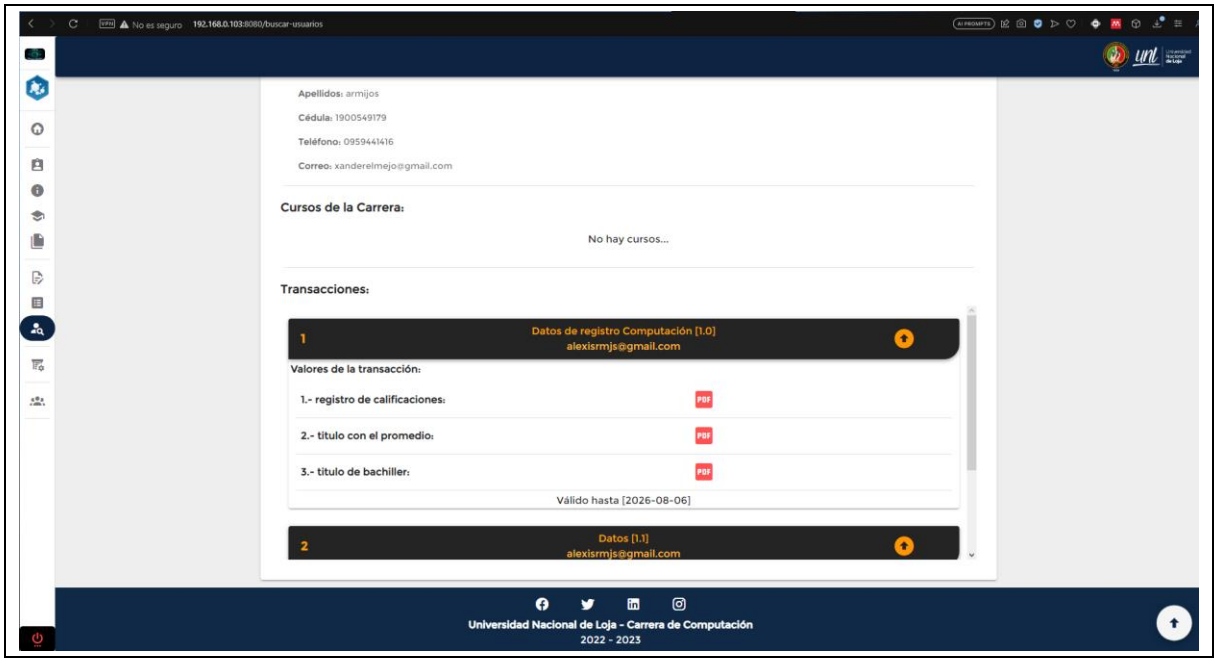
- Mensaje informativo al aceptar la cuarta fase (siendo remitente)



Buscar perfil de usuarios	**CPO8**	
	¿Prueba de despliegue?	Sí
Descripción: Se probará la respuesta del sistema cuando el usuario vaya a buscar el perfil de otro usuario		
Prerrequisitos: • Ingresar a la página. • Tener una sesión activa • Los otros usuarios deben tener su perfil público		
Pasos: 1. Escribir el enlace de la página web en el navegar 2. Iniciar sesión en el sistema 3. Hacer clic en la opción “Buscar usuarios” del menú de la izquierda 4. Esperar el mensaje informativo		
Resultado esperado: • Se muestran los usuarios disponibles • Se muestra la información del perfil del usuario seleccionado		
Resultado obtenido: • Se muestran los usuarios disponibles		



- Se muestra la información del perfil del usuario seleccionado



Administrar el perfil de información del usuario	**CPO9**	
	¿Prueba de despliegue?	Sí

Descripción:

Se probará la respuesta del sistema cuando el usuario vaya a administrar su perfil

Prerrequisitos:

- Ingresar a la página.
- Tener una sesión activa
- Tener información personal visible
- Tener información de cursos visible
- Tener información de transacciones visible

Pasos:

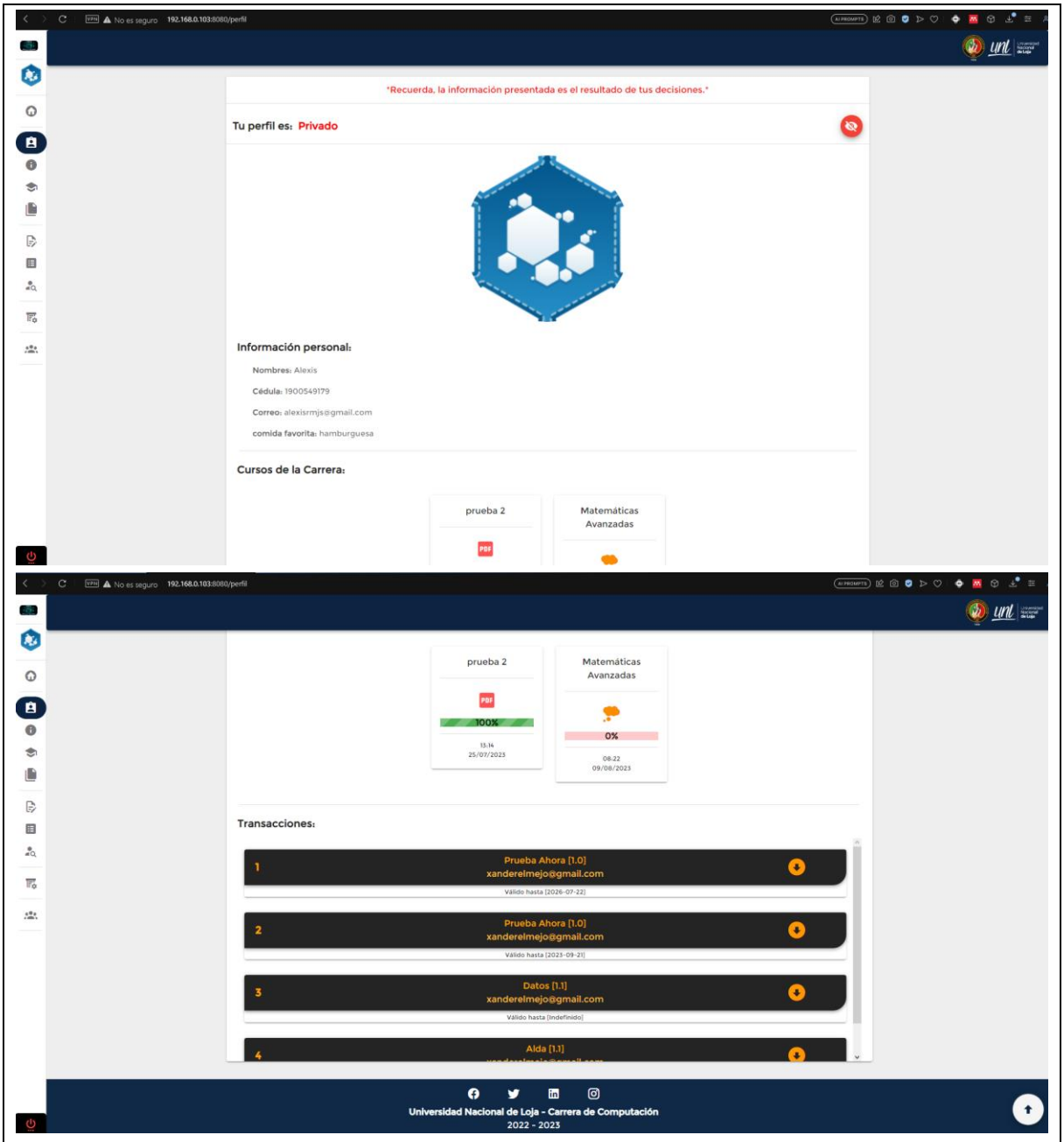
1. Escribir el enlace de la página web en el navegar
2. Iniciar sesión en el sistema
3. Hacer clic en la opción “Perfil” del menú de la izquierda
4. Cambiar el estado de del perfil, al hacer clic en botón que está a la derecha
5. Seleccionar el usuario para ver su perfil
6. Esperar el mensaje informativo

Resultado esperado:

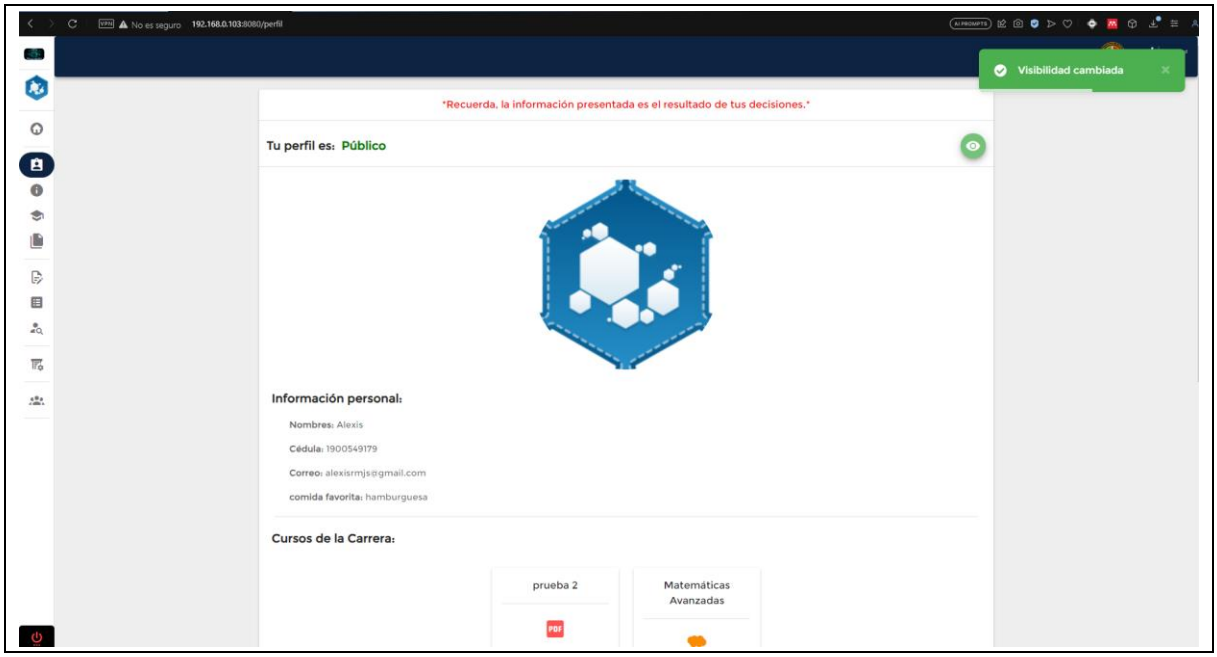
- Se muestra la información del perfil del usuario
- Se muestra un mensaje informativo al cambiar el estado del perfil

Resultado obtenido:

- Se muestra la información del perfil del usuario



- Se muestra un mensaje informativo al cambiar el estado del perfil



Administrar los esquemas	CP10	
	¿Prueba de despliegue?	Sí

Descripción:

Se probará la respuesta del sistema cuando el usuario vaya a administrar los esquemas

Prerrequisitos:

- Ingresar a la página.
- Tener una sesión activa
- Tener el rol de creador
- Saber que esquemas administrar

Pasos:

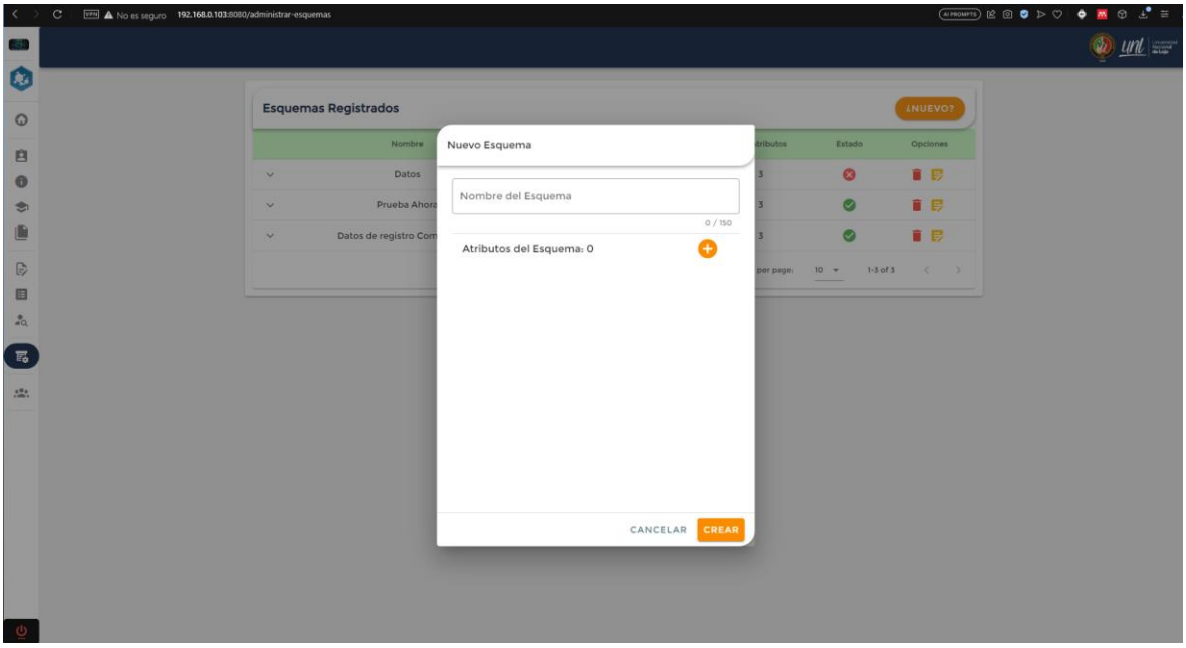
1. Escribir el enlace de la página web en el navegar
2. Iniciar sesión en el sistema
3. Hacer clic en la opción “Administrar Esquemas” del menú de la izquierda
4. Hacer clic en botón “¿Nuevo?” que está a la derecha del encabezado de la tabla, se abrirá un modal
5. Completar el Nombre del Esquema
6. Hacer clic en el botón un símbolo de +, para agregar un nuevo atributo al esquema
7. Completar el nuevo atributo agregado
8. Hacer clic en el botón “Crear”
9. Esperar mensaje informativo
10. Hacer clic en el símbolo de un archivo con un lápiz, en la columna Opciones según el esquema, abrirá un modal
11. Actualizar el Nombre del Esquema
12. Hacer clic en el botón un símbolo de +, para agregar un nuevo atributo al esquema
13. Completar el nuevo atributo agregado o actualizar los atributos existentes
14. Hacer clic en el botón “Actualizar”
15. Esperar mensaje informativo
16. Hacer clic en el símbolo de basurera, en la columna Opciones según el esquema lo borrará
17. Esperar mensaje informativo
18. Hacer clic en el símbolo de la columna Estado según el esquema, cambiará su disponibilidad de utilización para los usuarios
19. Esperar mensaje informativo

Resultado esperado:

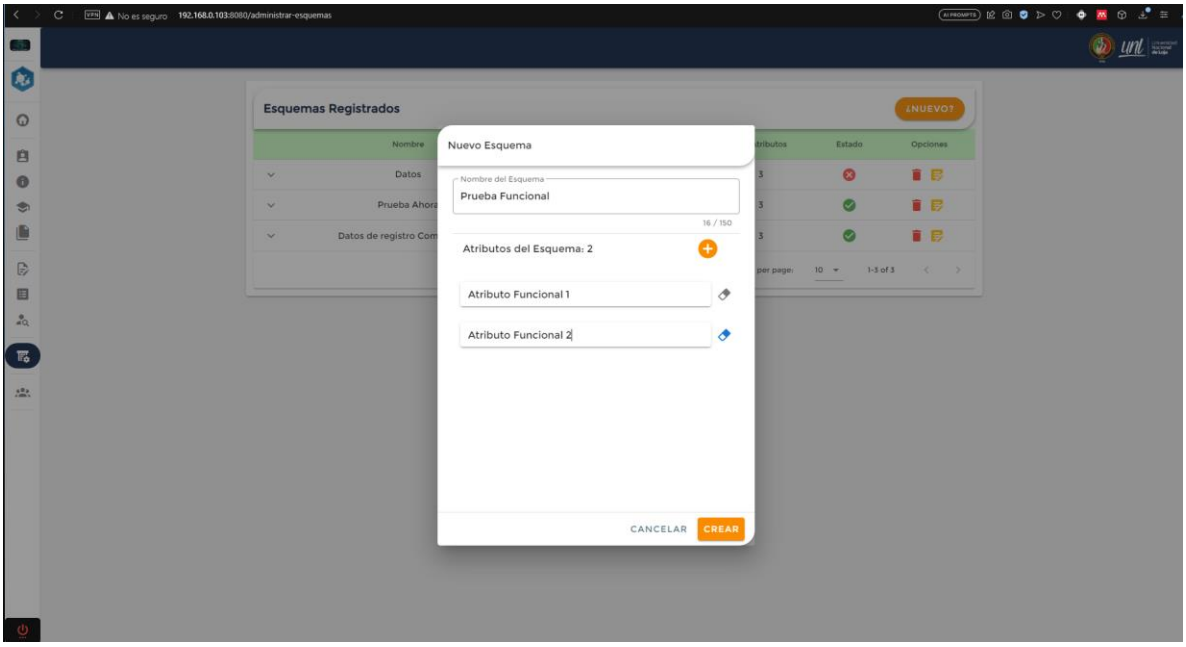
- Se muestra el modal al hacer clic en el botón “¿Nuevo?”
- Ningún mensaje de error antes de hacer clic en el botón “Crear”
- Mensaje informativo sobre el esquema creado
- Se muestra el modal al hacer clic en el símbolo de un archivo con el lápiz según el esquema
- Ningún mensaje de error antes de hacer clic en el botón “Actualizar”
- Mensaje informativo sobre el esquema creado
- Hacer clic en el símbolo del basurero según el esquema, mostrará mensaje informativo de eliminación
- Hacer clic en el símbolo de la columna Estado según el esquema, mostrará mensaje informativo de cambio de estado

Resultado obtenido:

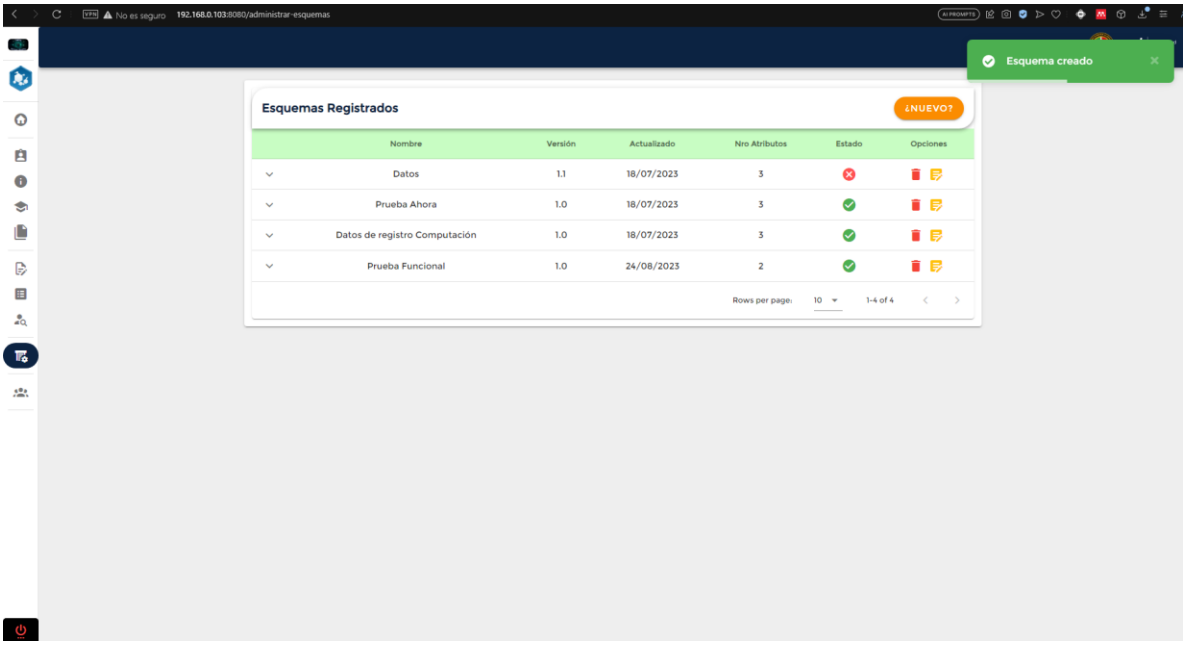
- Se muestra el modal al hacer clic en el botón “¿Nuevo?”



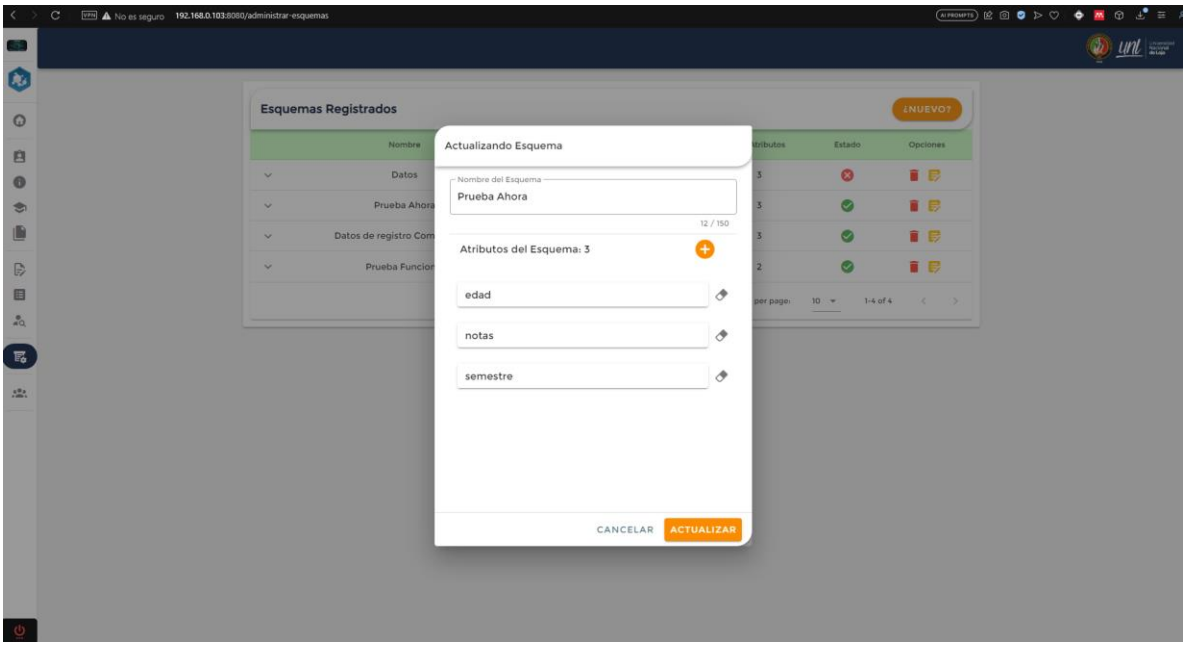
- Ningún mensaje de error antes de hacer clic en el botón “Crear”



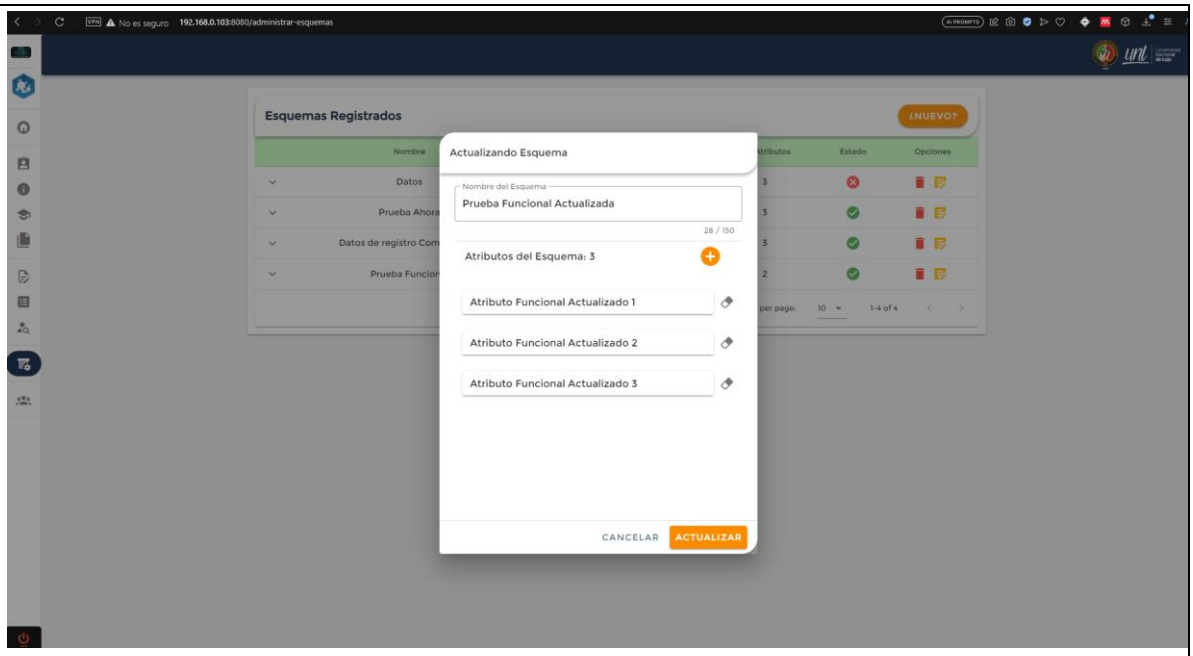
- Mensaje informativo sobre el esquema creado



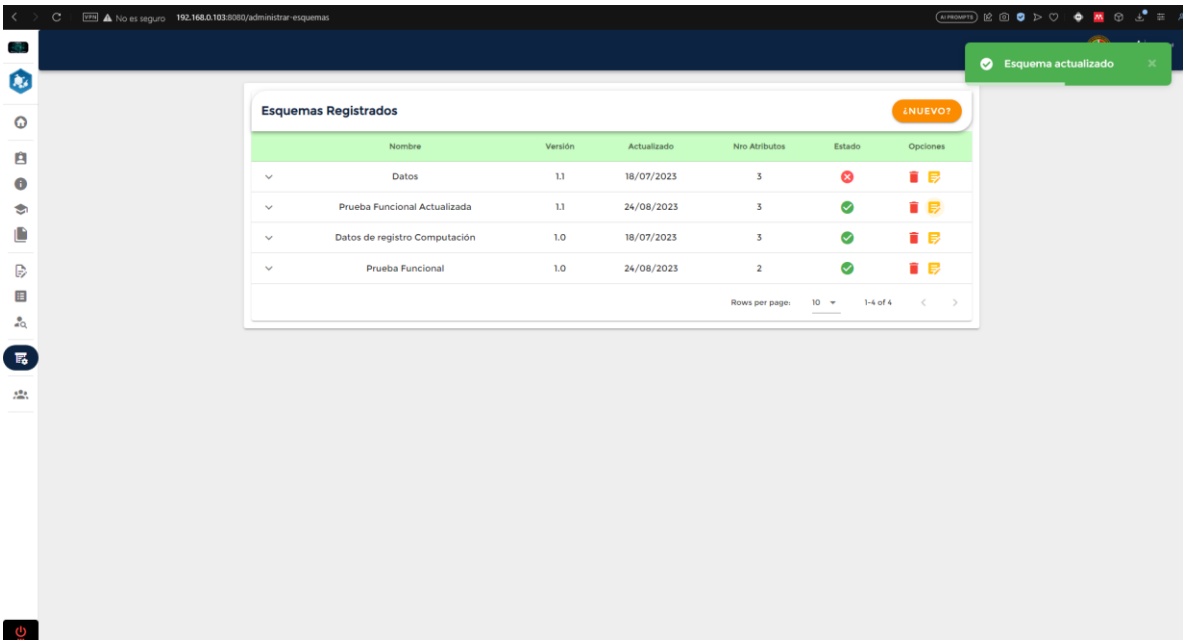
- Se muestra el modal al hacer clic en el símbolo de un archivo con el lápiz según el esquema



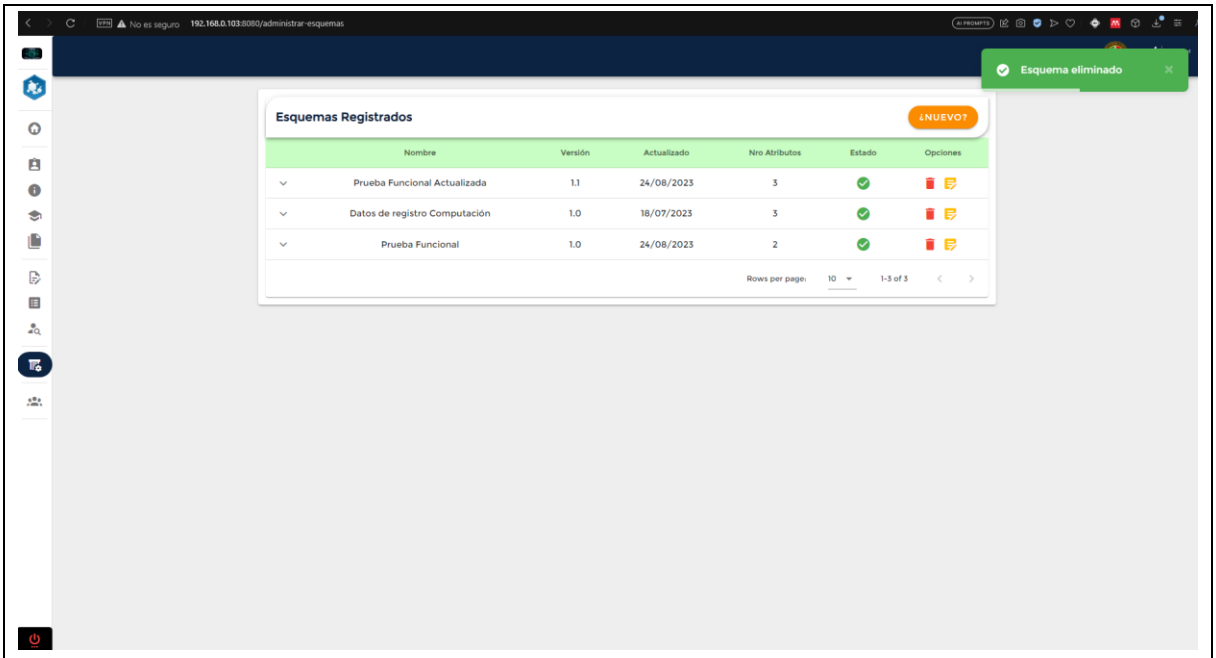
- Ningún mensaje de error antes de hacer clic en el botón “Actualizar”



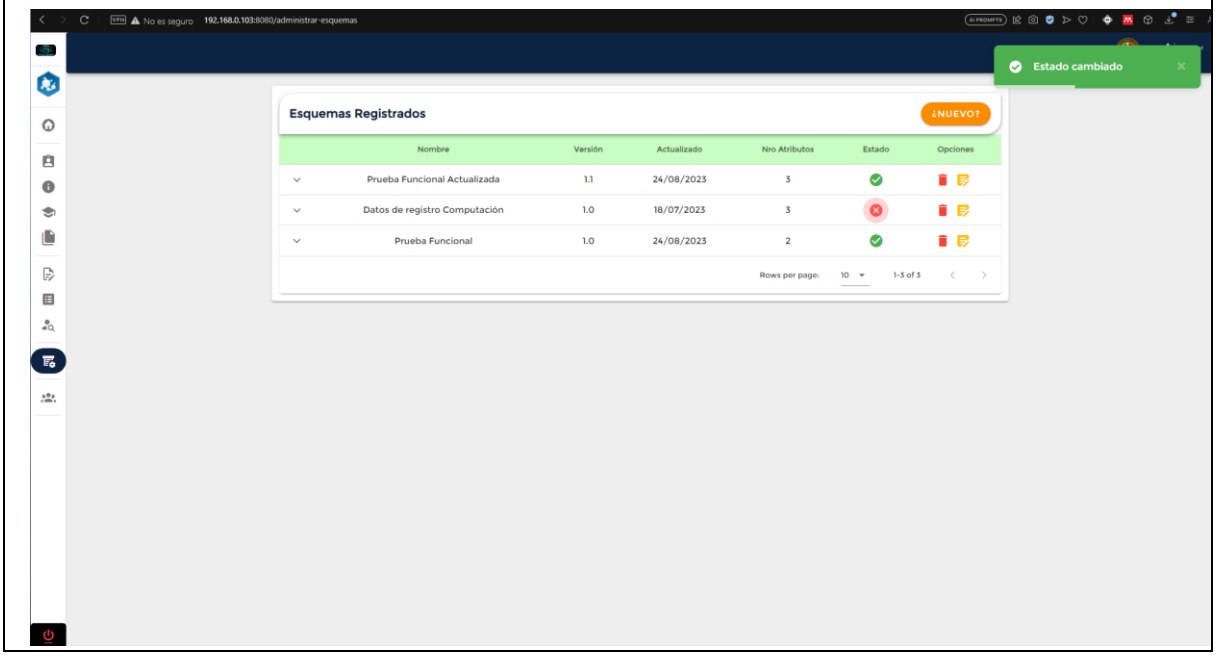
- Mensaje informativo sobre el esquema actualizado



- Mensaje informativo de eliminación, al hacer clic en el símbolo del basurero según el esquema



- Mensaje informativo de cambio de estado, al hacer clic en el símbolo de la columna Estado según el esquema



Administrar usuarios	**CP11**	
	¿Prueba de despliegue?	Sí
Descripción: Se probará la respuesta del sistema cuando el usuario vaya a administrar a los usuarios		
Prerrequisitos: • Ingresar a la página. • Tener una sesión activa • Tener el rol administrador • Saber que usuarios administrar		
Pasos:		

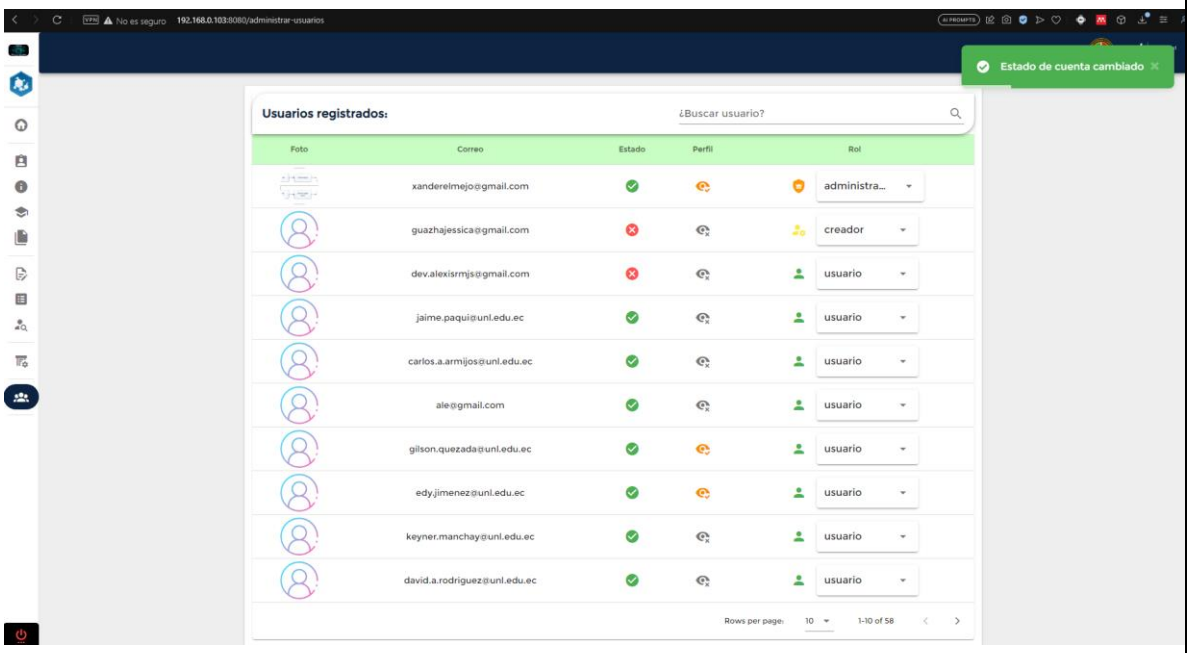
1. Escribir el enlace de la página web en el navegar
2. Iniciar sesión en el sistema
3. Hacer clic en la opción “Administrar usuarios” del menú de la izquierda
4. Hacer clic en el símbolo de la columna Estado según el usuario, mostrará mensaje informativo de cambio de estado de la cuenta
5. Seleccionar otro rol de la columna Rol según el usuario, mostrará mensaje informativo de cambio de rol

Resultado esperado:

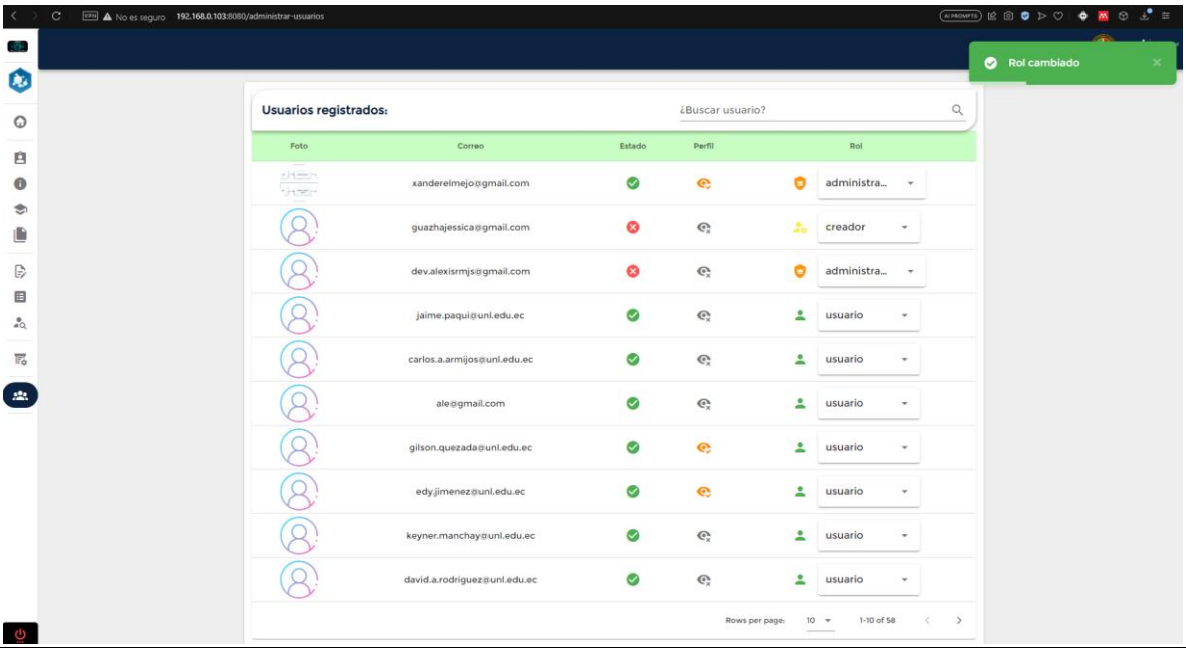
- Mensaje informativo de cambio de estado de cuenta, al hacer clic en el símbolo de la columna de Estado según el usuario
- Mensaje informativo de cambio de rol, al seleccionar otro rol de la columna Rol según el usuario

Resultado obtenido:

- Mensaje informativo de cambio de estado de cuenta, al hacer clic en el símbolo de la columna de Estado según el usuario



- Mensaje informativo de cambio de rol, al seleccionar otro rol de la columna Rol según el usuario



4 Estrategia de ejecución de pruebas

En este apartado se indica los diferentes casos de prueba que se realizan en cuatro ciclos.

	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4
CP01		x		
CP02	x			
CP03		x		
CP04		x		
CP05		x		
CP06			x	
CP07		x		
CP08				x
CP09			x	
CP10		x		
CP11		x		

5 Glosario

A continuación, se muestran las definiciones de todos los términos utilizados en el presente documento.

Término	Descripción
Esquema	Es el Esquema de Transcripción (también conocido como contrato inteligente) que proporciona Hyperledger Indy para que los usuarios puedan intercambiar información válida por medio de su blockchain
Wallet	Es la billetera virtual que tiene el usuario de la blockchain de Hyperledger Indy, dentro de ella podrá manipular su propia información.
PDF	Formato de Documento Portátil (Portable Document Format)

6 Bibliografía y referencias

Referencia	Título
1	Plantilla de Plan de Pruebas Funcionales de la Junta de Andalucía (https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/462)
2	Anexo 1: Especificación de requisitos de software 1 (Del Documento del Proyecto de Integración Curricular)

Anexo 9: Plan de Pruebas de Aceptación.

Plan de Pruebas de Aceptación

Proyecto: Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja

Versión: 2.0

Fecha: 25/08/2023

Autor: Alexis Armijos
Correo electrónico: carlos.a.armijos@unl.edu

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja		
Entregable	Plan de Pruebas de Aceptación		
Autor	Carlos Alexis Armijos Rios		
Versión/Edición	2.0	**Fecha Versión**	24/08/2023
Aprobado por	Cristian Ramiro Narváez Guillen, Mg. Sc.	**Fecha Aprobación**	24/08/2023
		N.º Total de Páginas	12

Registro de cambios

Versión doc.	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas de Aceptación	Carlos Alexis Armijos Rios	23/08/2023
2.0	Versión final del Plan de Pruebas de Aceptación	Carlos Alexis Armijos Rios	24/08/2023

Control de distribución

Nombre y Apellidos
Carlos Alexis Armijos Rios
Cristian Ramiro Narváez Guillen, Mg. Sc

Introducción

Objetivo

El objetivo de este documento es conocer el nivel de aceptación del sistema a partir de la aplicación de encuesta a una muestra de estudiantes de la Universidad Nacional de Loja, para validar el desempeño de las funcionalidades construidas en base a los requerimientos establecidos para el sistema.

Propósito

Comprobar que el nivel de aceptación del sistema cumple con las ponderaciones mínimas, aplicando encuesta hacia una muestra de personas de la Universidad Nacional de Loja.

Parámetros de evaluación

Nro.	Pregunta
1	¿Es simple el vocabulario utilizado?
2	¿Se proporciona el tiempo suficiente para realizar las entradas de información?
3	¿Se entienden la interfaz y su contenido?
4	¿Resulta fácil identificar alguna acción?
5	¿Resulta fácil entender el resultado de una acción?
6	¿Está diseñada la interfaz para la realización de las tareas de forma eficiente?
7	¿Son apropiados los mensajes presentados por el sistema?
8	¿Actúa el sistema en la prevención de errores?
9	¿El sistema informa claramente sobre los errores presentados?
10	¿Permite una cómoda navegación dentro del sistema y una fácil salida de este?
11	¿Se presenta al usuario la información que sólo necesita?

Así mismo, se plantea una tabla para determinar el estado de las Pruebas de Aceptación:

Estado de las Pruebas de Aceptación	Criterio		
	Sí	Parcialmente	No
Positivo	>=70$	<=30%	<=3%
Negativo	<70$	>30%	>3%

Encuestas

La encuesta presentada a continuación tiene por objetivo determinar el nivel de aceptación del sistema de identidad digital auto-gestionada desarrollado, para ello se hacen uso de 11 preguntas que permiten comprobar si se cumplen con las funcionalidades creadas en base a los requerimientos establecidos. La encuesta se elaboró utilizando los formularios de la Suite de Google, y se aplicó a una muestra de 50 estudiantes de la Universidad Nacional de Loja, pertenecientes a la Carrera de Ingeniería en Computación (ver Figura 1).

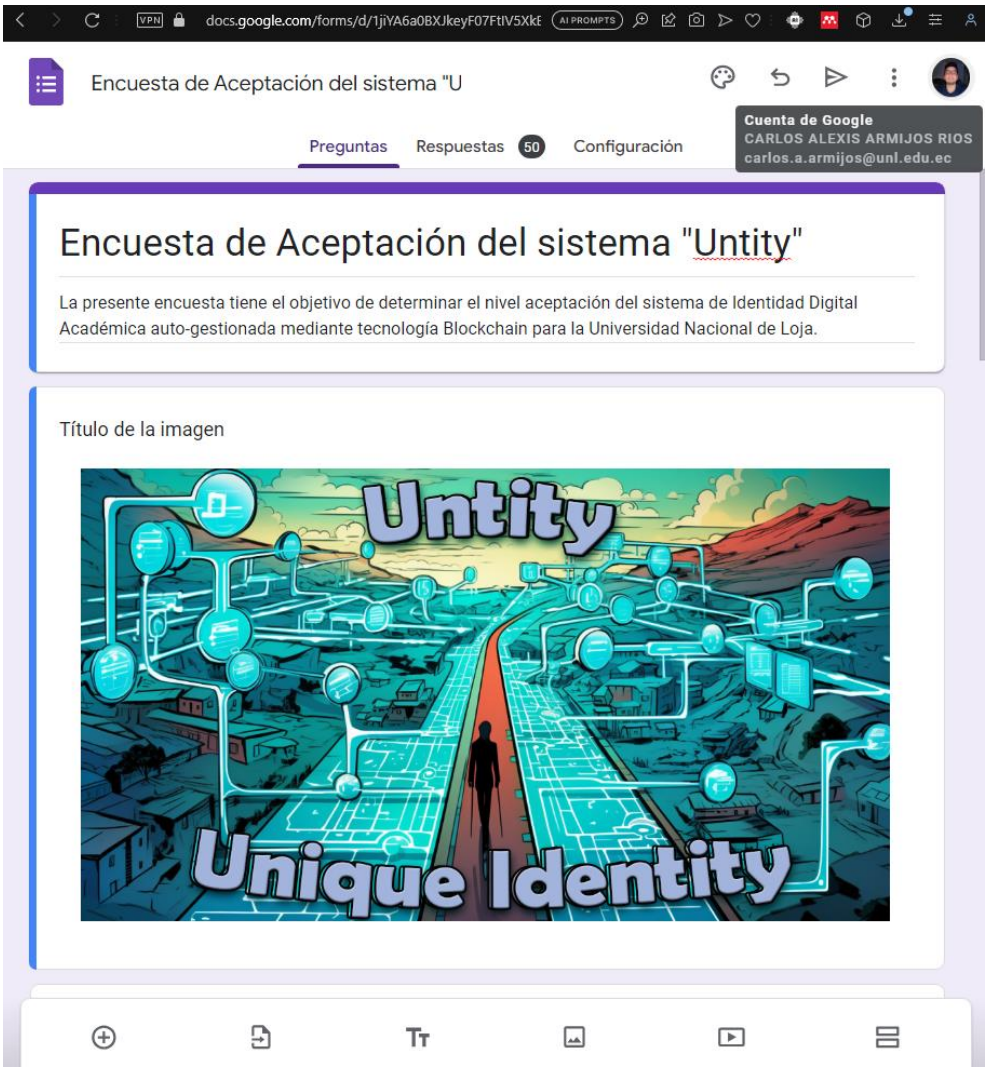


Figura 1: *Encuesta de Aceptación del sistema.*

La encuesta generó los siguientes resultados que se representan en gráficos y porcentajes elaborados automáticamente por Google Forms.

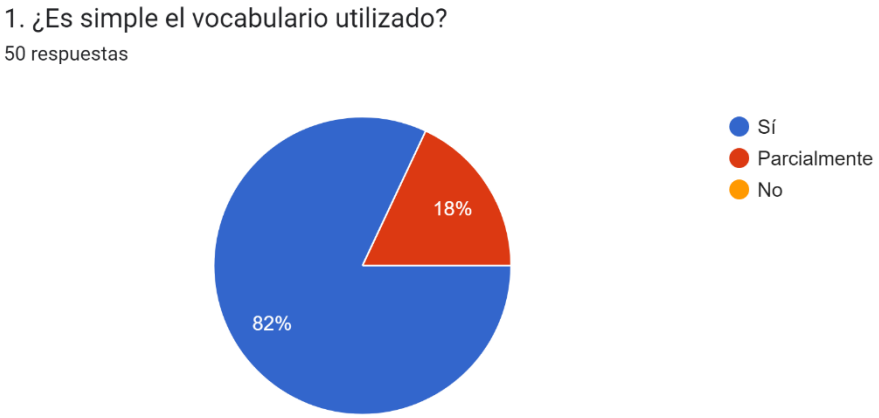


Figura 1: *Pregunta evaluadora número 1.*

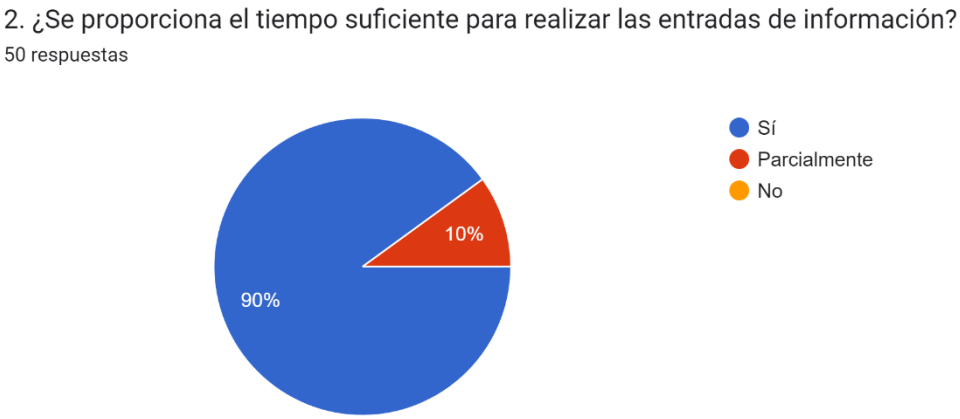


Figura 2: *Pregunta evaluadora número 2.*

Figura 3: *Pregunta evaluadora número 3.*

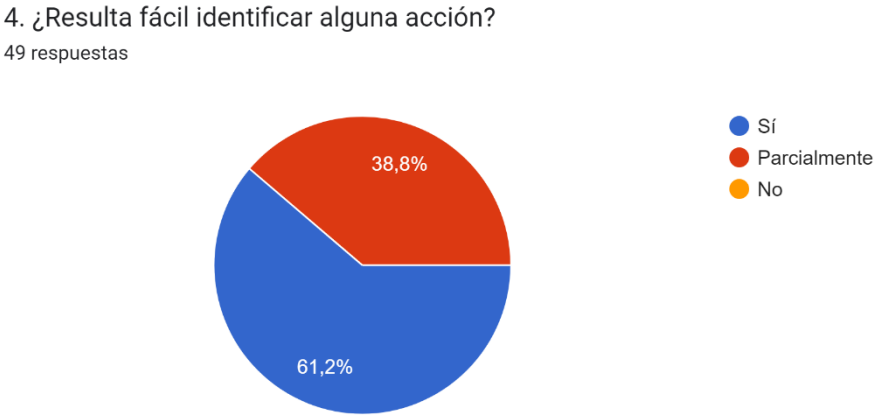


Figura 4: *Pregunta evaluadora número 4.*

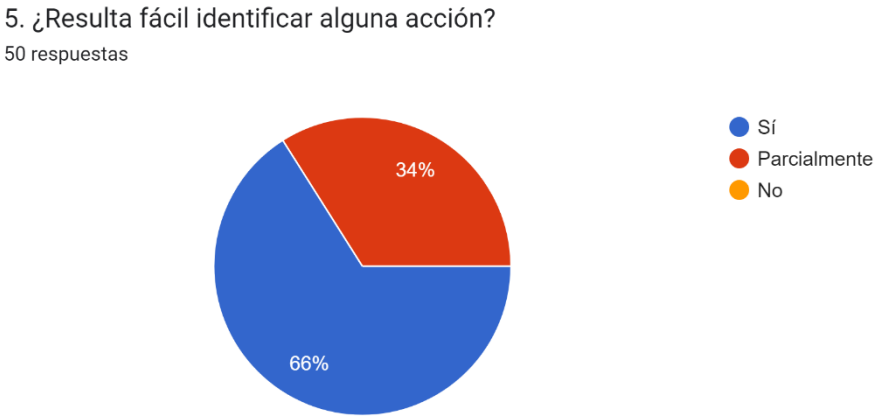


Figura 5: *Pregunta evaluadora número 5.*

6. ¿Está diseñada la interfaz para facilitar la realización de las tareas de forma eficiente?
50 respuestas

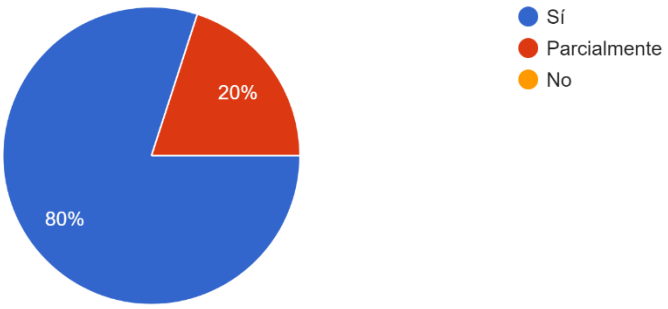


Figura 6: *Pregunta evaluadora número 6.*

7. ¿Son apropiados los mensajes presentados por el sistema?
50 respuestas

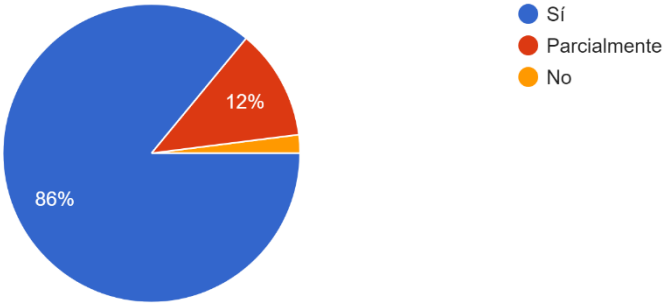


Figura 7: *Pregunta evaluadora número 7.*

8. ¿Actúa el sistema en la prevención de errores?
50 respuestas

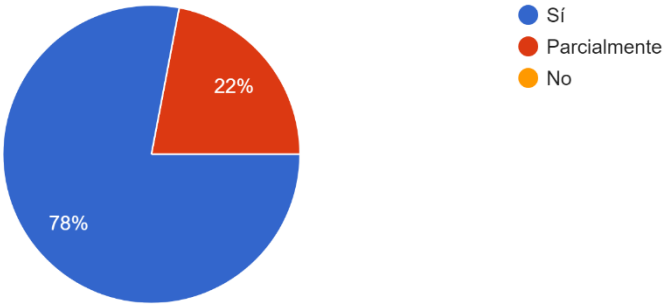


Figura 8: *Pregunta evaluadora número 8.*

9. ¿El sistema informa claramente sobre los errores presentados?
50 respuestas

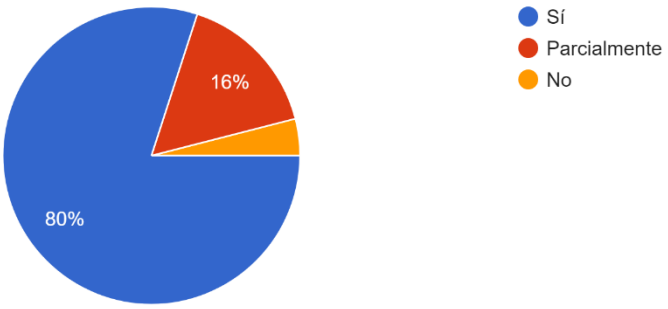


Figura 9: *Pregunta evaluadora número 9.*

10. ¿Permite una cómoda navegación dentro del sistema y una fácil salida de este?
50 respuestas

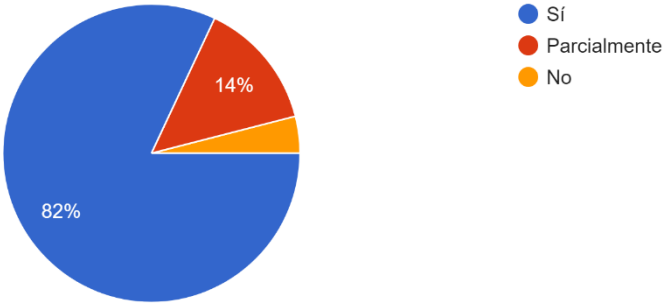


Figura 10: *Pregunta evaluadora número 10.*

11. ¿Se presenta al usuario la información que sólo necesita?
50 respuestas

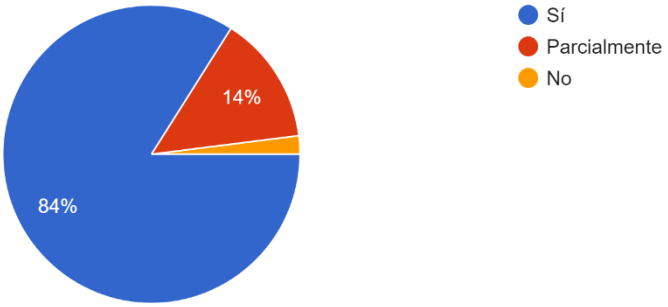


Figura 11: *Pregunta evaluadora número 11.*

Aplicación del criterio de aceptación

En este apartado se indican los resultados de los promedios de las 3 respuestas disponibles en cada pregunta de la encuesta.

Id. Pregunta Evaluadora	Resultados		
	Sí	Parcialmente	No
PE01	82%	18%	0%
PE02	90%	10%	0%
PE03	66%	28%	6%
PE04	61.2%	38.8%	0%
PE05	66%	34&	0%
PE06	80%	20%	0%
PE07	86%	12%	2%
PE08	78%	22%	0%
PE09	80%	16%	4%
PE10	82%	14%	4%
PE11	84%	14%	2%
Promedio	77.74%	20.61%	1.45%

Con los promedios obtenidos de los criterios sí (77.74%), parcialmente (20.61%) y no (1.45%), se determina que el nivel de aceptación del sistema de identidad digital auto-gestionada tiene un estado positivo.

Bibliografía y referencias

Referencia	Título
1	Anexo 1: Especificación de Requisitos de Software (del Documento Proyecto de Integración de Curricular)

Anexo 10: Paper TICEC 2024.

Decentralized Academic Identity: Building Self-Managed Profiles with Hyperledger Indy [1]

Armijos, Alexis[1][0000−0002−6305−3439], Narvaez, Cristian[2][0000−0002−9096−1010], Torres, Hernan[3][2222−−3333−4444−5555], and Cueva, Mario[4][2222−−3333−4444−5555]

Universidad Naciona de Loja, Loja, Ecuador https://www.unl.edu.ec

Abstract. This paper explores the implementation of a DApp for selfmanaged academic digital identity using Hyperleadger Indy. It discusses the fundamental principles of self-managed digital identities, highlighting their self-managed digital identities, and highlighting their importance in the educational environment. It discusses a secure decentralized framework and the role of Hyperledger Indy in managing verifiable credentials and certificates, providing a safe and decentralized framework. The acceptance of university students and the enhanced privacy and control of personal data are discussed, and a potential use case for the educational institution is presented. The study concludes by assessing the adoption, challenges, and opportunities involved in securing digital identity in the academic sector.

Keywords: Blockchain · distributed leadger technology(DLT) · Indy · Hyperledger · Digital Identity · DID

Introduction

The user's digital identity is based on personal information such as addresses, jobs, names, and social networks [1]. There is no mechanism to ensure that this identity cannot be disclosed, exposed, or altered, leading to potential leaks without consent [7].

In contemporary times, a considerable number of students are actively seeking information on academic topics via the Internet. Numerous websites require personal information for access, yet assure users of the security of their provided data. Consequently, to address this issue of data insecurity, the concept of self-sovereign digital identity has emerged as an innovative solution [8]. This paradigm shift aims to endow users with comprehensive control over their digital identity, ensuring a level of validity and trust that parallels their physical world identity.

The self-managed digital identity gives users complete control over their information without needing approval from an administrative entity [8]. This secure technology ensures that digital identities are only exposed with user authorization. Hyperledger Indy [17]excels in decentralized identity management, using blockchain, libraries, and methods to keep user information safe from unauthorized access or modification [15].

This paper [14] presents a DApp for self-managed academic digital identity using decentralized identifiers (DIDs) provided by Hyperledger Indy. The ABCDE methodology [9] is used to create a secure and efficient hybrid application.

Ultimately, the outcome is that individuals who utilize the self-administered digital identity DApp will possess the capability to govern their data, disseminate it with their explicit consent, and have the assurance that it cannot be divulged or altered by unauthorized entities. This security is assured by the Hyperledger Indy Blockchain.

[1] Universidad Nacional de Loja

1.1 Digital Identity

Digital identity encompasses various categories of data contingent on the user's discretion to disclose them, culminating in a public identity encompassing selfdisclosed information, an active identity derived from the individual's actions, and an inferred identity formulated through societal analysis of these actions [1]. This aggregation of information facilitates the construction of a conceptual understanding of the individual's persona.

In detail, the categories of data instrumental in constructing this digital identity are classified as follows:

- Personally identifiable information: entails identifiable attributes such as name, identification number, driver's license number, credit card information, date of birth, and social media identifiers related to website interactions.
- Behavioral data: encompasses transactional records, browsing histories, geolocation data, call logs, purchase patterns, and similar activities.
- Derived data: consists of analytically derived attributes used to profile individuals, for example, to assess creditworthiness for loan applications or evaluate their influence within a particular domain.

1.2 Self-managed Identity Model

The self-managed digital identity model (Fig.1) encompasses three core components that enable the user to exercise comprehensive oversight over the information that constitutes their digital identity: individual control, security, and portability. The transition from a centralized information system (databases) to a decentralized information system (blockchain) empowers users to possess, regulate, and administrate their identity autonomously. Consequently, external entities are precluded from altering, supplying, or expunging information pertaining to the user's identity.

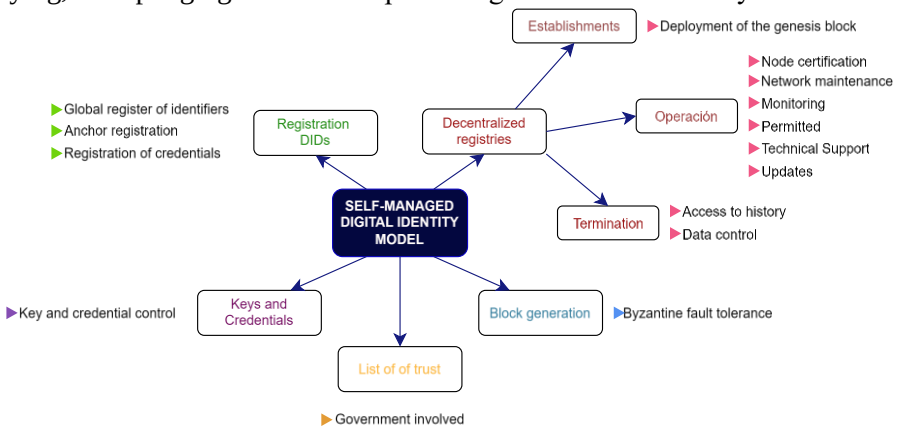


Fig.1. Self-managed Identity Model.

Within this framework, the user is empowered to disclose identity information selectively or in full, according to their preference. Furthermore, contingent upon their consent, they may disseminate their information to other users, contextually aligned with specific transaction events. These information exchanges are restricted exclusively to the parties involved, without third-party mediation[2].

1.3 Blockchain

The blockchain is a decentralized database that can be shared by a large number of users, allowing information to be stored in an immutable and orderly manner. Information can only be added to the blockchain if there is agreement between the majority of nodes[3].

According to [4] describes the implementation of a blockchain offers the following benefits:

- Enhanced Trust: Blockchain transaction records are disseminated exclusively to authorized entities, thereby precluding unauthorized access and ensuring the integrity of the data.
- Increased Security: The consensus mechanism of all nodes within the blockchain network ensures the addition of blocks, making the transaction data immutable and impervious to subsequent alterations.
- Improved Efficiency: The intrinsic veracity of all transaction data obviates the need for verification, and the utilization of smart contracts accelerates transactional procedures.

There exist various blockchain networks that can be deployed depending on the specific project requirements [5]. For the purposes of this self-managed digital identity project, a publicly-permissioned network was selected comprising the following subsidiary networks:

1. The Authorization network, which employs a series of restrictions and regulatory frameworks to determine the nodes authorized for access and participation in the blockchain.
2. The Consensus network, which functions as an authorization network using consensus protocols to authenticate and log transactional data. In this context, all participating nodes collectively share the duty of voting to incorporate new transactions.

Furthermore, to improve transactional efficiency, a series of predefined rules, known as *smart contracts*, are embedded within the blockchain. Smart contracts delineate the conditions that govern the transfer of information, encompassing variables such as names, dates, amounts, files, and secrets.

1.4 Hyperledger Indy

The project operates within the Hyperledger domain and is supported by the Linux Foundation. It constitutes a blockchain specifically designed to enable the creation and registration of decentralized digital identities. Indy offers a suite of reusable tools, libraries, and components that facilitate the provisioning of digital identities, ensuring interoperability across various administrative domains and applications[6]. Its core features are as follows:

- Blockchain focused especially on digital identity.
- Robust and secure node structure, all are authorized.
- Use of DIDs (decentralized identifiers) that are globally unique and trusted without requiring any centralized authority.
- Uses Zero-Knowledge Proof (ZKP) protocol, which can verify specific data within a dataset without exposing it.

1.5 Blockchain in Self-Managed Digital Identity in Ecuador

Acquiring a Self-Managed Digital Identity is paramount in contemporary society, as technology permeates every aspect of our personal and professional lives. This digital identity is essential to mitigate risks such as information theft, identity theft, and unauthorized disclosure of personal data. On 16 September 2019, Ecuador experienced a significant data breach that affected at least 20.8 million citizens. The leaked information, including full names, dates of birth, home addresses, ID card numbers, employment histories, educational levels, and credit details, was commoditized and sold to companies that sell goods or services. The breach highlighted the vulnerabilities inherent in the databases maintained by public and private institutions [7]. TBlockchain technology is proposed as a

robust solution to protect individual identities from such breaches. At the National University of Loja, a decentralized application (DApp) named "Untity" has been developed to facilitate self-managed academic digital identities for students, faculty, and staff.

It is necessary to know the following fundamental concepts on which it is built to understand how Untity works[8]:

- Decentralized identifiers (DIDs): They are identifiers that grant a verifiable digital identity to a subject (person, company, organization, etc.). The DID is the digital signature of the user in the blockchain network; therefore, it must be used to perform any action. In addition, registering the DID in the digital wallet will grant the user access to the management and full control of their information.
- Transcription Schemas: These are the digital facilitators of entity exchange. A schema, a digital file, contains the name of the schema, the attributes, and the version. It holds the information that the 'requesting' entity deems necessary for the 'recipient' entity. When entities use a schema, they follow a transaction process to define and confirm transcription credentials, similar to executing a contract. The information becomes a valuable asset to be exchanged, making the process practical and efficient.
- Public permissive decentralized network: This is a network that values transparency and fairness. It's a set of nodes that initiates the network and allows the integration of new nodes if they meet the established authenticity and regulation requirements. This type of network is self-sufficient and is characterized by its transparency and transactions without operating costs, providing a fair and cost-effective platform for all participants.

Materials and Methods

Our project began with a comprehensive application of research techniques and meetings. These were instrumental in gathering, verifying, and validating all the necessary information about the functional and non-functional requirements. We employed analytical, deductive, and inductive methods to capture all required information. The IEEE 830 requirements specification document was our key tool in this process. Additionally, we utilized the Agile Block Chain DApp Engineering (ABCDE) [9] development methodology for the integration of a traditional application with a decentralized application, resulting in a DApp [10].

The primary technological materials used are detailed in Table 1: 1:

Results

3.1 Defining the DApp

Using the IEEE 830 standard, functional requirements (see Table 2) and nonfunctional requirements were determined.

Most functional requirements are focused on the digital identity developed on the Hyperledger Indy blockchain, except for RF02 and RF04 that make use of a traditional database to fulfill their purpose. In addition, the DApp architecture was designed (see Fig. 2)), which consists of two subsystems:

- Application Subsystem eThis subsystem focuses on user interaction. The user interacts with the DApp interface (Front-end module), which in turn

Table 1. Main materials used.

Software	
Detalle	Description
Draw.io	Free online software that allows the construction of diagrams, figures, flows, etc.
Hyperledger Indy	The blockchain-based framework focused on digital identity
Vue	Frontend framework required for building user interfaces
Express	Backend framework required to build custom middleware
Flask	Backend framework required for the integration of Hyperledger Indy functionalities.
GitHub	Online repository for saving and updating Untity code
Docker	Thanks to its containers, software is required to implement the blockchain network and modules.

Table 2. Functional requirements.

ID	Name	Description
RF01	Authenticate	The system will allow the user to authenticate
RF02	Login	The system will allow the user to log in
RF03	Registrar	Register
RF04	Reset password	The system will allow the user to reset his password.
RF05	Manage information	The user will be able to request information from another through the transcription schemes defined in the Blockchain.
RF06	Manage courses	The system will allow users to manage their courses.
RF07	Using Transcription Schemes	The system will allow the user to use transcription schemes
RF08	Search user profile	The system will allow the user to search for information of other users, being exposed if they have such public information profile.
RF09	Information Profile Manager	The system will allow the user to manage his or her profile information.
RF10	Manage Transcription Schemes	The system will allow the user-creator to manage Transcription Schemas
RF11	Manage users	The system will allow the administrator role to manage the other users.

interacts with the security server (Middleware module). These two modules are key to granting access to the blockchain and, therefore, to digital identity.

- Smart Contracts Subsystem: This subsystem focuses on user interaction with the blockchain. Once the user interaction is validated, it reaches the Chain Code server (Back-end module), which is responsible for processing the request. If the request is correct through Indy Service, it can be written or read in the blockchain within the Hyperledger Indy Blockchain server.

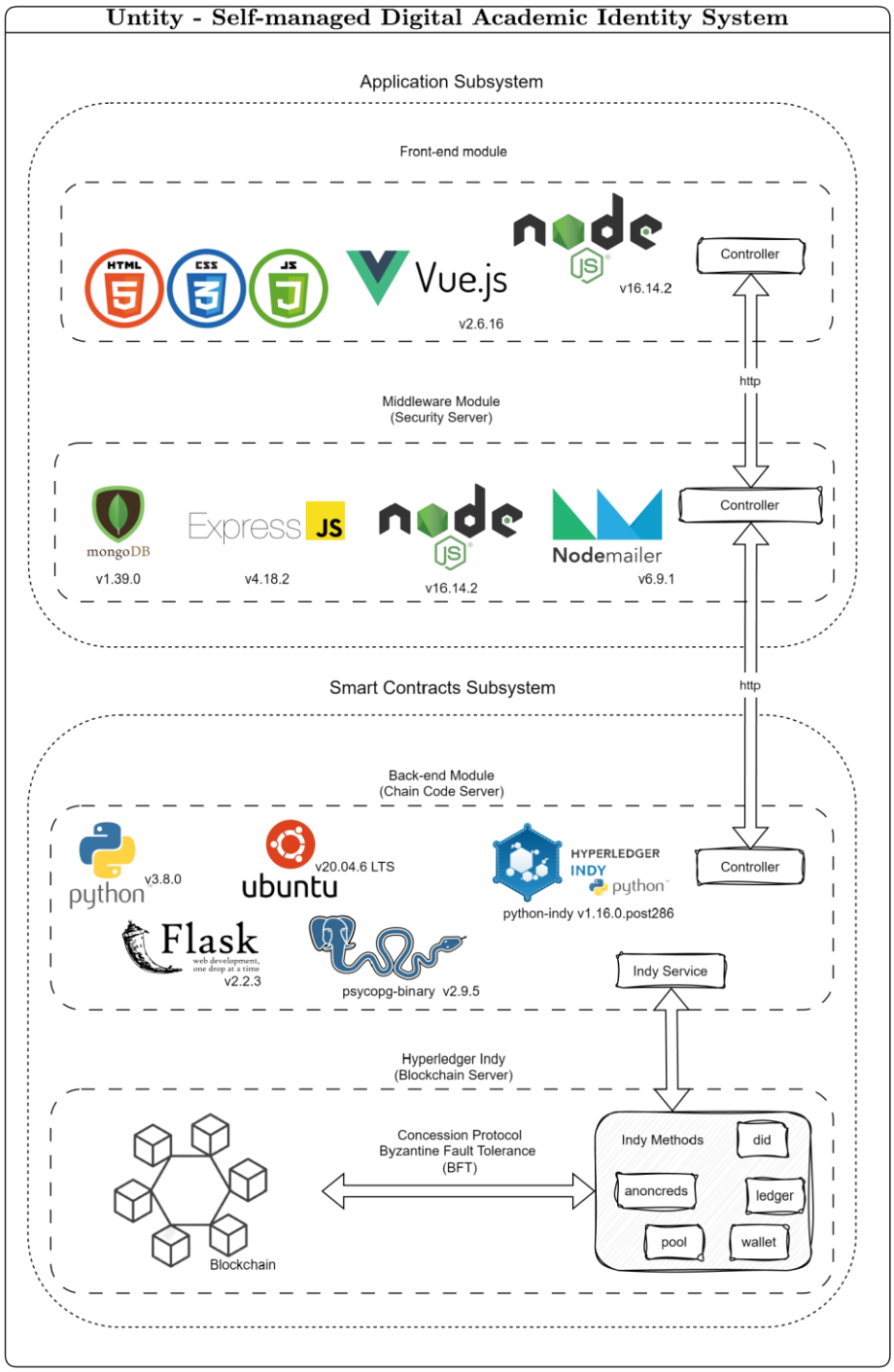


Fig. 2. DApp architecture.

3.2 Building the DApp

The architectural framework is constructed to prioritize the integration of the wallet and the smart contract as foundational components for the decentralized application (DApp).

- Wallet: It is the digital wallet provided to the user, with which they can transact in the Hyperledger Indy blockchain network, but above all, it is where the digital identity will be stored 3 shows how the wallet is created in a simple way. The wallet is created/verified, and if it does not exist, it is created; otherwise, its identifier is obtained. When a new wallet is made, it must be registered under the government domain to transact in the blockchain network.

– Smart Contract or Chaincode: These are rules established for transacting over the blockchain network. Hyperledger Indy calls them "Transcription Schemas, " which users can transact with their information 4 shows how to create the Transcript Schema simply. First, the schema is configured with the name, version, and attributes (information to be transferred); second, the schema is created, and finally, it is sent to the ledger (blockchain network) so that users can use it.

*Note: Government refers to the entire blockchain; if any wallet is not registered under the blockchain's domain, it will not be able to use any functionality; literally, neither the wallet nor the user will exist.

3.3 DApp Testing

To test the Self-Managed Digital Identity DApp, 3 stages were conducted:

Stage 01. Integration Testing: This process enables the comprehensive validation of the system modules' collective functionality, thereby confirming the successful integration of the smart contract and application subsystems (see Table 3).

Table 3. Integration Test Results.

ID	Description	State
CP01	User is correctly registered in the system	Successful
CP02	Successfully logged on to the system	successfully
CP03	The user manages his personal information	success
CP04	The user manages his courses from the Computer Science Career page.	Success.
CP05	The user manages his transactions	Successful
CP06	The user uses the transcription schemes	successful
CP07	The user views his transaction history	Successful
CP08	The user successfully completes the transaction process	Successful

```
async def create_wallet(identity):
    logger.info("\"{}\" -> Create wallet".format(identity['name']))
    try:
        await wallet.create_wallet(identity['wallet_config'],
                                   identity['wallet_credentials'])
    except IndyError as ex:
        if ex.error_code == ErrorCode.PoolLedgerConfigAlreadyExistsError:
            pass
    identity['wallet'] = await wallet.open_wallet(identity['wallet_config'],
                                                  identity['wallet_credentials'])

async def getting_verinym(from_, to):
    from_['info'] = {
        'did': to['did'],
        'verkey': to['key'],
        'role': to['role'] or None
    }

    await send_nym(from_['pool'], from_['wallet'], from_['did'], from_['info']['did'],
                   from_['info']['verkey'], from_['info']['role'])

async def send_nym(pool_handle, wallet_handle, _did, new_did, new_key, role):
    nym_request = await ledger.build_nym_request(_did, new_did, new_key, None, role)
    await ledger.sign_and_submit_request(pool_handle, wallet_handle, _did, nym_request)
```

Fig.3. Wallet configuration.

```
    # Computacion -> Configura el Esquema de Transcripción
    transcript = {
        'name': 'Esquema_Cursos',
        'version': '1.2',
        'attributes': ['cedula', 'nombre del curso', "tiempo de validez"]
    }

    # Computacion -> Crea Esquema de Transcripción
    (computacion['transcript_schema_id'], computacion['transcript_schema']) = \
        await anoncreds.issuer_create_schema(computacion['did'], transcript['name'],
transcript['version'],
                                             json.dumps(transcript['attributes']))
    transcript_schema_id = computacion['transcript_schema_id']

    # Computacion -> Envía el Esquema de Transcripción hacia el Ledger (Blockchain)
    await send_schema(computacion['pool'], computacion['wallet'], computacion['did'],
computacion['transcript_schema'])
```

Fig.4. Transcription Scheme Configuration.

Stage 02. Functional Testing: The DApp was found to meet and satisfy all specified functional requirements (see Table 4).

Table 4. Summary of the test cases of the Functional Test Plan.

ID	Description	State
CP01	The system response shall be checked when the user is about to log in. [RF01 - RF02]	Successful
CP02	The response of the system will be checked when the user is going to register. [RF01 - RF03]	Successful
CP03	The response of the system will be tested when the user is about to reset his password. [RF01 - RF03 - RF04]	Successful
CP04	The responsiveness of the system will be tested when the user is going to manage his personal information. [RF01 - RF02 - RF05 -RF09]	Successful
CP05	The system response will be checked when the user goes to administer his/her courses. [RF01 - RF02 - RF06 - RF09]	Successful
CP06	The system's response will be checked when the user goes to manage his transactions. [RF01 - RF02 - RF09]	Successful
CP07	The response of the system shall be tested when the user is going to use a scheme. [RF01 - RF02 - RF07]	Successful
CP08	The response of the system will be tested when the user goes to search for another user's profile. [RF01 - RF02 - RF08 - RF09]	Successful
CP09	The response of the system will be tested when the user goes to manage his/her profile. [RF01 - RF02 - RF05 - RF06 - RF09]	Successful
CP10	The response of the system shall be tested when the user is going to administer the transcription schemes. [RF01 - RF02 - RF10]	Successful
CP11	System response will be tested when the user is going to administer the users [RF01 - RF02 - RF11]	Successful

Stage 03. Acceptance Testing: allowed to determine the level of acceptance of the DApp, therefore, a survey was applied to a sample of 50 students of the National University of Loja belonging to the Computer Engineering Career. The sample of students carried out a complete interaction with the DApp and at the end they completed a satisfaction survey. The results were as follows: yes (77.74%), partially (20.61%) and no (1.45%), determining that the level of acceptance of the self-managed digital identity DApp is positive.

Discussion

When reviewing related works on blockchain, it was observed that none of the applied methodologies follows an official methodology that integrates a blockchain with an application. For this reason, the ABCDE software development methodology is the only one for this type of project, which integrates blockchain smart contracts (chaincode) with a traditional application, resulting in a DApp. The development of the smart contract subsystem relied on Hyperledger Indy's official documentation to implement various blockchain methods. This enabled the configuration and creation of nodes, the Pool, Wallets, DIDs, Transcription Schemes, and Transaction Process. Unlike other frameworks, Indy allows control over wallet and DID creation, preventing credential duplication errors. The transaction process is divided into 4 phases for better user control, with methods in the wallet to store personal information. Unlike other projects focused solely on transactions, this project emphasizes securing user information. Since Hyperledger Indy does not store DIDs, Transcription Schemas, and Transaction Processes, a PostgreSQL database is used for managing this data. Transactions through transcription schemes are free on the Hyperledger Indy blockchain, a decentralized, permissioned network. Users can exchange information without economic costs, provided they follow system rules; noncompliance results in account deactivation.

Conclusions

The implementation of Blockchain technology allows securing digital identity due to its immutability, transparency, reliability, and security features through Hyperledger Indy methods (Wallet, DID, Ledger, among others) and also gives users full control of their digital identity.

Using the ABCDE agile methodology ensures creating a robust, reliable, and secure DApp through phased design, coding, and testing of subsystems, validating performance before integration.

Transcription schemes in Hyperledger Indy act as smart contracts, enabling users to exchange information based on their specifications..

Hyperledger Indy's blockchain is devoid of transfer fees, thereby streamlining the exchange of information among its users. The process is contingent solely upon the users' intention at the commencement and conclusion of a transaction.

Developing a Decentralized Application (DApp) centered on autonomous digital identity empowers users to exert comprehensive control over their personal information, mitigating concerns regarding unauthorized manipulation or exposure, a prevalent issue in conventional applications.

Acknowledgments We would like to thank the Universidad Nacional de Loja for having been the promoter of this type of research project, as well as the research team that contributed to the development of the project.

References

1. FIDE, https://www.fide.edu.pe/es-ec/blog/detalle/que-es-la-identidad-digital-tuhuella-en-el-mundo-online/. Last accessed on 21 May 2024
2. Infominer, https://decentralized-id.com/organizations/sovrin-foundation/. Last accessed on 21 May 2024
3. Wilfredo Z. U.: Blockchain y la innovación en las tecnologías. Centro de investigación & producción científica IDEOs 2 (2022)
4. WeLiveSecurity, https://www.welivesecurity.com/la-es/2022/05/13/blockchainque-es-como-funciona-y-como-se-esta-usando-en-el-mercado/. Last accessed 21 May 2024
5. SAP, https://www.sap.com/latinamerica/products/artificial-intelligence/what-isblockchain.html. Last accessed on 21 May 2024
6. Hasib A., https://101blockchains.com/hyperledger-indy/. Last accessed 21 May 2024

7. DPL News, https://dplnews.com/tras-masiva-filtracion-advierten-que-venta-dedatos-personales-no-es-nueva-en-ecuador/. Last accessed 21 May 2024
8. Marcos A. L.: Identidad digital auto-gestionada: El futuro de la identidad digital: Auto-gestión, billeteras digitales y blockchain. Inter-American Development Bank 2 (2020)
9. Marchesi, L., Marchesi, & M., Tonelli, R.: ABCDE—agile block chain DApp engineering. Blockchain: Research and Applications. (2020)
10. Cáceres Salamea, M. C., & Peralta Velecela, D. F.: Introducción a Blockchain, Contratos Inteligentes y Aplicaciones Descentralizadas. Universidad de Vigo. (2023)
11. José, M., Iglesias, & F..: Identidad digital basada en blockchain en instituciones educativas. Universidad de Los Andes. (2020)
12. Sanz González, G.: Diseño e implementación de un sistema de identidad digital descentralizada para ciudadanos de la Unión Europea en el ámbito sanitario. Universidad Politécnica de Madrid. (2023)
13. Cáceres Salamea, M. C., & Peralta Velecela, D. F.: Propuesta de identidad digital para historial clínico unificado utilizando tecnología blockchain. Universidad de Cuenca. (2021)
14. Madhuri A, Sawant, B. R., & Deshmukh, A.: Single Page Application using AngularJS. IJCSIT. (2023)
15. Jajodia, S., Samarati, P., Lopez, J., & Vaidya, J.: Blockchains. Springer. (2024)
16. Paik, H. Y., Liu, Y., Lu, Q., & Kanhere, S. S.: Decentralized Identity Management and Blockchains: Design Patterns and Architectures. ResearchGate. (2024)
17. Indy, https://www.hyperledger.org/projects/hyperledger-indy. Last accessed 25 Jun 2024
18. Zwitter, A. J., Gstrein, O. J., & Yap, E.: Digital Identity and the Blockchain: Universal Identity Management and the Concept of the "Self-Sovereign" Individual. Frontiers in Blockchain. (2020)
19. Sherriff, A., Young, K., & Shea, M.: Establishing Self Sovereign Identity with Blockchain. Frontiers in Blockchain. (2022)
20. Ishmaev, G.: Sovereignty, privacy, and ethics in blockchain-based identity management systems Springer. (2021)
21. Schardong, F., & Custódio, R.: Self-Sovereign Identity: A Systematic Review, Mapping and Taxonomy. Ethics and Information Technology. MDPI. (2021)
22. Gans, R. B., Ubacht, J., & Janssen, M.: Governance and societal impact of blockchain-based self-sovereign identities. Oxford Academic. (2022)

Anexo 11: Certificado de traducción

Loja, 15 de junio de 2024

Lic. Karina Yajaira Martínez Luzuriaga

LICENCIADA EN CIENCIAS DE LA EDUCACIÓN MENCIÓN INGLÉS

CERTIFICO:

Yo, Karina Yajaira Martínez Luzuriaga con cédula de identidad Nro. 1104902679, **Licenciada en Ciencias de la Educación Mención Inglés** por la Universidad Técnica Particular de Loja, con número de registro 1031-2022-2574017 en la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación, señalo que el presente documento es fiel traducción del idioma español al idioma inglés del resumen del Trabajo de Integración Curricular denominado **"Propuesta de identidad digital académica auto-gestionada mediante tecnología Blockchain para la Universidad Nacional de Loja."** elaborado por el Sr. Carlos Alexis Armijos Rios, con cédula de identidad Nro. 1900549179, estudiante egresado de la carrera Ingeniería en Computación de la Universidad Nacional de Loja.

KARINA YAJAIRA MARTINEZ LUZURIAGA

Lic. Karina Yajaira Martínez Luzuriaga

C.I. 1104902679

REGISTRO SENESCYT N°: 1031-2022-2574017

URL: *https://drive.google.com/file/d/1EpGoNJxTH7Jw9Uqkg0ZhxypR9S3Z0lZ5/view?usp=drive_link*