PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito final para obtener el Título de INGENIERO DE SISTEMAS

Desarrollo de un Sistema de Gestión de Identidad de Pacientes en Instituciones de Salud mediante Tecnología de Cadena de Bloques

Por

Br. JOSMAR ALEJANDRO MUÑOZ PÉREZ
Tutor: Dr. Jonás Montilva

Agosto 2024

Resumen

Este proyecto desarrolló un sistema de gestión de identidad de pacientes basado en tecnología de cadena de bloques, con el objetivo de mejorar la calidad de la atención médica y fomentar la interoperabilidad entre instituciones de salud. Se empleó la metodología ágil SCRUM, utilizando *sprints* (iteraciones del proyecto) de dos semanas y roles definidos, para gestionar el proyecto. La metodología Blue Watch guió el desarrollo iterativo del prototipo, que incluyó funcionalidades como registro de usuarios, gestión de consentimiento y generación de credenciales digitales. Se realizaron pruebas unitarias, de integración y de sistema para garantizar la calidad, seguridad y eficiencia del sistema. Los resultados demostraron la viabilidad técnica del prototipo y su capacidad para mejorar la privacidad y accesibilidad de los datos de los pacientes. El sistema desarrollado contribuye a una atención médica más segura y eficiente, al tiempo que empodera a los pacientes al otorgarles el control sobre sus datos médicos.

Palabras clave: Gestión de identidad, Cadena de Bloques, Interoperabilidad, Instituciones de Salud.

Índice

Índice de Figuras

Índice de Tablas

Capítulo 1
Introducción

En la era digital en la que vivimos, la atención médica se encuentra en constante evolución, adoptando innovaciones tecnológicas para mejorar la calidad de los servicios y la gestión de información crítica. Uno de los aspectos fundamentales en este contexto es la identificación precisa y segura de pacientes en instituciones de salud. En este sentido, la tecnología *blockchain* (cadena de bloques) ha emergido como una solución prometedora para abordar los desafíos relacionados con la gestión de identidades en el ámbito médico.

Sin embargo, a pesar de los avances tecnológicos en otros campos de la atención médica, la identificación de pacientes sigue siendo un problema persistente. Errores de identificación, problemas de privacidad y falta de interoperabilidad entre instituciones de salud son cuestiones que afectan negativamente tanto a pacientes como a profesionales de la salud. En este contexto, surge la pregunta central de nuestro estudio: ¿Cómo puede la tecnología *blockchain* mejorar la gestión de identidades de pacientes en instituciones de salud y resolver los problemas existentes?

Para responder a esta pregunta, este trabajo se enfoca en el desarrollo de un sistema de gestión de identidad de pacientes basado en tecnología de cadena de bloques. El objetivo general de este proyecto es diseñar, desarrollar y evaluar un sistema que aborde eficazmente los desafíos actuales en la identificación de pacientes, mejorando la calidad de la atención médica, garantizando la privacidad de los pacientes y fomentando la interoperabilidad de las instituciones de salud. Para alcanzar este objetivo, se han definido objetivos específicos que guiarán el desarrollo de la investigación y la implementación del sistema.

1.1. Planteamiento del Problema

En el entorno de las instituciones de salud, la gestión precisa y segura de la identificación de pacientes es fundamental para garantizar la atención médica adecuada y la integridad de los registros clínicos. A pesar de los notables avances tecnológicos en diversas áreas de la atención médica, la identificación precisa de pacientes continúa siendo un problema persistente y vulnerable a errores. Este desafío se ve agravado por la carencia de un sistema confiable y eficaz para gestionar las identidades en el ámbito de la atención médica.

Actualmente, la mayoría de las instituciones de salud dependen de sistemas de identificación basados en registros de papel o bases de datos centralizadas, que pueden ser vulnerables a errores humanos, accidentes laborales, fraudes y ataques cibernéticos. Estos métodos tradicionales también pueden resultar en la duplicación de registros, la pérdida de información crucial y la falta de interoperabilidad entre diferentes instituciones de salud.

Además, la creciente preocupación por la privacidad y la protección de datos en el ámbito de la atención médica hace que sea crucial encontrar soluciones que aborden estos problemas de manera eficaz. La tecnología *blockchain*, con su capacidad para proporcionar una plataforma segura, descentralizada y transparente para la gestión de identidades, se presenta como una solución prometedora para abordar los desafíos actuales en la identificación de pacientes en instituciones de salud.

Sin embargo, a pesar de su potencial, la implementación efectiva de sistemas de identificación de pacientes basados en *blockchain* en entornos de atención médica aún enfrenta numerosos obstáculos, que incluyen la falta de normativas y estándares claros, la resistencia al cambio por parte de los profesionales de la salud y la necesidad de garantizar la escalabilidad y la interoperabilidad con sistemas existentes.

Por lo tanto, el presente proyecto de grado se enfocará en abordar este problema mediante el desarrollo de un sistema de gestión de identidad de pacientes basado en tecnología de cadena de bloques, que sea eficiente, seguro y capaz de superar los desafíos actuales en la identificación de pacientes en instituciones de salud. La investigación y desarrollo de este sistema contribuirá a mejorar la calidad de la atención médica, garantizar la privacidad de los pacientes y facilitar la interoperabilidad entre instituciones de salud en beneficio de pacientes y profesionales médicos.

1.2. Objetivos

Objetivo General

Desarrollar un sistema de gestión de identidad de pacientes en instituciones de salud que implemente mecanismos de descentralización, privacidad y seguridad que mejore la calidad de la atención médica y fomente la interoperabilidad entre instituciones de salud en beneficio de pacientes y profesionales médicos utilizando tecnología *blockchain*.

Objetivos Específicos

- Investigar y analizar los desafíos existentes en la identificación de pacientes en instituciones de salud para comprender a fondo las deficiencias y problemáticas actuales en la gestión de identidades de pacientes a través de revisión bibliográfica y entrevistas a los usuarios finales del sistema.
- Analizar la tecnología *blockchain*, sus características y aplicaciones con el propósito de determinar su idoneidad en la gestión de identidades de pacientes en el ámbito de la salud, mediante un estudio detallado de la tecnología y sus casos de uso.
- Diseñar un sistema de gestión de identidad de pacientes basado en tecnología *blockchain* con el objetivo de abordar los desafíos identificados y mejorar la calidad de la atención médica, mediante la creación de un diseño conceptual y arquitectónico.
- Desarrollar un prototipo funcional del sistema de gestión de identidad de pacientes para demostrar la viabilidad técnica del enfoque propuesto mediante la implementación de la tecnología *blockchain* y considerando aspectos de seguridad y privacidad de datos.
- Evaluar la eficiencia y la seguridad del sistema de gestión de identidad de pacientes para asegurar su capacidad de prevenir errores de identificación y proteger la información confidencial a través de pruebas exhaustivas y análisis de vulnerabilidades.

- Validar la interoperabilidad del sistema propuesto para facilitar el intercambio seguro de datos médicos entre instituciones de salud mediante pruebas de integración y análisis de compatibilidad.

1.3. Ámbito de aplicación de la solución

El ámbito de aplicación de este proyecto está compuesto específicamente por las instituciones de salud tanto públicas como privadas, ya que todas se pueden beneficiar de un sistema descentralizado de identificación de pacientes que promueva la interoperabilidad entre ellas y mejore la satisfacción que proporciona el servicio a sus usuarios. Este ámbito también se ve limitado por las posibles regulaciones gubernamentales en torno al uso de la tecnología *blockchain* para el manejo de datos de salud, por lo que el producto final del proyecto sólo puede ser utilizado donde dichas regulaciones lo permitan.

1.4. Justificación

La implementación de un sistema de gestión de identidad de pacientes basado en tecnología *blockchain* se justifica por la necesidad apremiante de abordar desafíos críticos en el ámbito de la atención médica. A pesar de los avances tecnológicos en diversas áreas de la salud, la identificación precisa de pacientes sigue siendo un problema persistente y propenso a errores. Los errores de identificación pueden tener consecuencias graves, desde tratamientos incorrectos hasta la falta de acceso a historiales médicos vitales. Por lo tanto, la búsqueda de soluciones innovadoras que mejoren la precisión en la identificación de pacientes es imperativa.

Además, la gestión segura de los datos de salud y la privacidad de los pacientes son preocupaciones fundamentales en el campo de la atención médica. En un mundo donde los datos son valiosos y se utilizan para tomar decisiones críticas en el tratamiento de enfermedades, es esencial proteger la información confidencial de los pacientes. La tecnología *blockchain*, con su capacidad para proporcionar una plataforma segura y transparente, tiene el potencial de transformar la forma en que se protegen y comparten los datos médicos, otorgando a los pacientes un mayor control sobre su información.

La interoperabilidad entre sistemas de salud también es un desafío importante. Las instituciones de salud a menudo utilizan sistemas de información heterogéneos que dificultan el intercambio de datos entre diferentes entidades médicas. La tecnología *blockchain* puede facilitar la interoperabilidad al ofrecer un marco común y seguro para la gestión de identidades, permitiendo una colaboración más efectiva y una atención médica más coordinada.

Además de abordar estos desafíos, la implementación de un sistema de gestión de identidad de pacientes basado en *blockchain* representa una oportunidad para avanzar en la innovación en el campo de la salud. La adopción de tecnologías emergentes puede impulsar el desarrollo de soluciones más efectivas y eficientes en el futuro, mejorando la calidad de la atención médica en su conjunto.

Finalmente, la adopción de tecnología *blockchain* también puede ayudar a las instituciones de salud a cumplir con las regulaciones y normativas estrictas que rigen la gestión de datos de salud y la identificación de pacientes, lo que proporciona un marco sólido para el cumplimiento normativo y la seguridad de los datos.

En resumen, la realización de este proyecto se justifica por su capacidad para abordar problemas significativos en la atención médica, mejorar la calidad de la atención, proteger la privacidad de los pacientes, promover la interoperabilidad, fomentar la innovación y cumplir con las regulaciones, brindando así beneficios sustanciales a pacientes y profesionales de la salud.

1.5. Antecedentes

La creciente necesidad de innovación en torno a la tecnología *blockchain* y su relevancia en diferentes sectores ha dado lugar a una continua exploración en diversos ámbitos científicos y prácticos, incluyendo el campo de la salud. Aunque aún se encuentra en sus fases iniciales de desarrollo, Sharma et al. (2020) sugiere que la tecnología *blockchain* se perfila como una solución progresiva para abordar las preocupaciones tecnológicas actuales, tales como la descentralización, la gestión de identidades, la confianza, la propiedad de los datos y la toma de decisiones basada en información verificable. Al mismo tiempo, el mundo se encuentra ante un crecimiento exponencial en la diversidad y cantidad de datos digitales generados por máquinas y usuarios. En medio

de la búsqueda de enfoques efectivos para almacenar y procesar datos en entornos en la nube, la innovación que aporta la tecnología *blockchain* se erige como un componente valioso en la consecución de este objetivo.

El almacenamiento seguro y el pleno uso de los registros médicos personales siempre ha sido una preocupación para la población en general, ya que los datos médicos precisos y completos son un activo valioso para los pacientes, y la protección de la privacidad y el almacenamiento seguro de datos médicos son cuestiones cruciales en el ámbito de los servicios médicos. Siendo consciente de las características de descentralización, verificabilidad e inmutabilidad de la tecnología *blockchain*, Chen et al. (2019) propone que esta se puede utilizar para almacenar de forma segura datos médicos personales y diseña un esquema de almacenamiento para gestionar datos médicos personales basado en *blockchain* y almacenamiento en la nube. Además, describe un marco de servicio para compartir registros médicos y presenta y analiza las características de una *blockchain* médica mediante una comparación con los sistemas tradicionales. El esquema de almacenamiento e intercambio que propone no depende de ningún tercero y ninguna parte tiene poder absoluto para afectar el procesamiento.

Por otra parte, como sugiere Vazirani et al. (2020), los esfuerzos por reformar los sistemas de registros médicos en el Reino Unido en años recientes han dejado una situación compleja y aún no completamente digitalizada. A pesar de avances significativos en la aplicación de tecnología en la práctica clínica y la inversión económica substancial en el Servicio Nacional de Salud, los aspectos administrativos de la atención sanitaria siguen padeciendo una marcada falta de interoperabilidad.

Esta carencia de interoperabilidad no solo puede dar lugar a errores médicos, sino también a problemas administrativos, como se evidenció en el reciente incidente en el que el Servicio Nacional de Salud británico falló al no invitar a casi 50.000 mujeres a una prueba de detección de cáncer cervical. Además, los pacientes se ven obligados a relatar sus historias médicas en repetidas ocasiones, un proceso ineficiente y tedioso que puede dar lugar a confusiones y errores clínicos debido a la falta de información completa.

Lo que es más crítico, la ausencia de una estructura unificada puede dar lugar a situaciones evitables en las que no se dispone de información a tiempo, lo que puede tener graves consecuencias clínicas. Cada vez más, la tecnología *blockchain* se está empleando

para abordar el problema de la interoperabilidad al brindar a los médicos datos más completos sobre los pacientes, obtenidos de sistemas interconectados pero gestionados de forma independiente. Esto representa un paso importante en la resolución de estos desafíos en la atención médica.

Capítulo 2
Marco Teórico

Con la finalidad de obtener una mayor comprensión de este proyecto de grado, a continuación, se presenta la fundamentación teórica conformada por diferentes conceptos y sus correspondientes definiciones sobre el campo de la salud, datos, sistemas y tecnología, adoptando un enfoque interdisciplinario.

2.1. Fundamentación Teórica

Los sistemas de salud son estructuras complejas y dinámicas cuyo objetivo principal es satisfacer las necesidades de salud de la población. Hay cuatro funciones básicas que cumplen los sistemas de salud según la Organización Mundial de la Salud. Estas funciones son: Gobernanza, finanzas, generación de recursos y prestación de servicios de salud. Esencialmente, estas funciones nos permiten alcanzar tres objetivos fundamentales: mantener o mejorar las condiciones de salud de la población, brindar una buena capacidad de respuesta (es decir, brindar un tratamiento adecuado a la población que lo necesita) y brindar un nivel de seguridad financiera a la población que requiere servicios de salud.

Por otra parte, las instituciones de salud, según Díaz (2015), son organizaciones que ofrecen servicios de salud a la población. Al igual que las empresas, utilizan recursos materiales y humanos para satisfacer las necesidades de sus clientes. Su objetivo puede ser la obtención de ganancias o la prestación de servicios de calidad.

De la misma forma, la atención médica se refiere al conjunto de servicios y acciones proporcionados por profesionales de la salud para diagnosticar, tratar y prevenir enfermedades y lesiones en los pacientes. Esta definición implica tanto la atención de

enfermedades físicas como mentales, y puede incluir consultas médicas, exámenes médicos, pruebas de laboratorio, procedimientos médicos, cirugías, terapias y seguimiento médico. La atención médica tiene como objetivo principal mejorar la salud y el bienestar de los pacientes y brindarles el cuidado necesario para su recuperación.

La fundamentación teórica de este proyecto consta de tres partes fundamentales, las cuales se listan y definen a continuación:

2.1.1. Gestión de identidad de pacientes

2.1.1.1. Identidad de pacientes

Como expresa Allamde (2019), la identificación de pacientes engloba los datos empleados en la identificación de un paciente en un contexto médico, como su nombre, fecha de nacimiento, número de identificación y otros datos personales relevantes para la institución de salud. Esta identificación resulta esencial para asegurar que los pacientes reciban la atención médica adecuada y mantener registros precisos y actualizados.

2.1.1.2. Gestión de identidad

La gestión de identidad de acuerdo con Arellano (2008), se refiere a los procesos y tecnologías utilizados para gestionar y controlar las identidades digitales de individuos o entidades dentro de una organización o sistema. Implica la administración de identidades de usuarios, autenticación, autorización y control de acceso para garantizar que las personas adecuadas tengan acceso adecuado a los recursos y la información.

2.1.1.3. Sistema de identificación

De acuerdo con Evans (2016), un sistema de identificación es un método o proceso utilizado para reconocer y distinguir de forma única individuos, entidades u objetos. Por lo general, implica asignar un código, número u otro identificador específico a cada entidad, lo

que permite una identificación fácil y precisa dentro de un sistema o base de datos. Los sistemas de identificación se utilizan comúnmente en diversos campos, incluida la atención médica, para rastrear y administrar registros de pacientes, dispositivos médicos y otras entidades relacionadas con la atención médica.

2.1.1.4. Interoperabilidad de sistemas de salud

La interoperabilidad de los sistemas de salud, como lo describe Gaynor (2014), se refiere a la capacidad de diferentes aplicaciones, sistemas y dispositivos de atención médica para intercambiar y utilizar datos de manera coordinada y eficiente. Implica la integración de varios estándares, tecnologías y protocolos para garantizar que la información de salud relevante pueda compartirse y comprenderse en diferentes plataformas y organizaciones dentro de la industria de la salud. Lograr la interoperabilidad de los sistemas de salud es crucial para mejorar la atención al paciente, mejorar la toma de decisiones clínicas y optimizar los procesos de atención médica.

2.1.1.5. Escalabilidad de sistemas de salud

La escalabilidad de los sistemas de salud se define como la capacidad de estos sistemas para adaptarse y crecer de manera eficiente en respuesta al aumento de la demanda y el volumen de información. Esto implica que el sistema pueda manejar sin problemas un mayor número de usuarios, transacciones y datos, sin que esto afecte negativamente su rendimiento y funcionamiento.

2.1.1.6. Seguridad de datos

La seguridad de datos se refiere a las medidas y prácticas implementadas para proteger la información y los datos almacenados en sistemas de información. En términos generales, se refiere a la protección de la confidencialidad, integridad y disponibilidad de los datos. La confidencialidad garantiza que los datos sólo sean accesibles para las personas autorizadas, la integridad asegura que los datos no sean alterados de manera no autorizada y

la disponibilidad permite el acceso y uso oportuno de los datos cuando sea necesario. Para lograr la seguridad de datos, se aplican diferentes técnicas y controles, como encriptación, autenticación, acceso basado en roles, firewalls, antivirus, entre otros.

2.1.1.7. Normativas en el manejo de datos médicos

Como lo menciona Contreras (2020), las normativas en el manejo de datos médicos son el conjunto de leyes, regulaciones y lineamientos que rigen la recolección, almacenamiento, uso y protección de datos de salud. Estas normativas establecen los derechos y obligaciones de los proveedores de atención médica, los pacientes y otras partes interesadas involucradas en la gestión de datos de salud. Su objetivo es garantizar la privacidad, confidencialidad y seguridad de la información de salud personal y, al mismo tiempo, facilitar el uso eficaz y eficiente de los datos con fines sanitarios.

2.1.1.8. Estándares en el manejo de datos médicos

Los estándares sobre gestión de datos de salud se refieren a las pautas y especificaciones establecidas que definen la estructura, formato, interoperabilidad y requisitos de seguridad para la recopilación, almacenamiento, intercambio y uso de datos de salud. Estos estándares garantizan coherencia, compatibilidad y calidad en las prácticas de gestión de datos sanitarios en diferentes sistemas y organizaciones. Facilitan el intercambio e integración fluidos de información sanitaria, promueven la interoperabilidad de los datos y respaldan la utilización eficaz de los datos sanitarios para la prestación de atención sanitaria, la investigación y la salud pública.

2.1.1.9. Errores de identificación

Los errores de identificación se refieren a errores o inexactitudes en el proceso de identificación de personas o entidades. Estos errores pueden ocurrir durante los procesos de ingreso de datos, mantenimiento de registros o verificación, lo que genera información de identificación incorrecta o no coincidente. Los errores de identificación pueden tener

consecuencias importantes, como diagnósticos erróneos, tratamientos incorrectos o comprometer la seguridad del paciente.

2.1.1.10. Privacidad del paciente

La privacidad del paciente se refiere al derecho de las personas a controlar la recopilación, el uso y la divulgación de su información personal y de salud. Abarca la protección de información sensible y confidencial relacionada con el historial médico, el diagnóstico, el tratamiento y otros datos relacionados con la atención médica de un paciente. La privacidad del paciente garantiza que las personas tengan la autonomía para tomar decisiones sobre quién puede acceder a su información de salud y cómo se utiliza, al mismo tiempo que protege sus datos personales y confidenciales contra el acceso o la divulgación no autorizados.

2.1.2. La cadena de bloques como solución

2.1.2.1. Cadena de bloques

Según Swan (2015), la cadena de bloques es una estructura de datos que se utiliza para almacenar información de manera segura y descentralizada. Es conocida en inglés como *blockchain*. La información se agrupa en conjuntos llamados bloques, a los que se les añade metainformaciones relativas a otro bloque de la cadena anterior en una línea temporal para hacer un seguimiento seguro a través de grandes cálculos criptográficos. De esta forma, gracias a técnicas criptográficas, la información contenida en un bloque solo puede ser repudiada o editada modificando todos los bloques anteriores. Esta propiedad permite su aplicación en un entorno distribuido de manera que la estructura de datos *blockchain* puede ejercer de base de datos pública no relacional que contenga un histórico irrefutable de información.

Tomando como referencia la especificación inicial que se planteó para la cadena de bloques de bitcoin por Nakamoto (2008), el funcionamiento de esta se puede describir de la siguiente forma:

Los activos digitales transferibles y almacenables en la cadena de bloques se definen como una cadena de firmas digitales. Cada dueño de los activos los puede transferir al siguiente firmando un hash de la transacción previa y la llave pública del siguiente dueño y agregándolas al final del activo, pudiendo el receptor verificar las firmas para comprobar la cadena de propiedad. Se podría presentar el problema de saber si el dueño anterior no ha gastado el activo dos o más veces, el cual generalmente se resuelve teniendo una autoridad central que verifique la validez de la transacción, pero en nuestro caso este problema se resuelve anunciando todas las transacciones públicamente, donde todos los participantes concuerden con un único historial del orden de transacciones recibidas, donde la primera realizada será la válida.

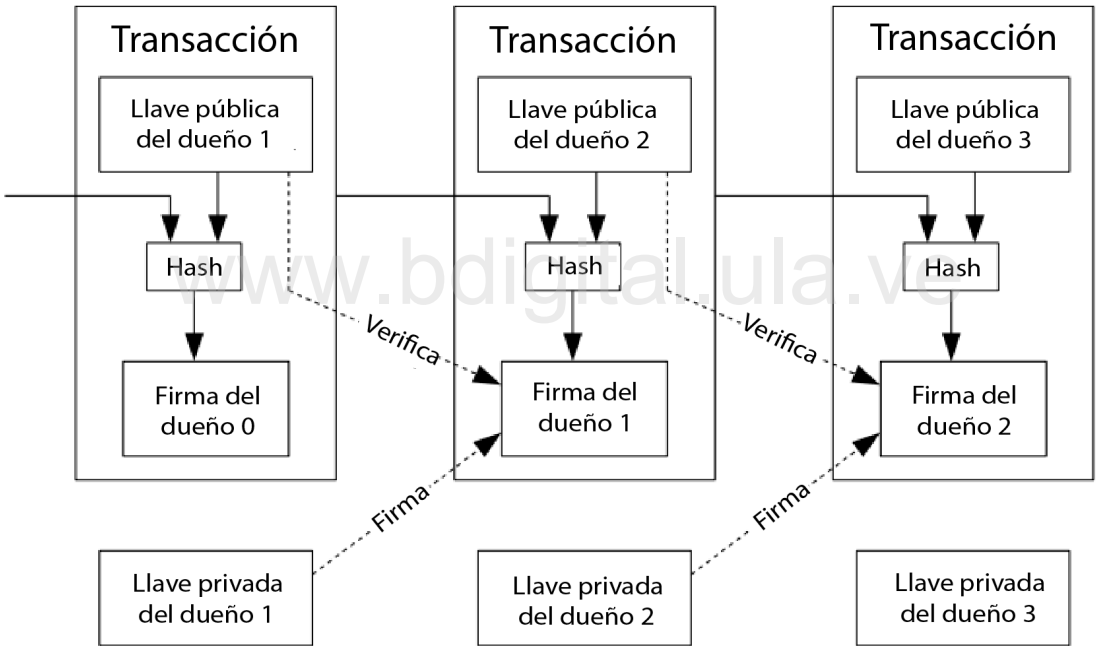


Figura 2.1: Transacciones en cadena de bloques

Para el funcionamiento de esta solución es necesario un servidor de marcas de tiempo, el cual toma el hash de un bloque para ser marcado y ampliamente publicado a todos los participantes. La marca de tiempo prueba que los datos debieron haber existido en ese momento para poder haber estado asociados al hash del bloque. Cada marca de tiempo incluye la marca de tiempo anterior en su hash, formando una cadena con cada marca de tiempo adicional, reforzando las anteriores.

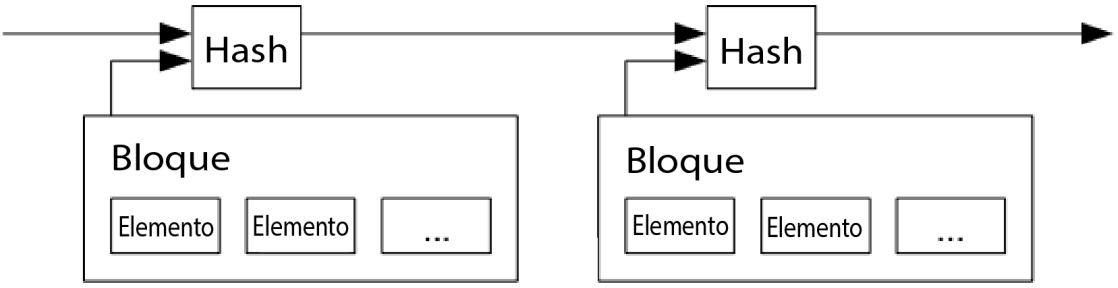


Figura 2.2: Servidor de marcas de tiempo

La implementación de un servidor de marcas de tiempo distribuido en un contexto de participantes iguales requiere la utilización de una prueba de trabajo que los haga merecedores de aprobar el siguiente bloque de la cadena. La prueba de trabajo implica la búsqueda de un valor que cuando es hasheado con un algoritmo como el SHA-256, el hash comience con un determinado número de bits cero. El trabajo requerido es exponencial en el número de bits cero que se necesitan y puede ser verificado ejecutando un solo hash.

Para esta red de marcas de tiempo, se implementa la prueba de trabajo incrementando un nonce (Abreviación en inglés de "número usado una vez") en el bloque hasta que se encuentre un valor que le dé al hash del bloque los bits cero requeridos. Una vez que la CPU ha realizado el esfuerzo para que satisfaga la prueba de trabajo, el bloque no se puede cambiar sin rehacer el trabajo. A medida que se encadenan bloques posteriores, el trabajo para cambiar el bloque incluiría rehacer todos los bloques posteriores.

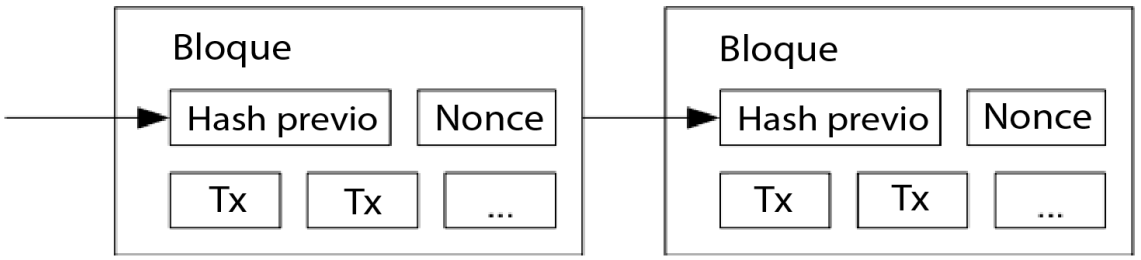


Figura 2.3: Cadena de bloques

En resúmen, los pasos para ejecutar la red son los siguientes:

1. Las nuevas transacciones se transmiten a todos los nodos.

2. Cada nodo recopila nuevas transacciones en un bloque.
3. Cada nodo trabaja para encontrar una prueba de trabajo difícil para su bloque.
4. Cuando un nodo encuentra una prueba de trabajo, transmite el bloque a todos los nodos.
5. Los nodos aceptan el bloque sólo si todas las transacciones que contiene son válidas y aún no se han gastado.
6. Los nodos expresan su aceptación del bloque trabajando en la creación del siguiente bloque en esa cadena, utilizando el hash del bloque aceptado como hash anterior.

En caso de recibir más de un bloque simultáneamente, el nodo siempre considera la cadena más larga como la correcta y sigue trabajando en extenderla. Es decir, si dos nodos envían diferentes versiones del siguiente bloque, el nodo que los recibe sigue trabajando sobre el primero recibido pero guarda el otro en caso de que la cadena se haga más larga sobre ese, y en dado caso los nodos que estaban trabajando sobre el otro hacen el cambio al que pertenece a la cadena más larga.

2.1.2.2. Tipos de cadenas de bloques

De acuerdo con Paul (2021), existen tres tipos principales de cadenas de bloques:

1. La cadena de bloques pública: En la cual cualquier interesado puede leer y enviar transacciones.
2. La cadena de bloques privada: En este tipo de cadena de bloques, solo una organización o todas las organizaciones subsidiarias dentro del mismo grupo pueden leer y enviar transacciones.
3. La cadena de bloques híbrida: Esta es una categoría en la que se combinan las *blockchains* públicas y privadas para facilitar las transacciones, y su mayor beneficio es que son altamente personalizables.

Hay otro tipo de *blockchain* semi-descentralizada, la comunitaria o de consorcio: En este tipo de Blockchain, varios grupos de organizaciones forman un consorcio y pueden enviar transacciones y leer datos transaccionales.

2.1.2.3. Bases de datos centralizadas

A diferencia de las bases de datos descentralizadas como las cadenas de bloques, Silberschatz et al. (2020) nos dice que las bases de datos centralizadas son sistemas de información que almacenan datos en un solo servidor. Los datos se pueden acceder desde cualquier lugar de la red, pero solo se pueden modificar en el servidor central. Entre sus ventajas destacan su eficiencia y facilidad de administración, mientras que en sus desventajas se encuentran que los datos almacenados en una base de datos centralizada son un objetivo atractivo para los ataques cibernéticos y que si el servidor central falla, los datos pueden perderse o quedar inaccesibles.

2.1.3. La metodología propuesta para abordar la solución

2.1.3.1. Metodología ágil

Para Montilva et al. (2021), una metodología ágil es un conjunto de técnicas aplicadas en ciclos de trabajo cortos que se basa en la iteración, la colaboración y la adaptación. Las metodologías ágiles conllevan una serie de elementos que aseguran la comunicación, interacción y productividad, de forma que el producto funcional se obtenga lo más rápido posible.

2.1.3.2. SCRUM

SCRUM, según Schwaber (2009), es un marco de trabajo ágil que se utiliza para el desarrollo de software. Se basa en la idea de que el software se debe desarrollar iterativamente, en ciclos cortos o *sprints*, y que el cliente debe estar involucrado en el proceso de desarrollo de principio a fin. Es por esto que se ha escogido SCRUM como metodología de gestión de este proyecto, dado que el mismo cuenta con poco tiempo disponible para su desarrollo y se tiene el apoyo de los diferentes interesados para formar parte de la evaluación y pruebas del producto conforme avanza su desarrollo.

2.1.3.3. Diseño de sistemas

De acuerdo con Cáceres (2014), esto implica investigar sistemas existentes y necesidades de información actuales y futuras para identificar lo que está funcionando correctamente, lo que está fallando, lo que es redundante y lo que falta, para poder planear y desarrollar un nuevo sistema que resuelva los problemas identificados y, en ocasiones, puede implicar cambios significativos. El diseño es una planificación detallada que garantiza que el nuevo sistema cumpla con los requisitos establecidos previamente, donde en sistemas manuales se materializa, pero en sistemas informatizados, el diseño es la etapa en la que se crea la estructura, con los analistas como arquitectos y los programadores como constructores. En el caso de sistemas informatizados, el desarrollo de software se convierte en la fase final, donde los programadores implementan el nuevo sistema siguiendo las pautas del diseño para materializarlo.

2.1.3.3.1. Diseño arquitectónico de sistemas

El diseño arquitectónico de un sistema de software según Martin (2018), es un proceso fundamental para garantizar que el sistema satisfaga las necesidades de todas las partes interesadas. Este proceso implica tomar decisiones sobre la organización del sistema, los elementos estructurales que lo componen, sus interfaces, su comportamiento y su composición en subsistemas.

2.1.3.3.2. Pensamiento de Diseño

Es un enfoque centrado en el ser humano que integra factores humanos, empresariales y tecnológicos en la formulación, resolución y diseño de problemas, según (Plattner et al., 2011). Su metodología se basa en la colaboración multidisciplinaria y la mejora iterativa para crear productos, sistemas y servicios innovadores. Se enfoca en satisfacer las necesidades humanas, aboga por mantener la ambigüedad para fomentar la experimentación, abarca la

reutilización y rediseño, y destaca la importancia de hacer tangibles las ideas para facilitar la comunicación.

2.1.3.4. Desarrollo de sistemas

El desarrollo de sistemas según Denis et al. (2015), es fundamentalmente un proceso compuesto por cuatro fases: Planeamiento, análisis, diseño e implementación. La fase de planeamiento implica comprender el propósito de construir un sistema de información y determinar el enfoque para construirlo. En la fase de análisis el equipo del proyecto investiga el sistema actual, identifica oportunidades de mejora y desarrolla un concepto para el nuevo sistema. La fase de diseño determina cómo funcionará el sistema en términos de hardware y software e incluye el desarrollo de una estrategia de diseño, diseño de arquitectura, diseño de interfaz, especificaciones de archivos, bases de datos, etc. Mientras que la fase de implementación implica la construcción del sistema.

2.1.3.4.1. Prototipado de un sistema

El prototipado, según Camburn et al. (2015), es el proceso de creación de una versión preliminar de un sistema o producto para probar y evaluar su diseño, funcionalidad y rendimiento. Este enfoque implica crear una versión simplificada o parcial del producto final, a menudo utilizando ciclos de desarrollo rápidos e iterativos, para recopilar comentarios, identificar problemas potenciales y realizar las mejoras necesarias antes de la producción a gran escala. El objetivo de la creación de prototipos es mejorar el resultado del proceso de desarrollo del producto al permitir pruebas tempranas y refinamiento del diseño.

2.1.3.4.2. Pruebas unitarias

Éstas suelen ser el primer paso en la evaluación de software, según Khorikov (2020), las pruebas unitarias son pruebas automatizadas que verifican un pequeño fragmento de

código, llamado unidad, y lo hacen relativamente rápido y de forma aislada, teniendo como meta habilitar el crecimiento sostenible de los proyectos de software.

2.1.3.4.3. Pruebas de integración

De acuerdo con Montilva et al. (2021), estas son las pruebas que se encargan de evaluar las interacciones entre los diferentes componentes del sistema. Son el último conjunto de pruebas que se le realizan al sistema antes de pasar a las pruebas del sistema completo como parte del proceso de evaluación o validación del sistema.

2.1.3.5. Evaluación de sistemas

Interpretando lo señalado por Montilva et al. (2021), la evaluación o validación de los sistemas forma parte de los procesos de soporte a la gestión y desarrollo de productos y servicios de software, donde es importante estudiar el aspecto operacional del sistema, el impacto organizacional, la opinión de los administradores y el desempeño del desarrollo.

2.1.3.5.1. Viabilidad de un sistema

La viabilidad de un sistema es un factor importante a tener en cuenta en el desarrollo de sistemas de información. Se evalúa en función de cuatro factores principales: la viabilidad técnica, la viabilidad económica, la viabilidad organizacional y la viabilidad de comportamiento. El estudio de viabilidad es una tarea importante que ayuda a los responsables del proyecto a tomar una decisión informada sobre si el proyecto es factible y merece la pena.

2.1.3.6. Sistema confiable

Para Ramos (2017), la confiabilidad es la capacidad de un sistema de mantener un nivel de rendimiento esperado bajo condiciones determinadas. Se suele expresar en términos

de probabilidad, es decir, la probabilidad de que el sistema funcione correctamente durante un periodo de tiempo determinado.

2.1.3.7. Sistema seguro

De acuerdo con Fretheim et al. (2023), los sistemas de software seguros son aquellos desarrollados con el objetivo de estar libres de defectos y vulnerabilidades, ser repetibles, mantenibles y bien documentados. En ellos también las medidas de seguridad se aplican a todo el sistema, no sólo al software en sí.

Capítulo 3

Metodología y gestión del proyecto

Conforme a las ideas de Kothari (2004), se identifican 8 categorías fundamentales de metodologías de investigación, a saber: Expositiva, Analítica, Práctica, Esencial, Cuantitativa, Cualitativa, Conceptual y Experimental. En el contexto de este proyecto académico, se optará por una metodología de investigación aplicada, la cual busca resolver de manera inmediata problemas específicos en la sociedad u organizaciones, a diferencia de la investigación esencial que se centra en generalizaciones y teorías para contribuir al conocimiento científico en términos generales.

El enfoque de la investigación adopta una perspectiva explicativa al concentrarse en la aclaración de los elementos más destacados del área de interés, partiendo de una visión general. La investigación se percibe como una herramienta orientadora para posibles temas futuros, sin buscar pruebas concluyentes, sino con el propósito de comprender de manera más efectiva la problemática en cuestión.

3.1 Población y Muestra

La muestra descrita en esta sección está compuesta por los usuarios potenciales del sistema a desarrollar, los cuales nos ayudarán a determinar los requisitos de la solución y validarla al finalizar el desarrollo de la misma.

3.1.1 Población

Cuando nos referimos a la población, hablamos del conjunto de individuos dentro del ámbito abordado por el problema de investigación. Según la definición de Kothari (2004), una población se caracteriza por sus atributos distintivos, por ende, el grupo de elementos que comparten dichos atributos se identifica como población o universo. En este estudio, la población considerada estará conformada por profesionales de la salud y pacientes asociados al IAHULA.

3.1.2 Muestra

Conforme a la perspectiva de Kothari (2004), una muestra se concibe como "una porción o fragmento del total de unidades de observación o análisis que ha sido seleccionado". En este trabajo, formarán parte de la muestra sólo un grupo representativo de la población que incluye alrededor de cuatro profesionales de la salud y cinco pacientes regulares del IAHULA.

3.1.3 Tipo de Muestra

En este proyecto, se empleará un tipo de muestra intencional u opinático, según la explicación de Kothari (2004), este es donde los elementos son seleccionados en base a criterios o juicios predefinidos por el investigador. Los criterios seleccionados para este proyecto fueron que los participantes fueran pacientes y profesionales de la salud vinculados al IAHULA con la suficiente experiencia en la dinámica del instituto para tener un criterio acertado con respecto al actual sistema de identificación del mismo.

3.2. Método de recolección de datos

Según Kothari (2004), después de que se ha definido un problema de investigación comienza la recopilación de datos para obtener información sobre un fenómeno o problema de interés. En este proyecto se utilizará el método de entrevistas, las cuales consisten de estímulos orales-verbales, donde las respuestas se presentan en términos de respuestas orales-verbales. Este método se puede utilizar mediante entrevistas personales y, si es posible, mediante entrevistas telefónicas o videoconferencias dependiendo de la disponibilidad de la persona entrevistada.

3.3. Procesamiento y codificación de datos

Una vez recolectados, los datos deben procesarse y analizarse de acuerdo con el plan establecido durante la elaboración del plan de investigación. Este proceso es esencial para un estudio científico y asegura la disponibilidad de datos pertinentes para realizar comparaciones y análisis previstos. Técnicamente, el procesamiento involucra la edición, codificación, clasificación y tabulación de los datos recopilados para que sean susceptibles al análisis. La fase de análisis implica el cálculo de ciertas medidas y la identificación de patrones de relación entre los grupos de datos. Kothari (2004) también aborda las operaciones de procesamiento, como la edición (revisión de errores y omisiones), codificación (asignación de números a respuestas) y clasificación (organización de datos en grupos homogéneos). Se destaca la distinción entre procesamiento y análisis, aunque algunos expertos opinan que ambas actividades están estrechamente vinculadas.

3.4. Metodología de gestión del proyecto

La metodología seleccionada fue Scrum, la cual según Montilva et al. (2021) es un marco metodológico iterativo e incremental, por lo que se ajusta perfectamente al propósito de este proyecto, el cual, mediante una serie de *sprints*, propone la elaboración de un producto mínimo viable que cumpla con los objetivos planteados. Las fases aplicadas de esta metodología serán:

3.4.1. Planificación

En esta etapa, se identifican las tareas clave y se recopila información detallada sobre el proyecto. Un representante del cliente elabora el "Backlog del Producto", enumerando las funciones o necesidades. Posteriormente, se lleva a cabo una reunión donde este documento es esencial para que el equipo conozca las funcionalidades a desarrollar, planificando cómo abordar la primera fase del producto final.

Dada la naturaleza del proyecto, en esta fase se utilizará una metodología complementaria llamada Pensamiento de Diseño en los procesos relacionados al diseño del producto resultante, ya que enfatiza la importancia de la empatía con los usuarios, la creatividad y la iteración continua como se expone en (Anderson, 2022), lo que asegura que la solución final sea verdaderamente efectiva y satisfaga las necesidades de los pacientes y profesionales de la salud, contribuyendo así a la mejora de la atención médica y de la privacidad de los pacientes. Las fases de esta metodología son las siguientes:

- **Fase 1. Empatizar:** La cual busca comprender las necesidades de los usuarios.
- **Fase 2. Definir:** En esta se establecen los problemas a resolver.
- **Fase 3. Idear:** Donde se generan soluciones innovadoras.
- **Fase 4. Prototipar:** En la cual se desarrollan prototipos que resuelvan los problemas.
- **Fase 5. Poner a prueba:** En esta finalmente se implementan los prototipos en el mundo real y se obtiene retroalimentación.

3.4.2. Ejecución

La fase de ejecución, también conocida como *Sprint*, se considera un intervalo de tiempo, con una duración máxima de un mes, durante el cual se desarrolla el producto potencialmente entregable siguiendo el "*Sprint Backlog*", una lista de actividades derivada del "*Product Backlog*". Esta etapa se puede visualizar como un mini proyecto en el que el equipo se centra en desarrollar tareas para alcanzar los objetivos definidos en la planificación del Sprint. En este proyecto académico, esta fase incluirá la implementación de las funcionalidades, como se abordará en el próximo capítulo.

3.4.3. Control

Esta fase implica medir el progreso del proyecto. El líder del equipo actualiza los gráficos al finalizar cada Sprint. Tanto la fase de control como las de planificación y ejecución se realizan de manera iterativa hasta alcanzar el producto final. En este proyecto

de grado, la fase de control se manifestará cada vez que se complete cada iteración, mediante reuniones que discutan avances, mejoras y ajustes de manera iterativa.

3.4.4. Plan de proyecto

Para la ejecución del proyecto, se adoptará un enfoque ágil siguiendo la metodología SCRUM utilizando incrementos o *sprints* con duración de dos semanas. Esta decisión se basa en la necesidad de mantener un ritmo constante de desarrollo y adaptación. La duración semanal de los incrementos permitirá ajustarse eficientemente a los plazos establecidos por la universidad, asegurando la entrega oportuna de resultados y la culminación exitosa del proyecto dentro del tiempo especificado. Los *sprints* estarán compuestos de las siguientes actividades:

1. ***Sprint* 1:**

    a. Planeamiento de *sprint*
    b. Recolección de datos
    c. Análisis de datos
    d. Selección de la metodología de desarrollo de software
    e. Revisión bibliográfica sobre la metodología de desarrollo de software
    f. Revisión de *sprint*
    g. Retrospectiva del *sprint*

2. ***Sprint* 2:**

    a. Planeamiento de *sprint*
    b. Gestión inicial del proyecto según la la metodología de desarrollo de software
    c. Redacción de la lista de interesados
    d. Elaboración de la descripción del problema
    e. Elaboración de la visión del producto
    f. Planificación inicial del proyecto de desarrollo
    g. Análisis y especificación inicial de requisitos
    h. Elaboración el informe de descripción del contexto del sistema
    i. Elaboración de la lista del producto (*Product Backlog*) con requisitos e historias de usuarios
    j. Documentación de la definición de requisitos
    k. Revisión de *sprint*
    l. Retrospectiva del *sprint*

3. ***Sprint* 3:**

    a. Planeamiento de *sprint*
    b. Modelado funcional y estructural de los requisitos
    c. Elaboración del documento de requisitos
    d. Diseño arquitectónico inicial de la aplicación
    e. Elaboración de la lista de requisitos arquitectónicos
    f. Elaboración del diagrama estructural de la arquitectura inicial de la aplicación
    g. Revisión de *sprint*
    h. Retrospectiva del *sprint*

4. ***Sprint* 4:**

    a. Planeamiento de *sprint*
    b. Diseño de la interfaz gráfica de usuario
    c. Elaboración del documento inicial de diseño
    d. Elaboración del plan de entregas e incrementos
    e. Actualización del Plan del Proyecto
    f. Desarrollo y pruebas del incremento 1
    g. Entrega del incremento 1
    h. Revisión de *sprint*
    i. Retrospectiva del *sprint*

5. ***Sprint* 5:**

    a. Planeamiento de *sprint*
    b. Desarrollo y pruebas del incremento 2
    c. Entrega del incremento 2
    d. Revisión de *sprint*
    e. Retrospectiva del *sprint*

6. ***Sprint* 6:**

    a. Planeamiento de *sprint*
    b. Desarrollo y pruebas del incremento 3
    c. Entrega del incremento 3
    d. Revisión de *sprint*
    e. Retrospectiva del *sprint*

7. ***Sprint* 7:**

a. Planeamiento de *sprint*
b. Desarrollo y pruebas del incremento 4
c. Entrega del incremento 4
d. Revisión de *sprint*
e. Refinamiento de los productos de la metodología de desarrollo de software

3.5. Metodología de desarrollo de software

De acuerdo con Montilva et al. (2021), el Blue Watch es un método para el desarrollo y operación de sistemas de software. Este combina diferentes aspectos de los enfoques ágiles y disciplinados de la Ingeniería del Software para proporcionar una guía práctica adaptable a las necesidades particulares de sus usuarios. De la misma forma actúa como un marco metodológico que proporciona una estructura conceptual y de trabajo para resolver un problema, desarrollar una solución o prestar un servicio, además que ayuda a los equipos de desarrollo y operación de software de una empresa u organización a crear sus propios métodos mediante la adaptación de un marco de trabajo completo. El método está compuesto por tres modelos:

- **Modelo de productos.** Identifica y describe los productos intermedios y finales que se deben producir durante el desarrollo de un sistema de software. Este modelo contiene un conjunto de plantillas que facilitan la documentación de los informes técnicos y de gestión que son necesarios para desarrollar el sistema.
- **Modelo de actores.** Identifica y describe los roles que se requieren para desarrollar y operar sistemas de software. Asigna las responsabilidades de cada rol y propone estructuras para organizar los Equipos de Desarrollo y Operaciones.
- **Modelo de procesos**. Describe los procesos técnicos, de gestión y de soporte que los Equipos de Desarrollo y Operaciones deben ejecutar para producir un sistema de software.

El Modelo Principal de Procesos del método describe el orden o secuencia en que los procesos técnicos, de gestión y de soporte del desarrollo de aplicaciones o sistemas de software deben ejecutarse.

Como puede apreciarse en este modelo, el desarrollo de una aplicación sigue el orden descrito a continuación:

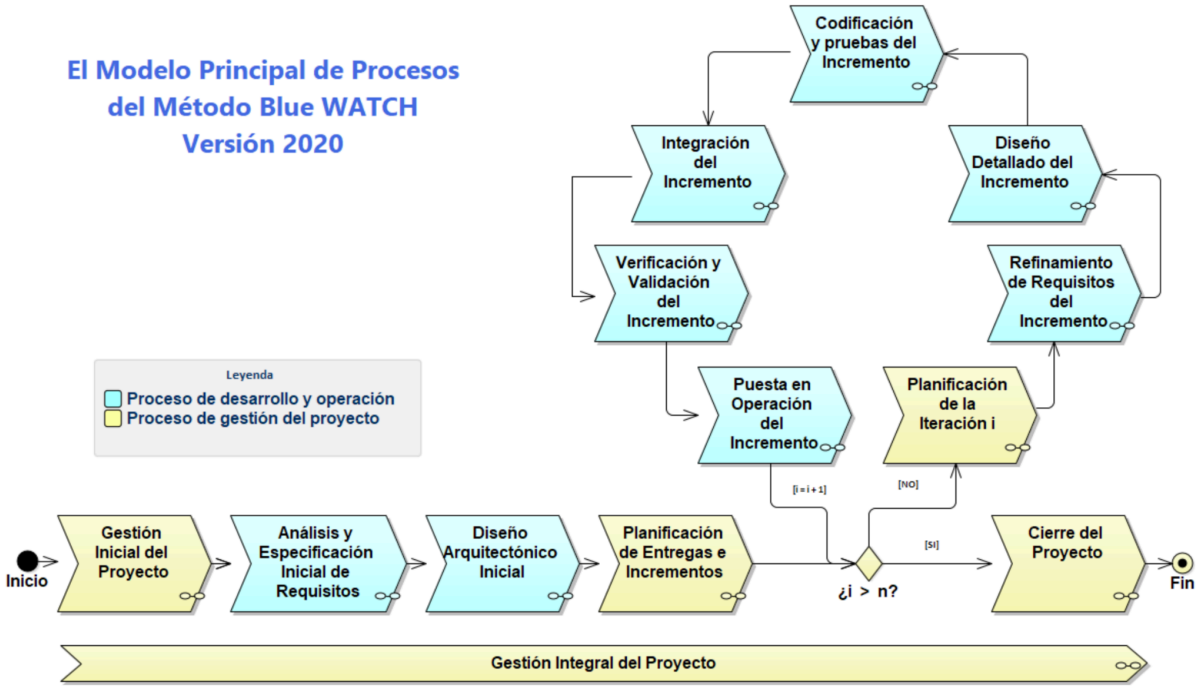


Figura 3.1: Modelo principal de procesos de Blue Watch

3.5.1. Gestión inicial del proyecto

En esta etapa del proyecto se designó el líder de proyecto, el cual en este caso fue el profesor Jonás Montilva, tutor de este proyecto que guió y supervisó las decisiones tomadas a lo largo de la planificación, gestión, desarrollo e implementación de la solución propuesta. De igual manera, se describen a continuación algunos aspectos fundamentales previos al inicio del proyecto.

3.5.1.1. Lista de interesados

Los interesados en el proyecto son aquellas personas u organizaciones que de una u otra forma se beneficiarán del desarrollo del sistema propuesto como solución al problema planteado en la descripción del problema, los cuales en este proyecto son:

1. Médicos que trabajan en alguna institución de salud pública o privada, o de forma independiente en consultorios que no están relacionados a ningún hospital o clínica pero están registrados en el ministerio de salud.
2. Pacientes de los médicos mencionados anteriormente.
3. Enfermeros y enfermeras que trabajan en alguna institución de salud pública o privada, o de forma independiente en consultorios que no están relacionados a ningún hospital o clínica pero están registrados en el ministerio de salud.
4. Instituciones de salud públicas, privadas o consultorios independientes registrados en el ministerio de salud.
5. Ministerio de salud.

3.5.1.2. Descripción del problema

En el Hospital Universitario de los Andes y varias otras instituciones de salud, la gestión de la identificación de pacientes y el almacenamiento de historias médicas se realiza de manera manual, utilizando hojas de papel que se archivan en un sistema físico centralizado. Este método presenta una serie de desafíos y limitaciones significativas que afectan la eficiencia y la seguridad de la atención médica.

Primero, el proceso manual de identificación de pacientes es propenso a errores humanos, lo que puede resultar en confusiones en la atención médica y en la pérdida de información crítica. Además, el archivo físico único dificulta el acceso rápido y eficiente a la información del paciente por parte de los diferentes departamentos del hospital, lo que puede retrasar los tratamientos y la toma de decisiones médicas.

Otro aspecto preocupante es la seguridad y la integridad de los datos. El almacenamiento en papel no solo es vulnerable a daños físicos, como incendios, inundaciones o deterioro por el paso del tiempo, sino que también carece de medidas efectivas de respaldo y recuperación en caso de pérdida de información. Esto pone en riesgo la continuidad de la atención médica y la privacidad de los pacientes.

Además, el hecho de que los pacientes no reciban copias de sus historias médicas dificulta su participación en su propio cuidado de salud y limita su capacidad para compartir información relevante con otros proveedores de atención médica.

En este contexto, la implementación de un Sistema de Gestión de Identificación Electrónica basado en tecnología *Blockchain* emerge como una solución integral para abordar estos desafíos. Esta tecnología tiene la capacidad de permitir la creación de registros electrónicos precisos y actualizados, accesibles desde cualquier punto dentro y fuera del hospital y con medidas sólidas de seguridad y respaldo de datos. Además, promueve la participación activa de los pacientes en la gestión de su salud al permitirles acceder a sus historias médicas y compartir información con otros profesionales de la salud de manera segura.

La implementación de este sistema no solo puede mejorar la eficiencia y la calidad de la atención médica en el Hospital Universitario de los Andes, sino que también sienta las bases para una práctica médica más segura, transparente y centrada en el paciente.

3.5.1.3. Visión del producto

3.5.1.3.1. Usuarios potenciales del sistema

- Personal médico: Médicos, enfermeras y enfermeros que necesitan acceder a la información del paciente para proporcionar atención médica adecuada y oportuna.
- Personal administrativo de las instituciones médicas: Personal encargado de gestionar la admisión de médicos, enfermeros y enfermeras a la institución.
- Personal administrativo del ente gubernamental: Personal encargado de gestionar la admisión de instituciones médicas a la red de instituciones autorizadas para brindar servicios médicos.
- Pacientes: Individuos que buscan acceder a su información médica y participar activamente en su cuidado de salud, teniendo la autoridad sobre sus datos.

3.5.1.3.2. Necesidades de los usuarios

- **Personal médico:** Acceso rápido y fácil a la información del paciente, así como manejar su propia información asociada a la institución.
- **Personal administrativo de las instituciones médicas**: Eficiencia en la gestión de identificación de médicos asociados a la institución.
- **Personal administrativo del ente gubernamental:** Eficiencia en la gestión de identificación y autorización de instituciones médicas en su red.
- **Pacientes:** Facilidad en el registro y acceso de datos médicos.

3.5.1.3.3. Objetivos del sistema

- **Descentralizar la Información del paciente**: Crear una plataforma interinstitucional que almacene y gestione de manera segura los registros médicos de todos los pacientes de diferentes instituciones médicas.
- **Acceso eficiente:** Permitir un acceso rápido y fácil a la información del paciente para el personal médico y administrativo, desde cualquier punto dentro del hospital.
- **Facilitar la participación del paciente**: Brindar a los pacientes un fácil registro y acceso a sus datos médicos.
- **Seguridad y confidencialidad:** Implementar medidas sólidas de seguridad para proteger la privacidad de los datos del paciente.

3.5.1.3.4. Beneficios para los usuarios

- **Mejora en la eficiencia:** Reducción de tiempos de búsqueda de información, agilización de procesos administrativos y optimización del flujo de trabajo en el hospital.
- **Mayor seguridad y confidencialidad:** Protección de la integridad y la privacidad de los datos del paciente mediante medidas de seguridad apropiadas.
- **Mejora en la calidad de la atención:** Facilitación de una atención médica más coordinada, precisa y personalizada, basada en un acceso rápido y completo a la información del paciente.

3.5.1.4. Plan inicial del proyecto

Se plantea desarrollar un sistema de gestión de identidad de pacientes en instituciones de salud que implemente mecanismos de descentralización, privacidad y seguridad que mejore la calidad de la atención médica y fomente la interoperabilidad entre instituciones de salud en beneficio de pacientes y profesionales médicos utilizando tecnología blockchain.

3.5.1.4.1. Requisitos del proyecto

- Descentralización de los datos médicos de los pacientes para ser accedidos de manera fácil y oportuna desde diferentes instituciones y sus departamentos.
- Diseño de una interfaz intuitiva y fácil de usar para el personal médico, administrativo y los pacientes.
- Implementación de medidas sólidas de seguridad y protección de datos para garantizar la confidencialidad y la integridad de la información del paciente.
- Desarrollo de funciones de búsqueda y filtros para acceder rápida y eficazmente a la información del paciente.

3.5.1.4.2. Selección del personal de desarrollo

El personal de desarrollo de este proyecto estuvo conformado por dos ingenieros de software con experiencia en el desarrollo de sistemas de gestión de información y conocimientos en tecnologías web y bases de datos. Debido a las limitaciones de personal disponible para el proyecto, los mismos ingenieros se encargarán de las pruebas de software para garantizar la funcionalidad, seguridad y rendimiento del sistema desarrollado.

3.5.1.4.3. Plataformas y herramientas de desarrollo

Considerando que el resultado de este proyecto es un producto mínimo viable, las plataformas de hardware utilizadas constaron de las tres computadoras personales de los ingenieros a cargo de su desarrollo, con sistemas operativos Windows y MacOS.

Los lenguajes de programación escogidos para la construcción de la solución fueron Golang, debido a su rendimiento, seguridad, simplicidad y capacidad para manejar sistemas distribuidos, y TypeScript, el cual es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases al mismo.

Para la interfaz gráfica de usuario, de la mano de TypeScript se escogió la librería React.js en conjunto con el *framework* Next.js para desarrollar una interfaz de usuario moderna y receptiva, aprovechando los beneficios del *framework* para agilizar el desarrollo. A su vez, las herramientas utilizadas para lograr el uso de la aplicación en diferentes dispositivos fueron Capacitor, que es un entorno de ejecución nativo para dispositivos móviles, y Electron, que es un *framework* de código abierto que permite desarrollar aplicaciones de escritorio utilizando tecnologías web.

En cuanto a la base de datos, se empleó PostgreSQL para almacenar algunos datos del sistema de manera segura y confiable, acotando que la información médica se almacena en la cadena de bloques implementada con Hyperledger Fabric, el cual es un marco de desarrollo blockchain de código abierto mantenido por la Fundación Linux altamente personalizable.

En relación al despliegue de la capa de negocios y datos, se utilizó Docker para aislar de forma portable y eficiente los componentes de estas capas, ya que sus contenedores pueden ejecutarse en cualquier máquina con Docker instalado, sin importar el sistema operativo subyacente, lo que facilita su migración y el despliegue.

Capítulo 4

Análisis y definición de requisitos

En esta fase inicial se establecieron las características técnicas de la aplicación, tanto funcionales como no funcionales. Este capítulo representa el documento denominado "Documento de Requisitos" que sugiere la metodología Blue Watch. Este documento se creó y se fue refinando a lo largo del proyecto, tomando en cuenta las diferentes etapas e iteraciones del mismo.

4.1. Análisis del contexto del sistema

4.1.1. Identificación del contexto del sistema

El contexto del sistema son los sistemas de salud, ya que es en ellos donde se podría implementar el producto final de este proyecto de grado.

4.1.1.1. Identificación de los objetivos del contexto del sistema

El objetivo del contexto del sistema es proveer a pacientes un servicio de salud eficiente y efectivo.

4.1.1.2. Identificación de los principales conceptos presentes en el contexto

- Pacientes.

- Médicos.
- Instituciones de salud.
- Ente gubernamental.
- Identidad.

4.1.1.3. Identificación de los actores que participan en el contexto del sistema

- Pacientes.
- Personal de salud:
    - Médicos.
    - Enfermeras.
- Personal administrativo de la institución de salud.
- Personal del ente gubernamental.

4.1.2. Identificación de los procesos y actividades que ejecutan los actores

- Pacientes:
    - Consultas médicas
    - Realizar exámenes médicos.
- Médicos:
    - Pasar consulta.
    - Solicitar exámenes.
    - Requerir/Administrar medicación al paciente.
- Enfermeros:
    - Auxiliar a los médicos durante procedimientos.
    - Administrar medicamentos.
- Personal administrativo:
    - Contratar/Despedir médicos y enfermeros.
    - Cobranzas y pagos.
- Ente gubernamental:

- Aprobar instituciones de salud en la red del sistema de salud.

4.1.3. Identificación de las necesidades y preocupaciones de los actores

De las entrevistas realizadas a pacientes y personal médico, se identificaron las siguientes necesidades y preocupaciones en los interesados del sistema:

- **Pacientes:** Las necesidades de los pacientes están directamente relacionadas a tener fácil acceso a sus datos médicos y poder compartirlos con el personal de salud que sea necesario, incluso en diferentes instituciones de salud. Sus preocupaciones se enfocan en que estos datos estén seguros y tener completa potestad sobre quién tiene acceso a ellos.
- **Médicos:** Los médicos necesitan que la información médica de los pacientes que atienden sea de fácil acceso durante el proceso de atención del paciente, así como que sea completa e independiente de la caligrafía humana. Las preocupaciones de los médicos están relacionados a la disponibilidad de los registros de los pacientes, es decir, que no dependa de su utilización en otros departamentos de la institución.
- **Enfermeras y enfermeros**: Las necesidades de las enfermeras y enfermeros relacionadas a la manipulación de la información de los pacientes son el acceso a los datos básicos de los pacientes y de su ubicación intrahospitalaria al estar hospitalizados para poder identificarlos inequívocamente al momento de suministrar medicamentos.
- **Personal administrativo de la institución de salud:** El personal administrativo de la institución de salud necesita poder gestionar el acceso a la plataforma de los médicos y enfermeras asociados a dicha institución. Sus preocupaciones son que este proceso sea fácil, rápido y eficiente.
- **Personal del ente gubernamental:** El personal del ente gubernamental necesita poder gestionar el acceso a la plataforma de las instituciones legalmente constituidas dentro de su red. Sus preocupaciones son que este proceso sea fácil, rápido y eficiente.

4.1.4. Identificar y analizar los procesos y actividades que el sistema modificará

- Gestión de identidad de pacientes en las instituciones de salud.
- Verificación de instituciones de salud en el sistema de salud por entes gubernamentales.
- Verificación de médicos, enfermeros y enfermeras en la institución de salud por el personal administrativo de la misma.

4.2. Identificación de requisitos

4.2.1 Identificación y clasificación de nuevos interesados y usuarios del sistema

- **Interesados internos:**
  - Desarrolladores de software.
  - Dueño del producto.
  - Líder del equipo.
- **Interesados externos:**
  - Dueños de clínicas y consultorios.
- **Interesados directos:**
  - Personal del departamento de salud del gobierno.
  - Personal administrativo de una institución de salud.
  - Personal médico de la institución de salud (Médicos, enfermeras y enfermeros).
  - Pacientes (Cualquier usuario que recibe cuidados por la institución de salud).
- **Interesados indirectos:**
  - Familiares de los pacientes.

- Bioanalistas, imagenólogos y demás profesionales que se encarguen de realizar análisis médicos, prótesis, férulas, etc.
- Farmaceutas.

4.2.2. Definición de los requisitos del dominio de la aplicación

De acuerdo a lo descrito por los médicos entrevistados y su conocimiento de la Ley del Ejercicio de la Medicina de la República Bolivariana de Venezuela, los requisitos del dominio de la aplicación son:

1. Gestión de Identidad del Paciente:

- **Registro Seguro**: El sistema debe permitir el registro seguro de pacientes con información personal.
- **Identidad Única**: Cada paciente debe tener una identidad única e inmutable en la red blockchain para evitar duplicaciones y fraudes.
- **Autenticación Robusta**: El sistema debe implementar mecanismos de autenticación robusta para garantizar el acceso autorizado a los datos de los pacientes, como claves criptográficas o biometría.
- **Control de Acceso Granular**: El sistema debe permitir un control de acceso granular sobre los datos de los pacientes, definiendo quién puede acceder a qué información y en qué circunstancias.

2. Gestión de Consentimientos:

- **Suministro de Consentimiento Informado**: El sistema debe facilitar el suministro de consentimiento informado de los pacientes para la recolección, uso y divulgación de sus datos de salud.
- **Gestión de Revocación de Consentimiento**: El sistema debe permitir a los pacientes revocar sus consentimientos en cualquier momento, actualizando automáticamente los permisos de acceso a sus datos.

4. Gobernanza y Seguridad:

- **Permisos y Roles**: El sistema debe definir un esquema de permisos y roles para los usuarios, restringiendo el acceso a las funcionalidades según su nivel de autorización.
- **Mecanismos de Seguridad**: El sistema debe implementar mecanismos de seguridad robustos para proteger la información confidencial de los pacientes, como criptografía de datos y control de acceso.
- **Cumplimiento de Normativas**: El sistema debe cumplir con las normativas de privacidad de datos y protección de la salud aplicables en cada jurisdicción.

5. Escalabilidad:

- **Arquitectura Escalable**: El sistema debe diseñarse con una arquitectura escalable para adaptarse al crecimiento del número de usuarios y la cantidad de datos.

4.2.3. Definición de los requisitos del proyecto

Mediante las entrevistas realizadas a los diferentes interesados del proyecto, se pudieron visualizar los siguientes requisitos funcionales agrupados por épicas extraídas de historias de usuarios similares:

- **Registro de usuarios**
    - **Registro de pacientes**

1. El sistema debe permitir el registro de nuevos pacientes.
2. El sistema debe solicitar información personal básica durante el registro (nombre, fecha de nacimiento, correo electrónico, etc.).

    - **Registro de doctores**

1. El sistema debe permitir el registro de nuevos doctores.
2. El sistema debe solicitar información personal y profesional durante el registro.

    - **Registro de enfermeros y enfermeras**

1. El sistema debe permitir el registro de nuevos enfermeros.
2. El sistema debe solicitar información profesional durante el registro.

- **Registro de usuarios de institución de salud**

1. El sistema debe permitir el registro de nuevos usuarios de instituciones de salud.
2. El sistema debe solicitar información relevante durante el registro.

- **Inicio de sesión de usuarios**

  - **Inicio de sesión de pacientes**

1. El sistema debe permitir que los pacientes inicien sesión con su correo electrónico y contraseña.
2. El sistema debe autenticar a los usuarios antes de permitir el acceso a sus datos.

  - **Inicio de sesión de doctores**

1. El sistema debe permitir que los doctores inicien sesión con su correo electrónico y contraseña.
2. El sistema debe autenticar a los doctores antes de permitir el acceso a los datos.

  - **Inicio de sesión de enfermeros y enfermeras**

1. El sistema debe permitir que los enfermeros inicien sesión con su correo electrónico y contraseña.
2. El sistema debe autenticar a los enfermeros antes de permitir el acceso a las órdenes de los pacientes.

  - **Inicio de sesión de usuarios de institución de salud**

1. El sistema debe permitir que los usuarios de instituciones de salud inicien sesión con su correo electrónico y contraseña.
2. El sistema debe autenticar a los usuarios antes de permitir el acceso a las funciones de procesamiento de solicitudes.

  - **Inicio de sesión de usuarios de gobierno**

1. El sistema debe permitir que los usuarios de gobierno inicien sesión con su correo electrónico y contraseña.
2. El sistema debe autenticar a los usuarios antes de permitir el acceso a las funciones de procesamiento de solicitudes.

- **Acceso a información personal e institucional**
    - **Acceso a información personal de los pacientes**

1. El sistema debe permitir visualizar la información personal de los pacientes a sí mismos y a los doctores, enfermeros y enfermeras con acceso concedido.

    - **Acceso a información personal de los médicos**

1. El sistema debe permitir visualizar la información personal de los médicos a sí mismos, a los pacientes a los que atiende y a las instituciones a las que solicita pertenecer.

    - **Acceso a información personal de los enfermeros y enfermeras**

1. El sistema debe permitir visualizar la información personal de los enfermeros y enfermeras a sí mismos y a las instituciones a las que solicita pertenecer.

    - **Acceso a información institucional de las instituciones de salud**

2. El sistema debe permitir visualizar la información institucional de la institución de salud a sí misma y al ente gubernamental a cuya red solicita pertenecer.

- **Concesión, denegación y revocación de acceso a la información personal**
    - **Concesión de acceso a la información personal de los pacientes**

1. El sistema debe permitir conceder acceso a la información personal de los pacientes a los doctores que lo soliciten.
2. El sistema debe notificar al doctor sobre el acceso otorgado.

    - **Denegación de acceso a la información personal de los pacientes**

1. El sistema debe permitir denegar el acceso a la información personal de los pacientes a los doctores que lo soliciten.

2. El sistema debe notificar al doctor sobre el acceso denegado.

- **Revocación de acceso a la información personal de los pacientes**

1. El sistema debe permitir revocar el acceso a la información personal de los pacientes a los doctores que ya cuentan con acceso.
2. El sistema debe notificar al doctor sobre el acceso revocado.

- **Edición de datos personales y profesionales**
  - **Edición de datos personales de los pacientes**

1. El sistema debe permitir la edición de datos personales como número telefónico y correo electrónico.

- **Edición de datos profesionales de los doctores**

1. El sistema debe permitir la edición de los datos profesionales del doctor.

- **Filtrar y buscar información**
  - **Filtrar y buscar información de los doctores**

1. El sistema debe permitir la búsqueda de doctores por especialidad.
2. El sistema debe permitir filtrar doctores.

- **Filtrar y buscar información de los pacientes**

1. El sistema debe permitir buscar pacientes en el directorio del doctor.
2. El sistema debe permitir filtrar pacientes por criterios como edad, sexo, estatus y fecha de consulta.

- **Gestión de solicitudes**
  - **Gestión de solicitudes para usuarios de institución de salud**

1. El sistema debe permitir listar todas las solicitudes pendientes.
2. El sistema debe permitir listar a los profesionales de salud aceptados.
3. El sistema debe permitir conceder o denegar la solicitud de pertenencia a profesionales de salud.

4. El sistema debe permitir revocar la pertenencia a la institución a profesionales de salud que ya pertenecen a ella.

- **Gestión de solicitudes para usuarios de gobierno**

1. El sistema debe permitir listar todas las solicitudes pendientes.
2. El sistema debe permitir listar a las instituciones aceptadas.
3. El sistema debe permitir conceder o denegar la solicitud de pertenencia de las instituciones.
4. El sistema debe permitir revocar la pertenencia al ministerio a instituciones de salud que ya pertenecen a él.

Además de los requisitos funcionales, también se identificaron los siguientes requisitos no funcionales agrupados por características en común, los cuales se deben asegurar en el desarrollo del proyecto:

- **Validación y seguridad**

1. El sistema debe validar la información ingresada para evitar duplicados.
2. El sistema debe asignar una identidad única al paciente (billetera/dirección) que se usará para todas las transacciones de la *blockchain*.
3. La seguridad de los datos de inicio de sesión debe garantizarse mediante encriptación.
4. La validación de la información debe ser rigurosa para evitar errores.
5. La integridad y consistencia de los datos deben mantenerse después de cada edición o cambio.

- **Interfaz de Usuario**

1. La interfaz debe ser intuitiva y fácil de navegar.
2. La información mostrada debe ser clara y detallada.
3. La aplicación debe ser multiplataforma.

- **Rendimiento**

1. Los cambios deben reflejarse en el sistema en menos de 2 segundos.

4.2.4. Documentación de los requisitos

Diagrama de clases general de la aplicación

Cabe acotar que las clases y casos de uso con fondo de color son externos a este proyecto, pero son parte del proyecto de grado complementario "Desarrollo de un Sistema de Gestión de Historias Clínicas en Instituciones de Salud mediante el uso de tecnología de cadena de bloques".

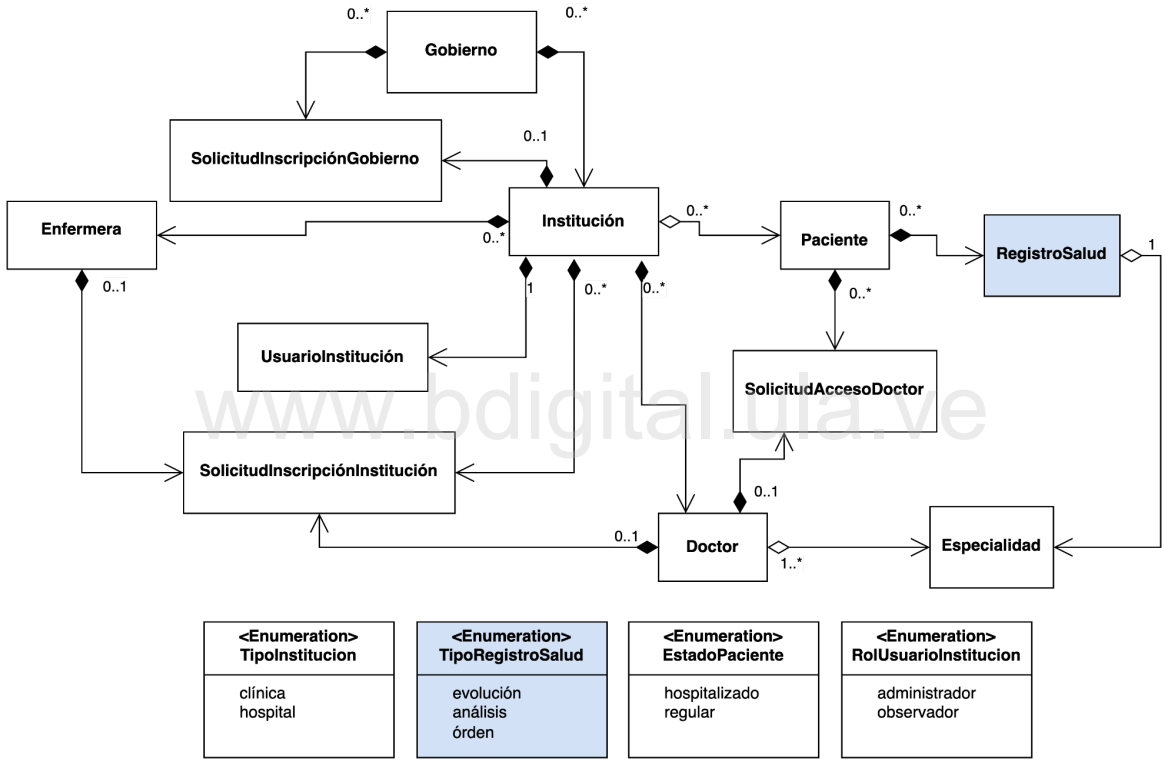
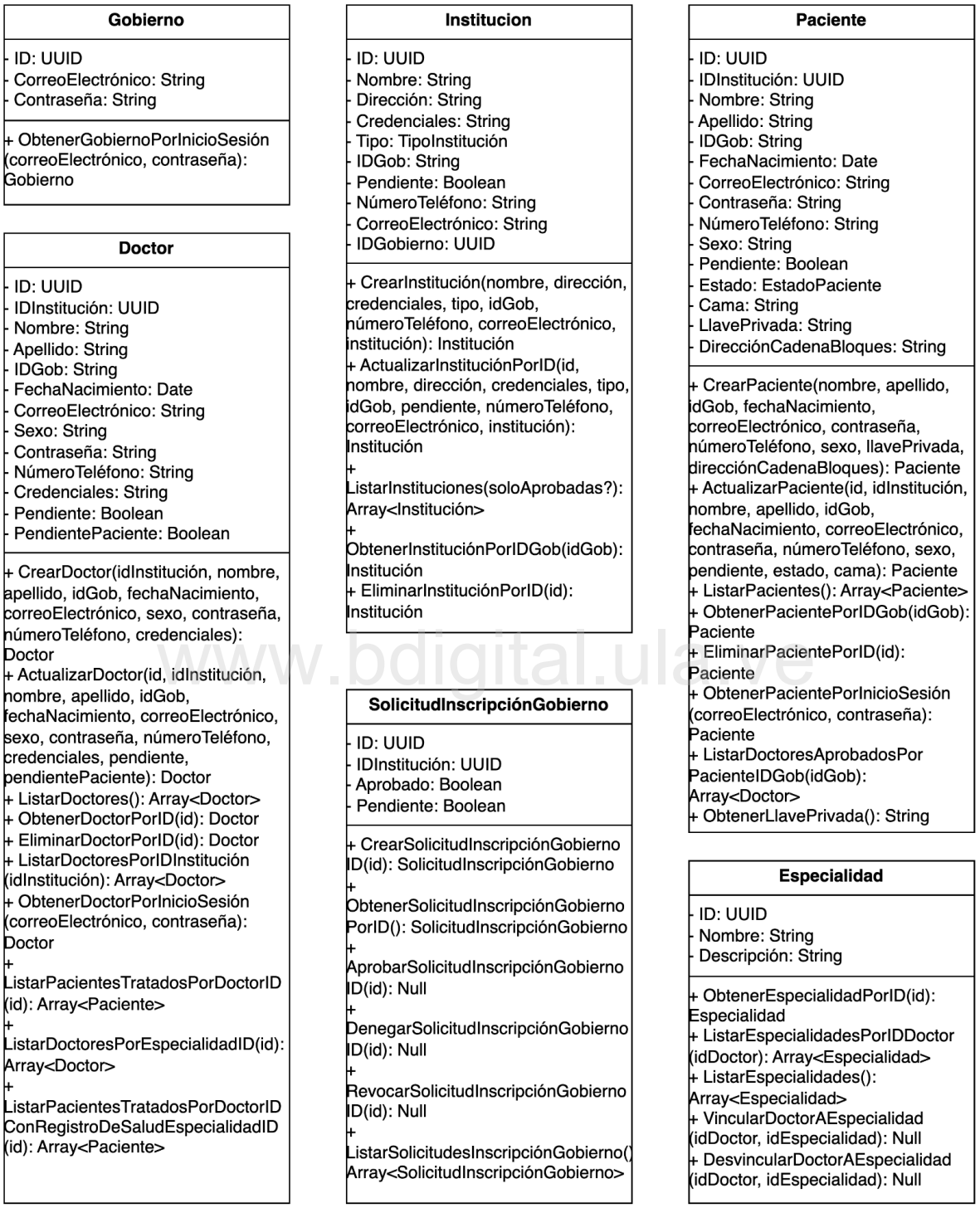


Figura 4.1: Diagrama de clases de la aplicación



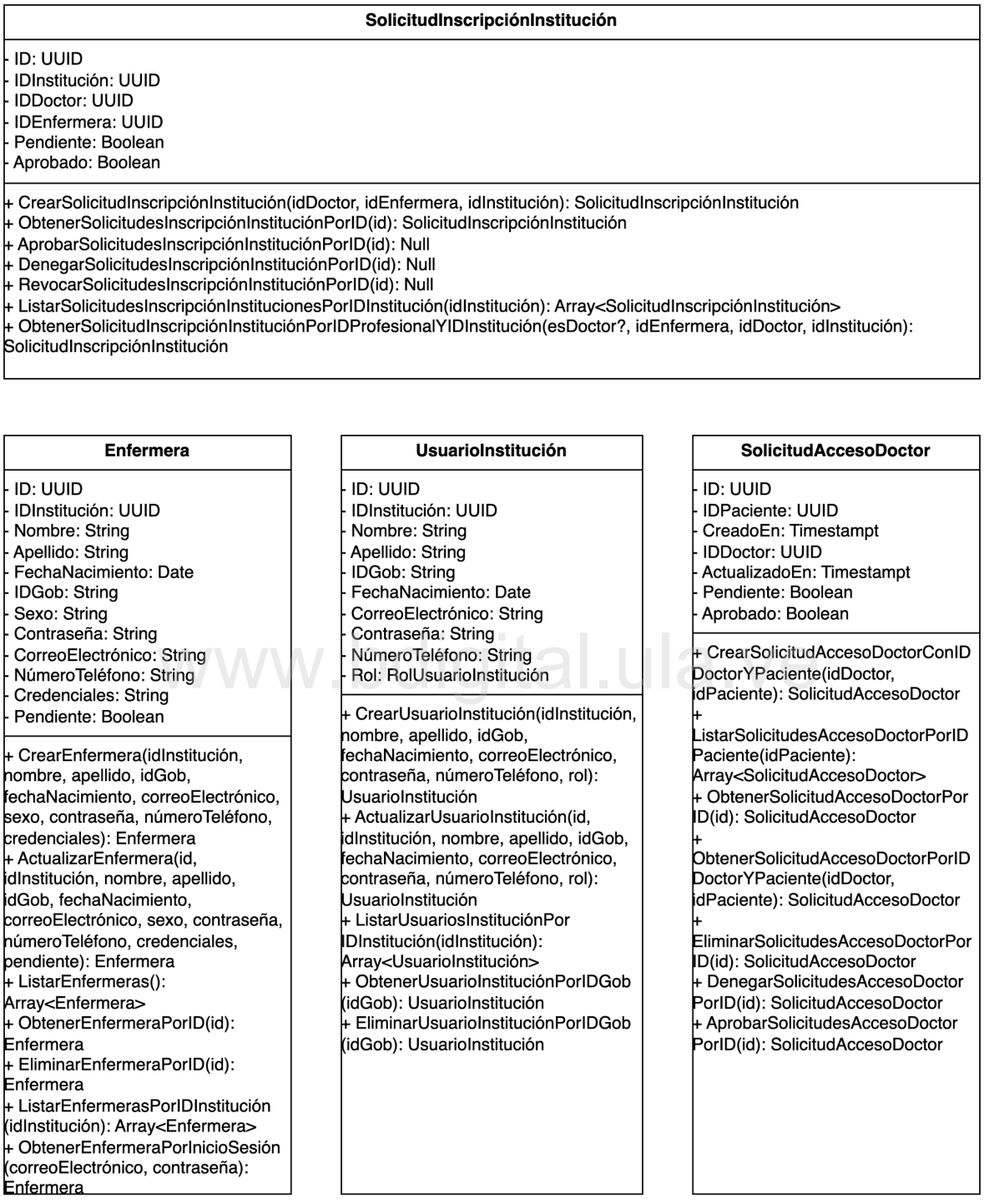


Figura 4.2: Atributos y métodos de las clases de la aplicación

Diagrama de casos de uso

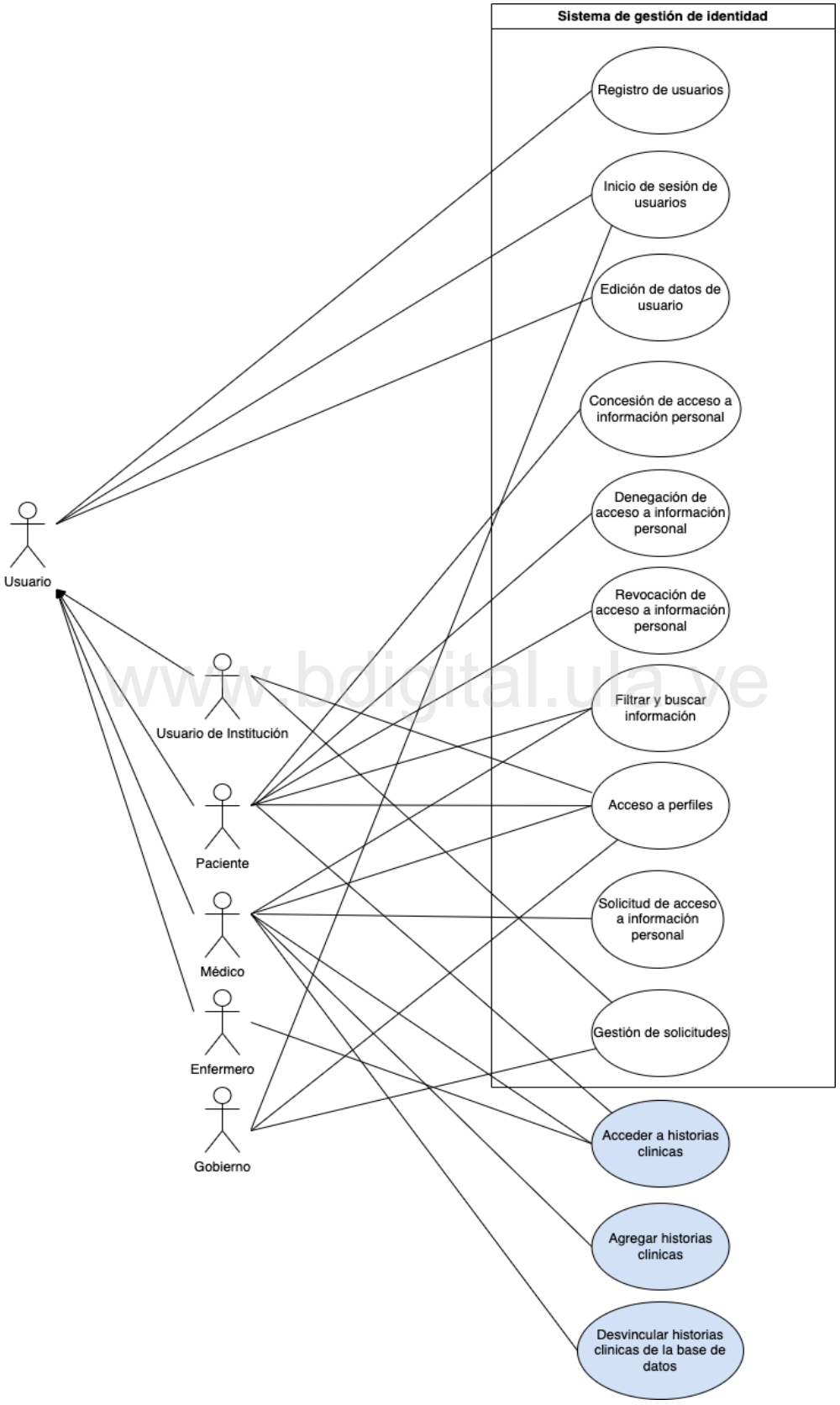


Figura 4.3: Diagrama de casos de uso

Casos de uso	Descripción	Actor
Registro de usuarios	Almacena datos del usuario dentro del sistema, en el caso de los pacientes genera la identidad única.	Paciente, Médico, Enfermero, Usuario de la institución
Inicio de sesión de usuarios	Verifica las credenciales de los usuarios y retorna los tokens para autorización de las peticiones al sistema.	Paciente, Médico, Enfermero, Usuario de la institución
Edición datos de usuario	Modifica los datos personales y profesionales de los usuarios dentro del sistema.	Paciente, Médico, Usuario de la institución
Concesión de acceso a la información personal	Le aprueba el acceso a un usuario de ver la información personal de otro usuario	Paciente
Denegación de acceso a la información personal	Le niega el acceso a un usuario de ver la información personal de otro usuario	Paciente
Revocación de acceso a la información personal	Le revoca el acceso a un usuario de ver la información personal de otro usuario	Paciente
Filtrar y buscar información	Lectura de la información que permite el filtrado en base a criterios especificados	Paciente, Médico
Acceso a perfiles	Lectura de la información de lo usuarios almacenados en el sistemas	Paciente, Médico, Usuario de la institución, Gobierno
Solicitud de acceso a información personal	Crea una solicitud por parte de un usuario para poder acceder a la información personal de otro usuario	Médico
Gestión de solicitudes	Permite la concesión, denegación, revocación y creación de solicitudes de pertenencia a instituciones	Usuario de la institución, Gobierno
Acceso a historias clínicas	Lee la información auxiliar del registro de salud de la base de datos	Paciente, Médico, Enfermero
Crear historias clínicas	Almacena archivos de registros de salud en el servicio externo de almacenamiento, su dirección encriptada en la	Médico

	cadena de bloques y su información auxiliar en la base de datos auxiliar	
Desvincular historias médicas de la base de datos	Elimina toda información auxiliar del registro de salud de la base de datos	Médico

Tabla 4.1: Descripción de casos de uso

Realizaciones de caso de uso (RCU)

A continuación, se muestran las realizaciones de caso de uso de los procesos más ilustrativos de la aplicación, ya que muchos de ellos varían levemente dependiendo del tipo de usuario.

Registro de usuarios

El registro de usuarios es el proceso de guardar los datos de los usuarios en la base de datos. En el caso de los pacientes, es en este proceso en el que se crea una identidad única asociada a una dirección y llave privada irrepetibles para cada paciente, las cuales son posteriormente asociadas al ingresar sus datos en la base de datos.

Figura 4.4: Diagrama de clases de la RCU de Registro de usuarios

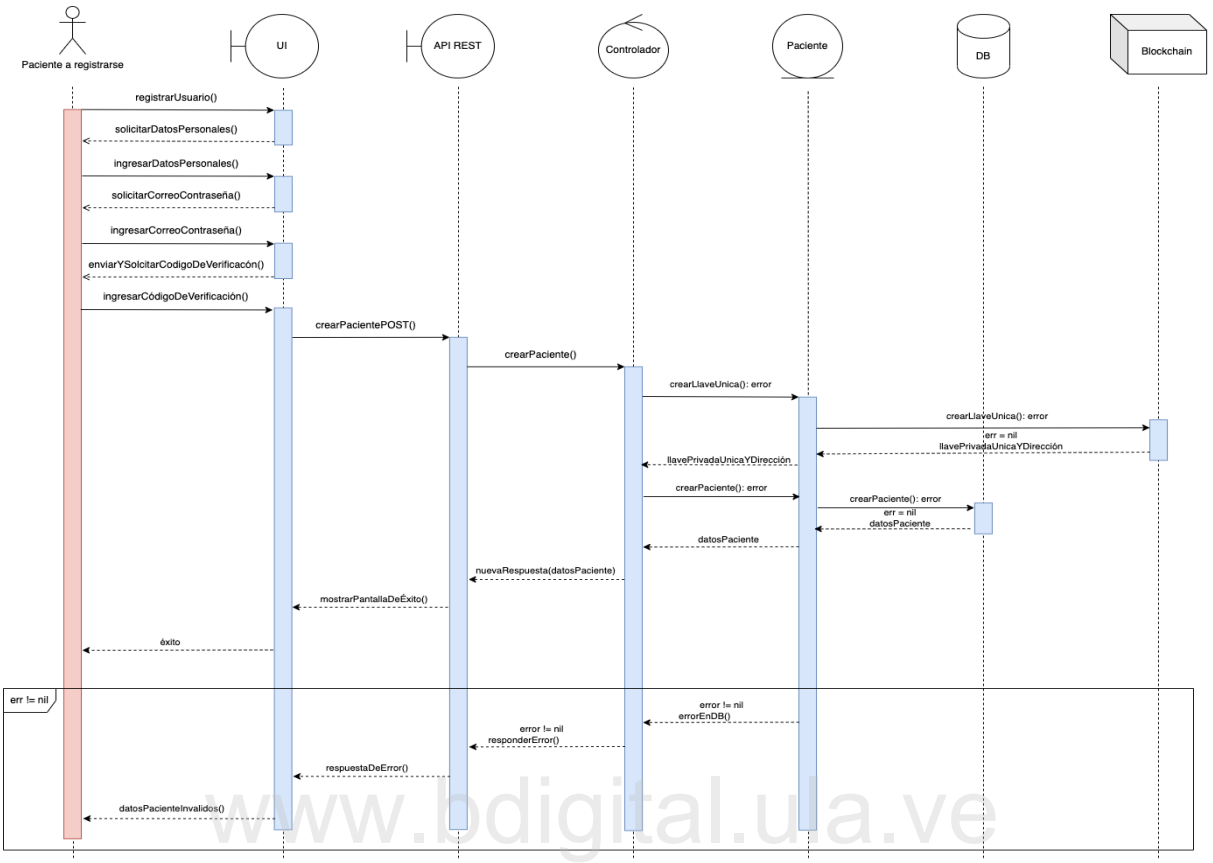


Figura 4.5: Diagrama de secuencia de la RCU de Registro de usuarios

Inicio de sesión de usuarios

Caso de uso que permite a los usuarios ingresar a la aplicación mediante su correo y contraseña guardados al registrarse. En caso de un inicio de sesión exitoso, los usuarios reciben un token (JSON web token) que es guardado en el cliente web y les permite autenticar las peticiones que hacen a la API REST.

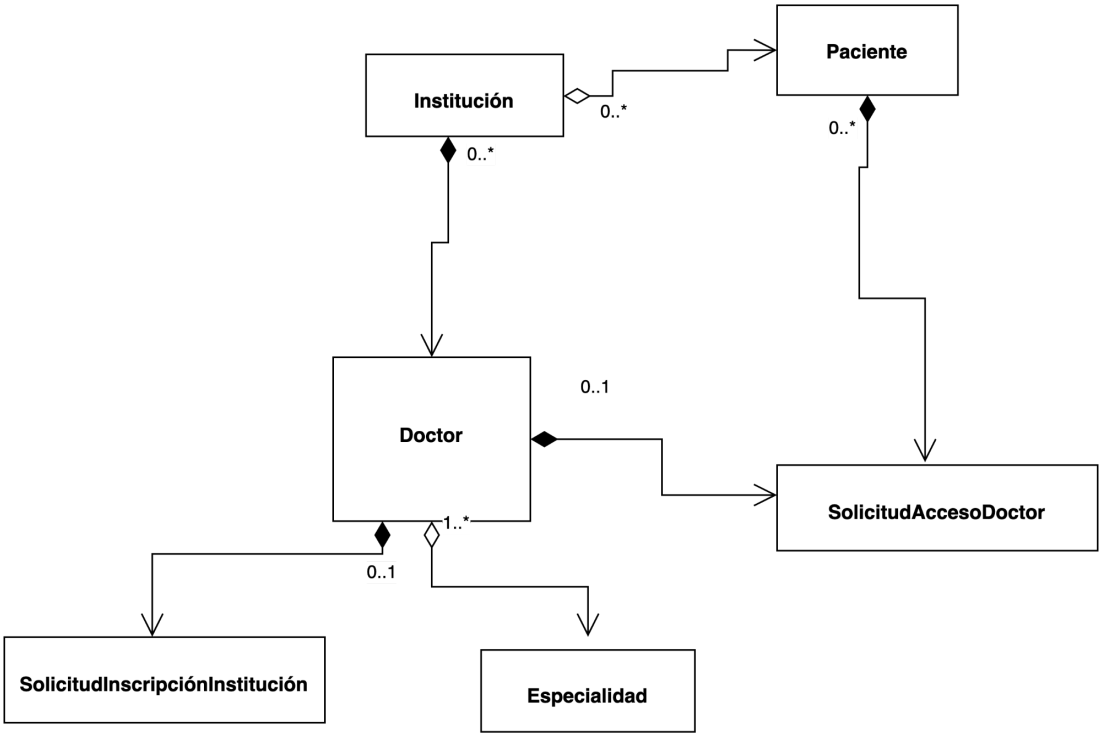


Figura 4.6: Diagrama de clases de la RCU de Inicio de sesión de usuarios

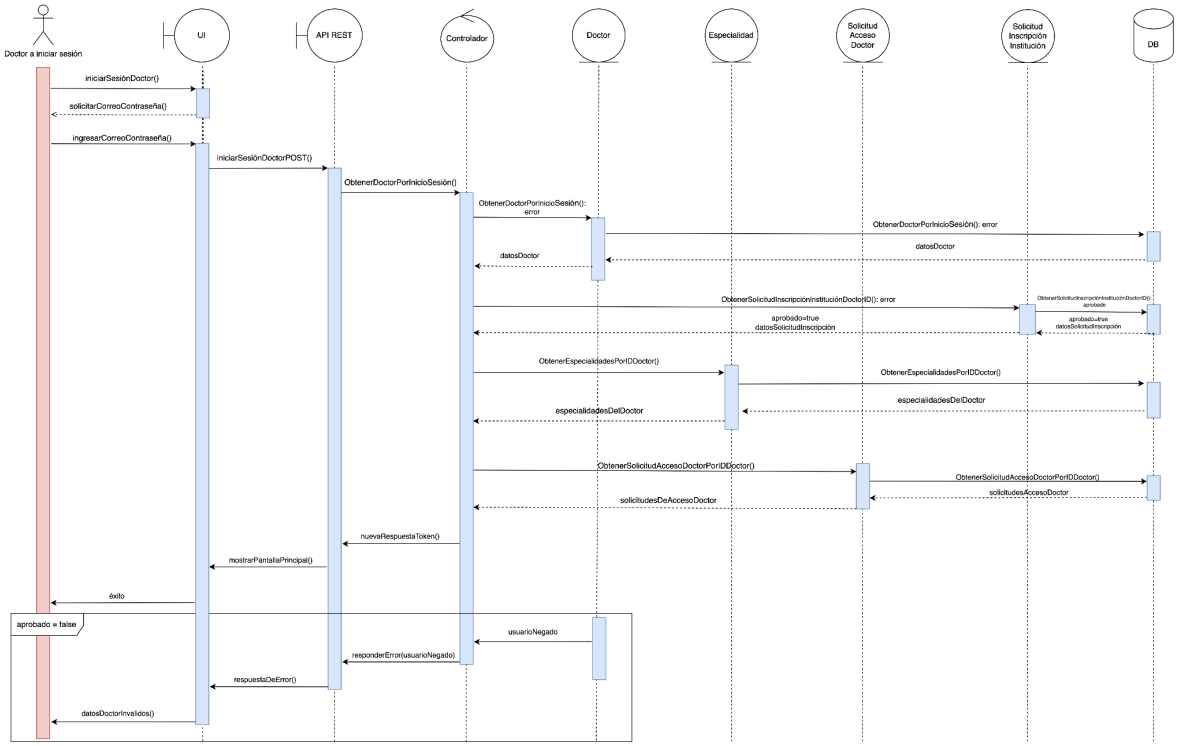


Figura 4.7: Diagrama de secuencia de la RCU de Inicio de sesión de usuarios

Filtrado y búsqueda de información

Funcionalidad que permite a los usuarios acceder y filtrar los datos a los cuales tienen acceso. En general, por limitantes de tiempo, luego de obtener los datos a través de la API REST, la interfaz gráfica se encarga de hacer el filtrado de los datos de ser necesario.

Figura 4.8: Diagrama de clases de la RCU de Filtrado y búsqueda de información

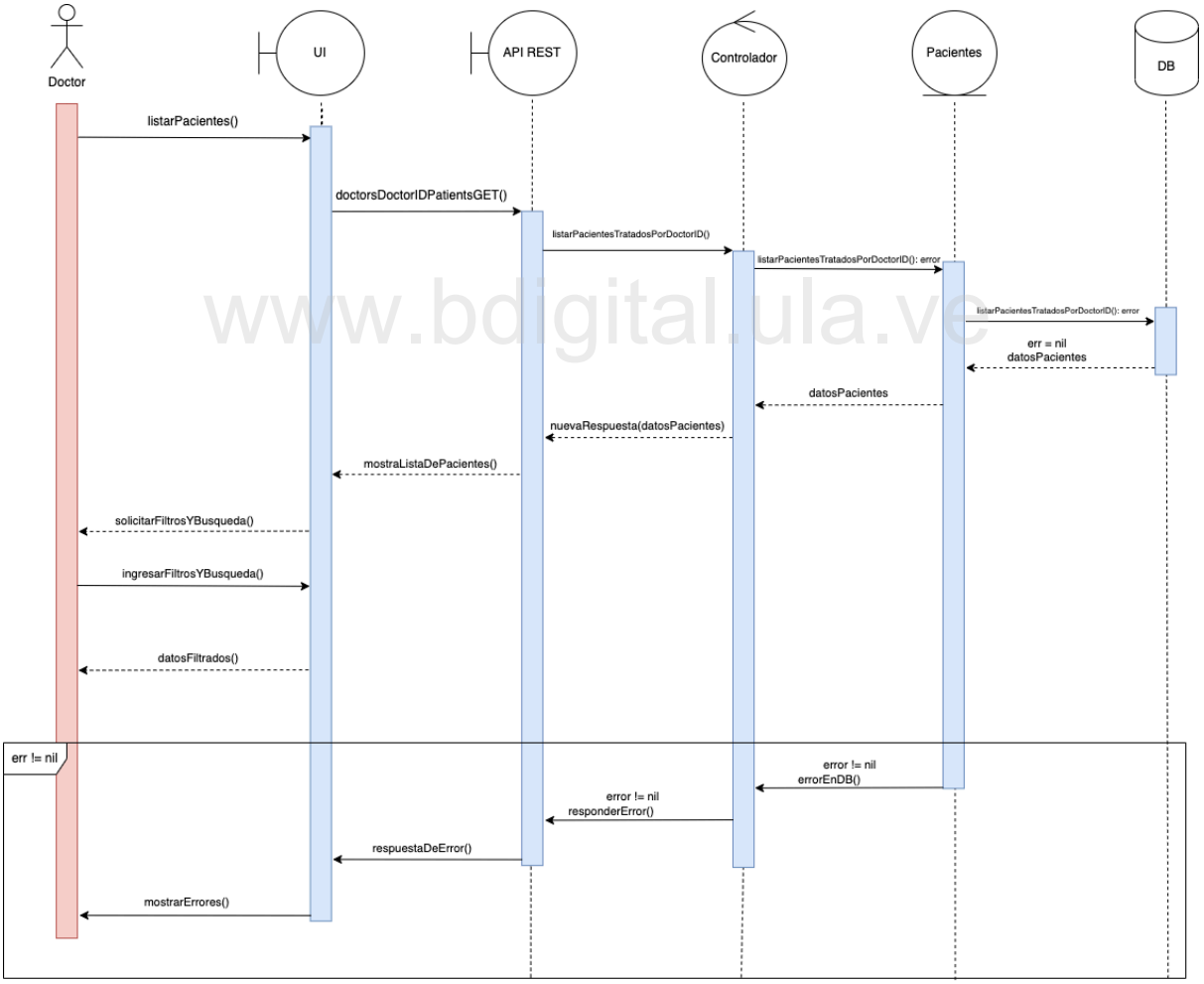


Figura 4.9: Diagrama de secuencia de la RCU de Filtrado y búsqueda de información

Edición de datos de usuario

Este caso de uso permite a los usuarios mantener sus datos al día o actualizarlos en caso de haber cometido un error al momento de registrarse. Es importante tomar en cuenta que en el caso de los médicos sólo se pueden actualizar sus datos profesionales, ya que la modificación de sus datos personales no deberían ser modificables luego de que la institución a la que forman parte los aprobó como parte de ella.

Figura 4.10: Diagrama de clases de la RCU de Edición de datos de usuario

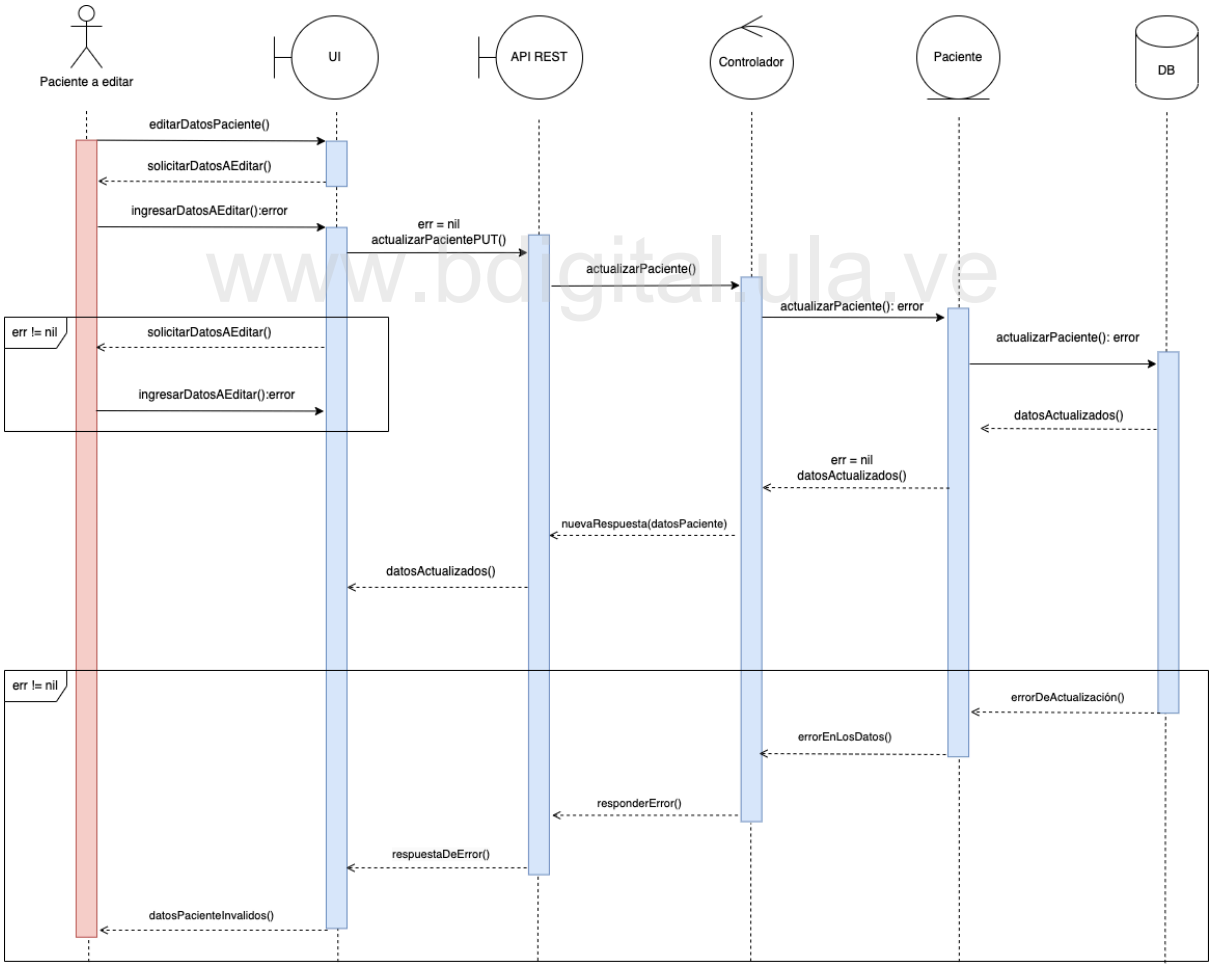


Figura 4.11: Diagrama de secuencia de la RCU de Edición de datos de usuario

Denegación de acceso a la información personal

Este proceso permite a los usuarios denegar el acceso a su información personal.

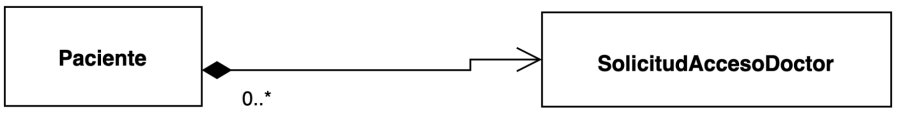


Figura 4.12: Diagrama de clases de la RCU de Denegación de acceso a la información personal

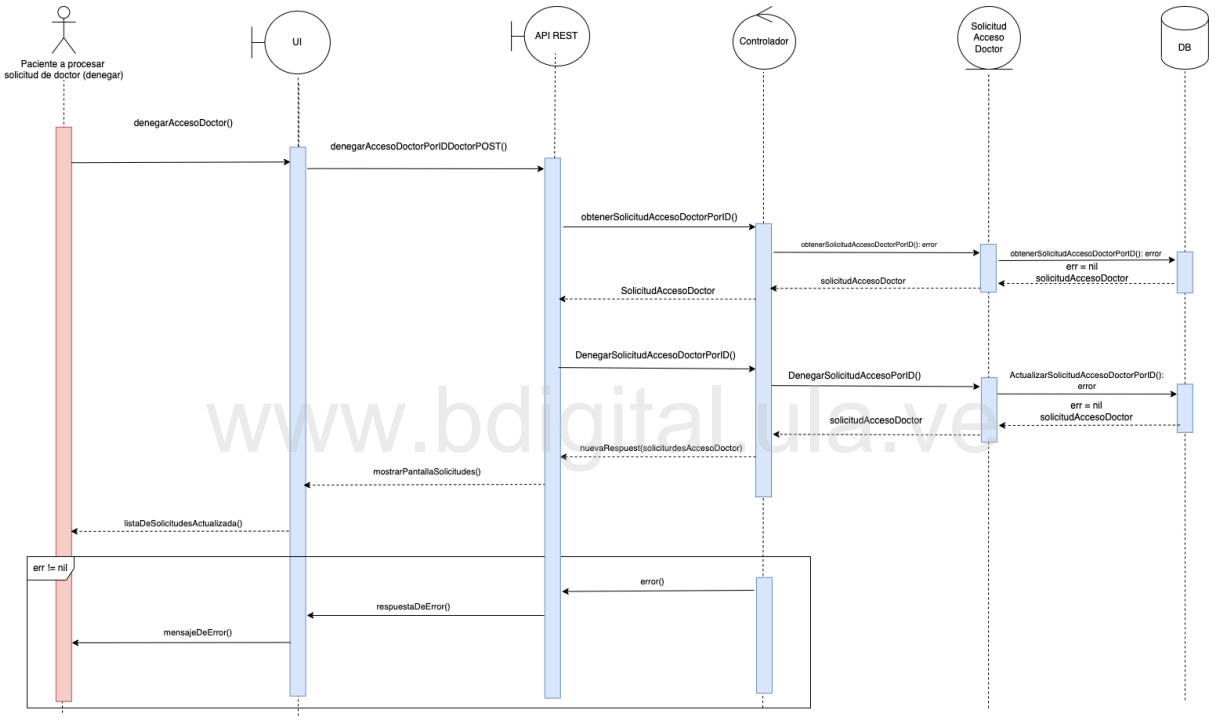


Figura 4.13: Diagrama de secuencia de la RCU de Denegación de acceso a la información personal

Concesión de acceso a la información personal

Este proceso permite a los usuarios aprobar el acceso a su información personal.

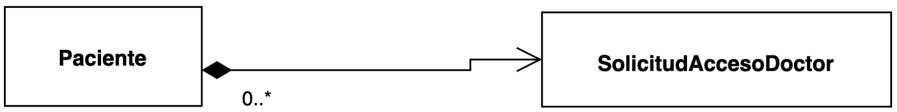


Figura 4.14: Diagrama de clases de la RCU de Concesión de acceso a la información personal

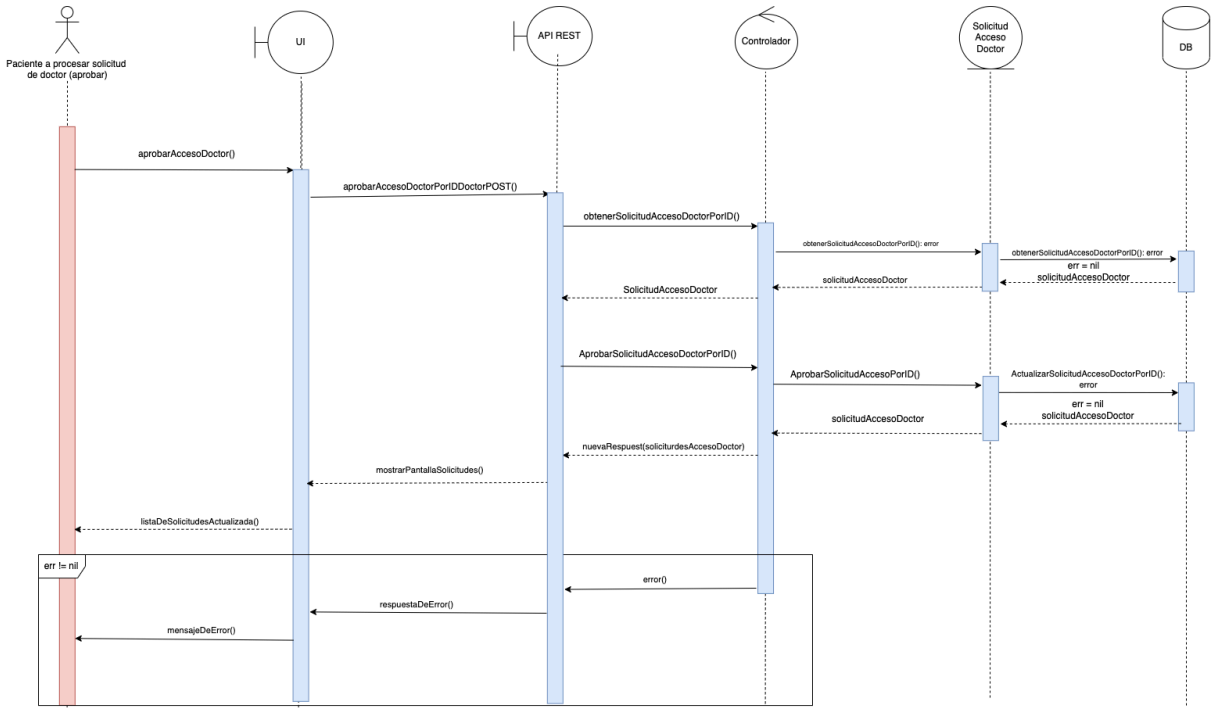


Figura 4.15: Diagrama de secuencia de la RCU de Concesión de acceso a la información personal

Revocación de acceso a la información personal

Este proceso permite a los usuarios revocar el acceso a su información personal una vez ya no es necesario que terceros tengan acceso a ella. Esto puede ocurrir específicamente en el caso de que un médico ya no atienda a un paciente.

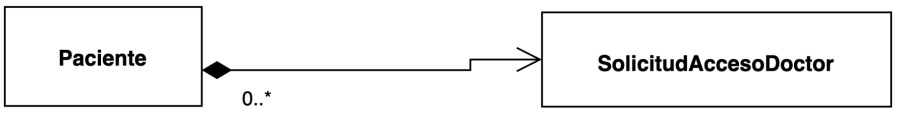


Figura 4.16: Diagrama de clases de la RCU de Revocación de acceso a la información personal

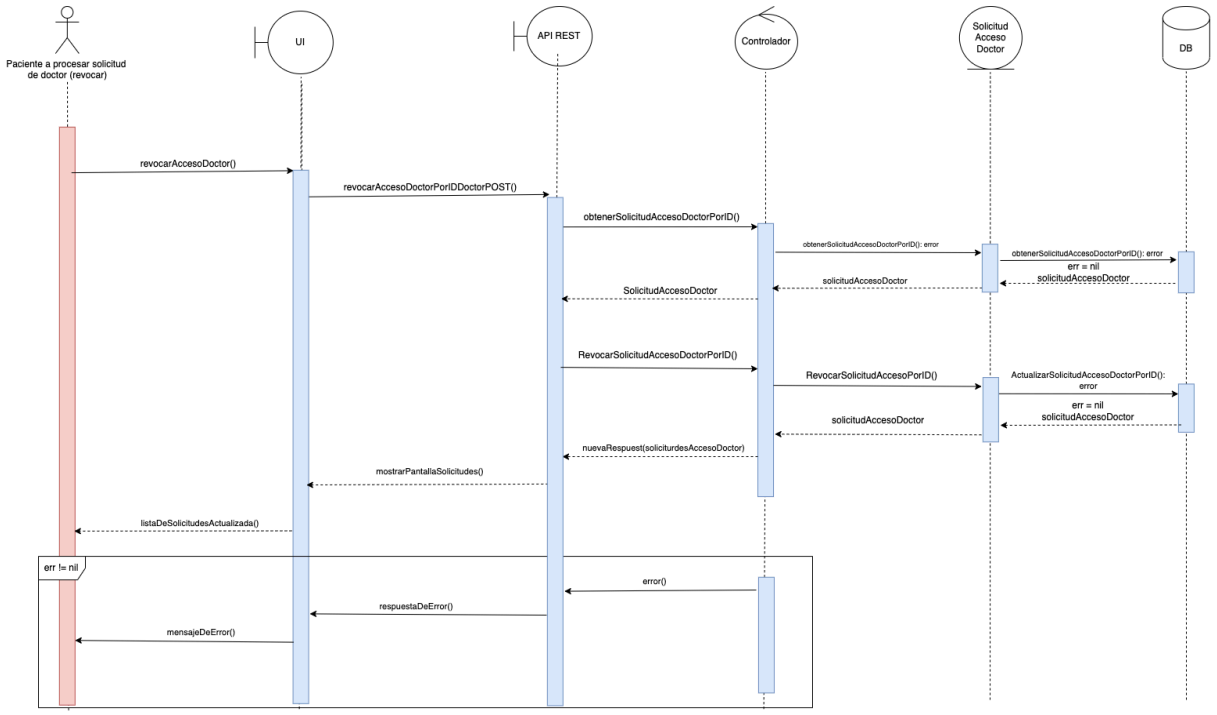


Figura 4.17: Diagrama de secuencia de la RCU de Revocación de acceso a la información personal

Acceso a información personal

Este es el proceso que permite a un usuario de la aplicación acceder a los datos personales de otro usuario del sistema a los cuales tiene acceso.

Figura 4.18: Diagrama de clases de la RCU de Acceso a información personal

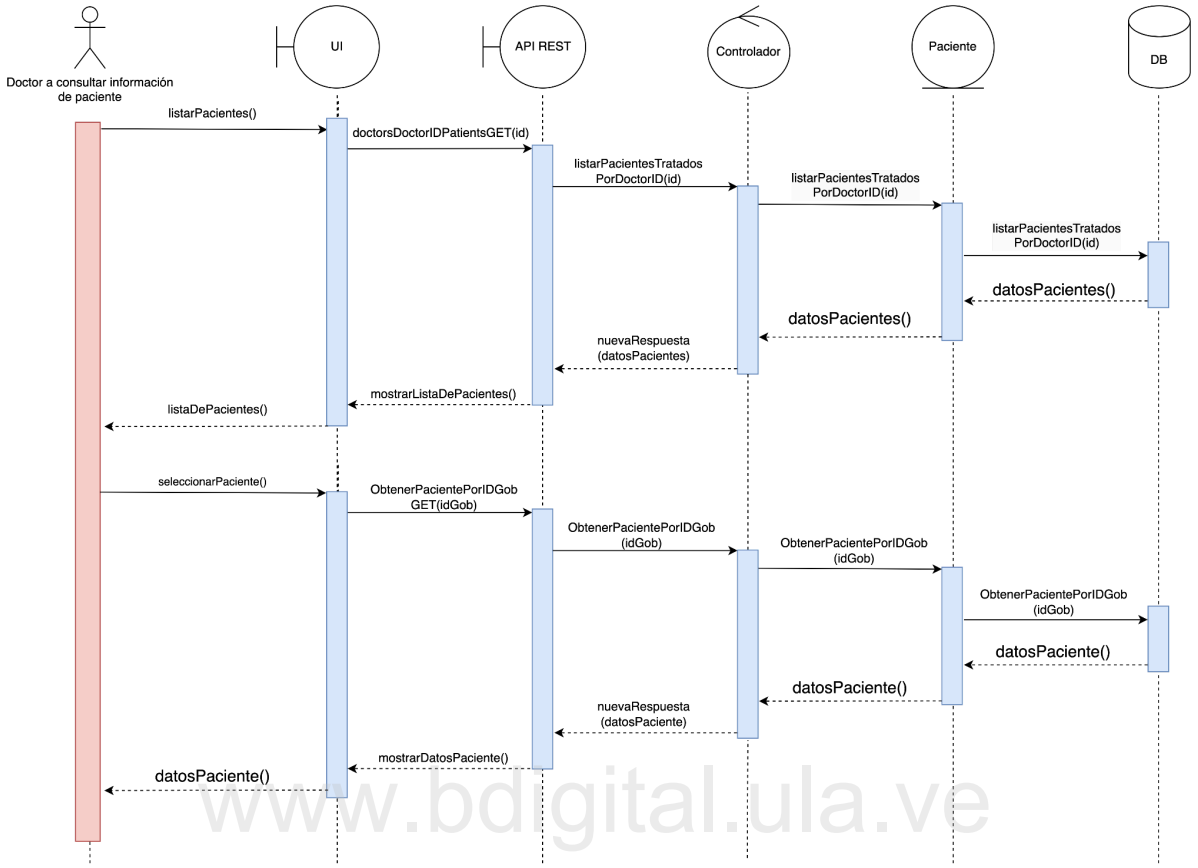


Figura 4.19: Diagrama de secuencia de la RCU de Acceso a información personal

Aprobación de pertenencia

Este proceso permite a los usuarios institucionales aprobar la pertenencia de un médico, enfermera, enfermero o institución a su red para trabajar en ella.

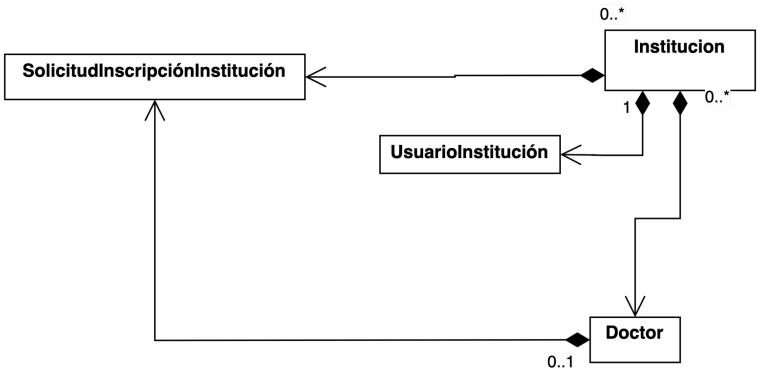


Figura 4.20: Diagrama de clases de la RCU de Aprobación de pertenencia

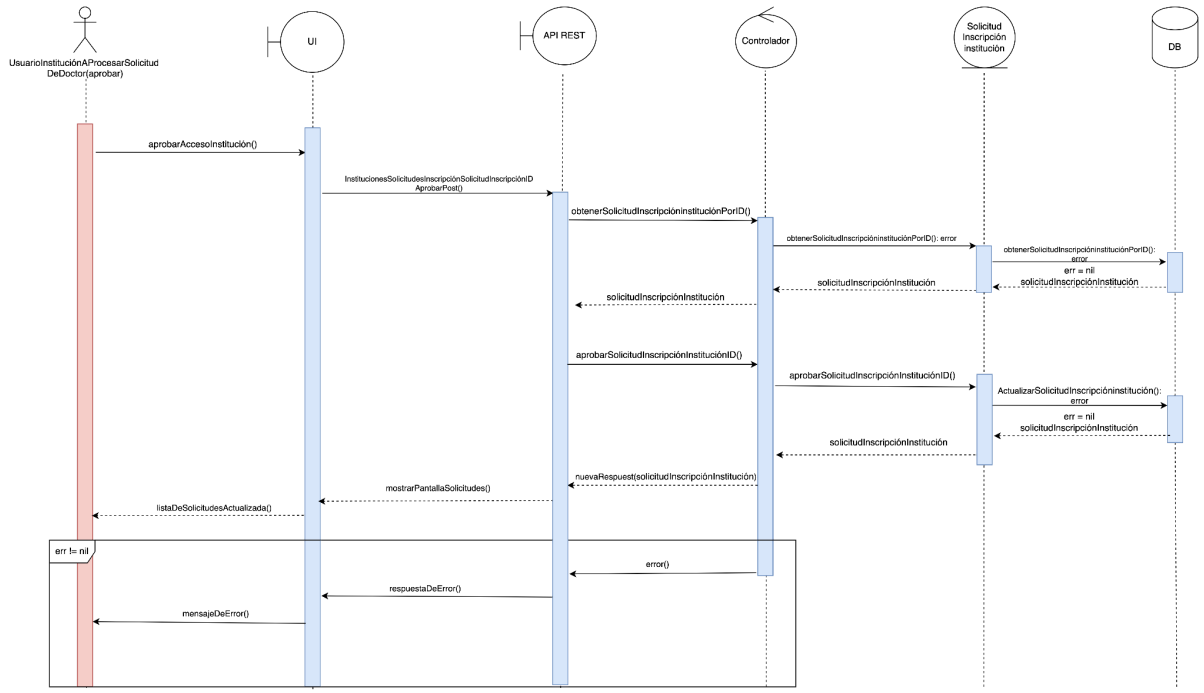


Figura 4.21: Diagrama de secuencia de la RCU de Aprobación de pertenencia

Denegación de pertenencia

Este proceso permite a los usuarios institucionales denegar la pertenencia de un médico, enfermera, enfermero o institución a su red para trabajar en ella.

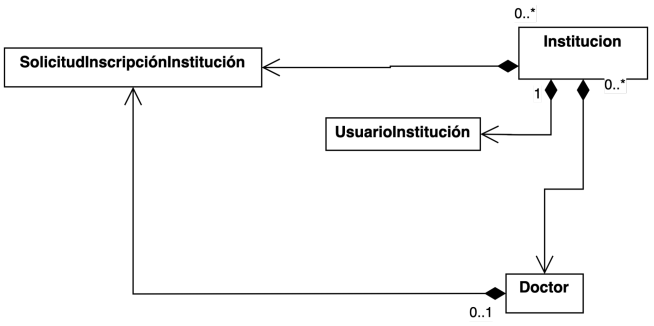


Figura 4.22: Diagrama de clases de la RCU de Denegación de pertenencia

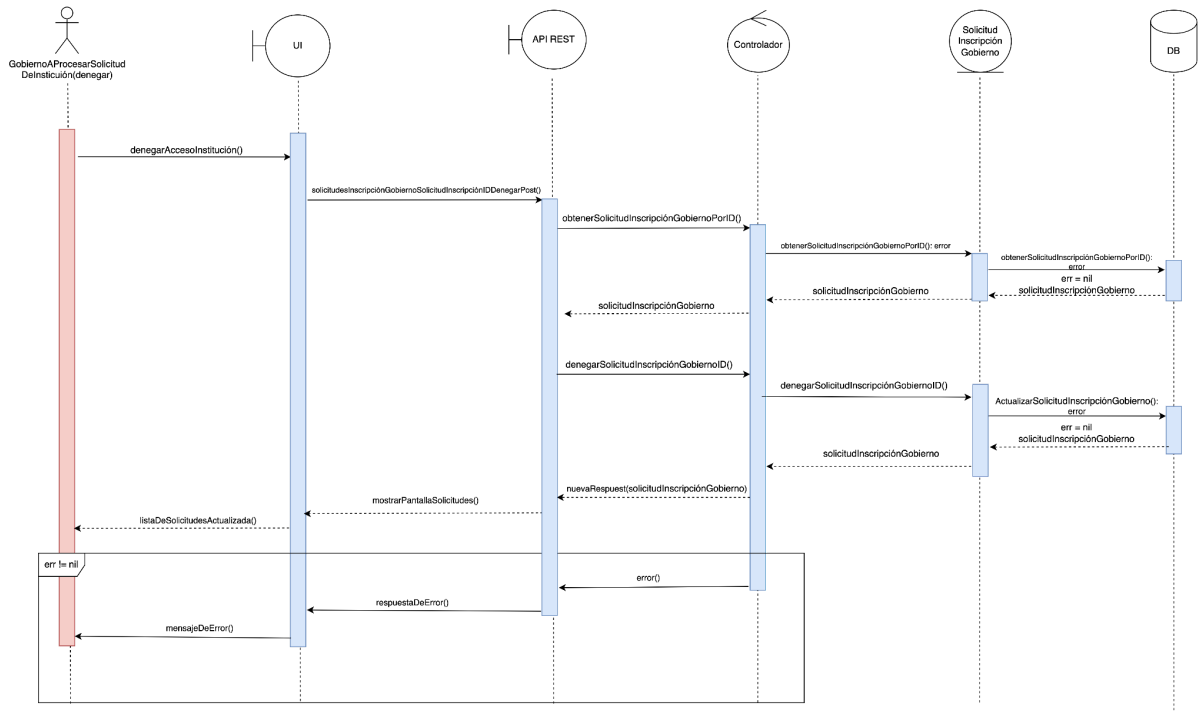


Figura 4.23: Diagrama de secuencia de la RCU de Denegación de pertenencia

Revocación de pertenencia

Este proceso permite a los usuarios institucionales revocar la pertenencia de un médico, enfermera, enfermero o institución a su red.

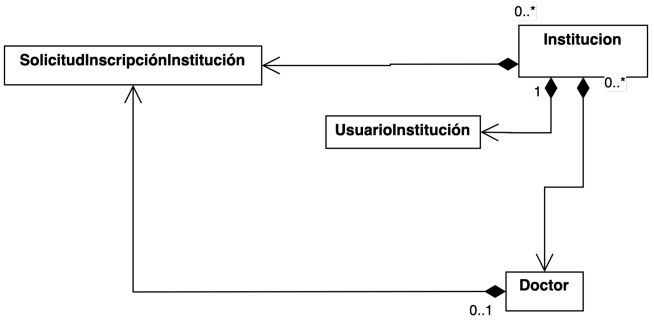


Figura 4.24: Diagrama de clases de la RCU de Revocación de pertenencia

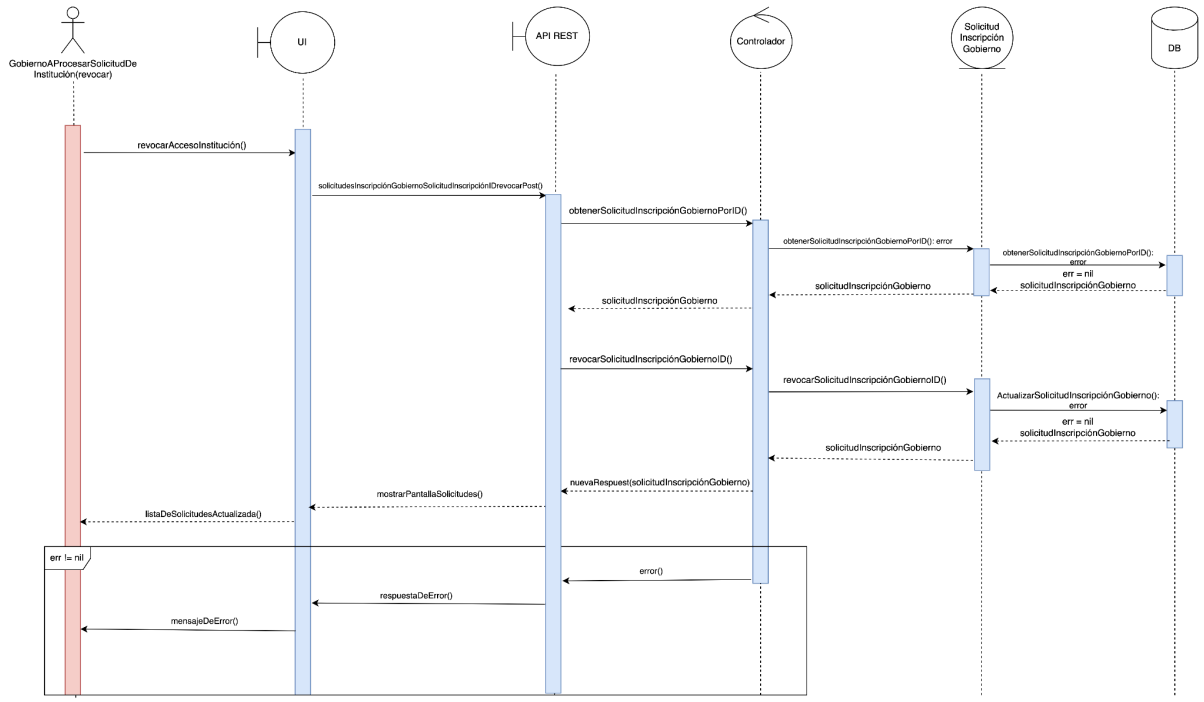


Figura 4.25: Diagrama de secuencia de la RCU de Revocación de pertenencia

Capítulo 5
Diseño de la aplicación

En este capítulo, nos enfocaremos en el diseño de la aplicación para el Sistema de Gestión de Identidad de Pacientes en Instituciones de Salud mediante el uso de tecnología de cadena de bloques, la cual, a su vez, está interrelacionada con el Sistema de Gestión de Historias Clínicas en Instituciones de Salud mediante el uso de tecnología de cadena de bloques planteado en el proyecto de grado complementario al presente. El propósito principal del capítulo es establecer la arquitectura de la aplicación y especificar sus componentes a un alto nivel de abstracción.

La gestión de identidad de pacientes es una tarea crítica en cualquier institución de salud, ya que garantiza la precisión y la seguridad de los datos médicos. La integración de la tecnología de cadena de bloques promete mejorar significativamente la seguridad, la transparencia y la eficiencia en la gestión de estos datos. En este contexto, diseñar una aplicación robusta y eficiente es crucial para asegurar que se cumplan estos objetivos.

Primero, se presentará una visión general de la arquitectura propuesta, describiendo cómo los diferentes componentes del sistema interactúan entre sí y con la cadena de bloques. Luego, se detallarán los principales componentes del sistema, incluyendo la interfaz de usuario, la lógica de negocios, y la capa de datos. Cada uno de estos componentes será abordado desde una perspectiva de alto nivel, proporcionando una visión clara de su función y sus interacciones dentro del sistema global.

A lo largo del capítulo, se destacarán los principios de diseño que guiarán la construcción del sistema, asegurando que la solución sea escalable, segura y fácil de

mantener. Además, se explicarán las decisiones técnicas clave y las justificaciones detrás de ellas, fundamentadas en las necesidades específicas del entorno de salud y las ventajas inherentes de la tecnología de cadena de bloques.

Con esta base, el diseño detallado permitirá una implementación efectiva del sistema, sentando las bases para una gestión de identidad de pacientes interinstitucional.

5.1. Refinamiento de requisitos arquitectónicos

5.1.1. Selección de requisitos que influyen en la arquitectura de la aplicación

Los requisitos que influyen en la arquitectura de la solución propuesta a la problemática de los usuarios potenciales del sistema planteado en este proyecto son los siguientes:

- **Registro Seguro:** La seguridad de los datos del producto mínimo viable se garantiza con el encriptamiento de los datos sensibles en la base de datos, así como con el control de acceso granular a la base de datos, restringiendo los roles disponibles para los usuarios de la base de datos.

- **Identidad Única:** Este requisito se cumple generando una llave privada y dirección única en la cadena de bloques para cada paciente que se registra en el sistema, identificándose inequívocamente dentro de la aplicación.

- **Autenticación Robusta:** La autenticación para el acceso a los datos implementada se considera robusta ya que las peticiones a la API REST se autentican con un token el cual se puede generar sólo con acceso a la clave generada por el paciente al registrarse, la cual está encriptada en la base de datos.

- **Control de Acceso Granular:** El acceso granular a los datos de los pacientes se asegura mediante tablas en la base de datos que contienen las tuplas de accesos

permitidos por los pacientes para los doctores. De igual forma, la aplicación cuenta con una jerarquía de permisos que aseguran que los médicos e instituciones están autorizados para formar parte de ella. Primero, se tiene un usuario del gobierno que se tiene que instanciar manualmente ya que no cuenta con un registro disponible al público en general para asegurar que sólo haya uno por ente gubernamental. Luego, las instituciones que quieran formar parte de la red de dicho ente gubernamental, tienen que ser verificadas y aprobadas por el usuario del gobierno antes de ser parte del sistema. Por último, los médicos tienen que ser aprobados por el usuario administrador de la institución para poder formar parte de la aplicación. Esta serie de verificaciones le proveen a la aplicación la misma jerarquía de dominios que posee el sistema de salud nacional.

- **Gestión de Revocación de Consentimiento:** La aplicación facilita la revocación del consentimiento al uso de los datos de los pacientes mediante la interfaz gráfica utilizando los métodos proveídos por la API REST.

- **Arquitectura Escalable:** La naturaleza distribuida de la cadena de bloques nos brinda la posibilidad de escalar según las necesidades del sistema. De igual manera, la forma en que está contenida la capa de negocios y la de datos permite su orquestación con herramientas como Kubernetes.

5.1.2. Especificación de requisitos arquitectónicos no funcionales

- **Seguridad**
    - **Confidencialidad de los datos:** Los datos del sistema deben protegerse del acceso no autorizado.
    - **Integridad de los datos:** El sistema debe asegurarse de que los datos no sean modificados o alterados sin autorización.
    - **Disponibilidad de los datos:** Los datos del sistema deben estar disponibles para los usuarios autorizados cuando los necesiten.

- **Autenticación:** El sistema debe verificar la identidad de los usuarios antes de otorgarles acceso.
- **Autorización:** El sistema debe controlar el acceso a los recursos y funciones en función de los roles y permisos de los usuarios.

- **Escalabilidad**

  - **Escalabilidad de usuarios y datos:** El sistema debe poder manejar un mayor número de usuarios y datos sin afectar significativamente el rendimiento o la disponibilidad.

- **Confiabilidad**

  - **Tolerancia a fallos:** El sistema debe ser capaz de continuar funcionando incluso en caso de fallas de software.
  - **Capacidad de recuperación:** El sistema debe poder recuperarse rápidamente de fallas y restaurar el servicio con un mínimo de pérdida de datos.
  - **Previsibilidad:** El sistema debe comportarse de manera predecible y consistente bajo condiciones normales y de carga.

- **Usabilidad**

  - **Facilidad de uso:** El sistema debe ser fácil de usar y aprender para los usuarios objetivo.
  - **Facilidad de aprendizaje:** Los usuarios deben poder aprender a usar el sistema rápidamente y sin necesidad de una capacitación extensa.

5.1.3. Identificación de los mecanismos de diseño e implementación de los requisitos arquitectónicos no funcionales

- **Confidencialidad de los datos:** Los datos de los pacientes, al ser altamente sensibles, deben almacenarse de forma segura en la cadena de bloques, aprovechando las características de inmutabilidad y resistencia a la manipulación que ofrece esta tecnología. Se deben emplear mecanismos de cifrado y control de acceso estrictos para proteger los datos contra accesos no autorizados, utilizando la identidad única del paciente relacionada a la *blockchain*.
- **Integridad de los datos:** La inmutabilidad de la cadena de bloques garantiza que el historial de los pacientes sea inalterable, asegurando la integridad de la información. Esto permite la trazabilidad de las acciones y decisiones médicas, previniendo modificaciones no autorizadas y mejorando la transparencia en la atención médica.
- **Disponibilidad de los datos:** La arquitectura debe garantizar la disponibilidad del sistema para que los profesionales de la salud puedan acceder a los datos de los pacientes de manera confiable, incluso en caso de fallas o interrupciones del servicio. Esto puede lograrse mediante la implementación de nodos redundantes en la red *blockchain* y mecanismos de recuperación ante desastres.
- **Autenticación:** El sistema debe implementar mecanismos de autenticación robustos para verificar la identidad de los usuarios, como la verificación por parte del ministerio a las instituciones de salud y de las instituciones a los médicos que pertenecen a ellas. Esto es crucial para garantizar que solo el personal médico autorizado pueda acceder a los datos sensibles de los pacientes.
- **Autorización:** La arquitectura debe permitir un control de acceso granular sobre los datos de los pacientes, en función de los roles y responsabilidades de cada usuario. Esto implica definir diferentes niveles de acceso y permisos para médicos, enfermeras, personal administrativo y otros actores relevantes del sistema de salud.
- **Escalabilidad de usuarios y datos:** La aplicación debe diseñarse con una arquitectura modular, utilizando microservicios o componentes independientes, para facilitar su escalabilidad horizontal. Esto permite agregar más nodos a la red *blockchain* y aumentar la capacidad de procesamiento a medida que crece la demanda del sistema.
- **Tolerancia a fallos:** La arquitectura debe ser tolerante a fallos, utilizando mecanismos de redundancia y replicación de datos en la cadena de bloques. Esto

asegura que el sistema continúe funcionando incluso si algunos nodos de la red fallan o experimentan interrupciones.

- **Capacidad de recuperación:** Se deben implementar mecanismos de recuperación robustos para restaurar el sistema rápidamente en caso de fallas o desastres. Esto puede incluir copias de seguridad regulares de los datos y planes de recuperación ante desastres bien definidos.
- **Previsibilidad:** Asegurar el comportamiento del sistema de manera predecible y consistente bajo condiciones normales y de carga implica realizar pruebas exhaustivas y monitorear el rendimiento del sistema de manera proactiva.
- **Facilidad de uso y de aprendizaje:** El sistema debe contar con una interfaz de usuario intuitiva, fácil de usar y accesible desde dispositivos móviles y de escritorio, tanto para profesionales médicos como para pacientes. Esto implica diseñar una aplicación multiplataforma con una interfaz gráfica amigable, con menús claros, íconos y flujos de trabajo intuitivos.

5.2. Diseño de la arquitectura

5.2.1. Análisis de la lista de requisitos arquitectónicos

La arquitectura del sistema debe estar cuidadosamente diseñada para cumplir con los estrictos requisitos de seguridad, escalabilidad, confiabilidad y usabilidad del sistema. En cuanto al cumplimiento de dichos requisitos, la tecnología *blockchain* ofrece características inherentes que la convierten en una opción ideal para la gestión segura de datos de pacientes, ya que los datos de los pacientes se almacenan en un registro distribuido e inmutable, accesible solo para usuarios autorizados mediante criptografía de su clave pública y cada bloque en la cadena de bloques contiene un hash del bloque anterior, creando una cadena auditable e inviolable que garantiza la integridad de los datos, además que la naturaleza descentralizada de la cadena de bloques distribuye los datos entre múltiples nodos, lo que garantiza la disponibilidad incluso en caso de fallos en nodos individuales.

Para lograr una gestión eficiente y escalable de la identidad de los pacientes, se propone una arquitectura modular compuesta por:

- ***Blockchain*:** Con la cual se genera la identidad única e inmutable del paciente utilizando la dirección y la clave privada generadas para cada uno. El tipo de *blockchain* que utilizamos es de tipo comunitaria o de consorcio, ya que por la naturaleza del proyecto no hace falta que terceros a la aplicación tengan acceso libre a los datos en ella, pero sí un acceso regulado dependiendo de su lugar de operación.
- **Base de datos auxiliar:** La cual almacena metadatos e índices de la información de identidad del paciente en la cadena de bloques, optimizando el rendimiento y la accesibilidad.
- **API REST:** Sirve como punto de entrada único para todas las solicitudes de la aplicación, proporcionando enrutamiento, autenticación y autorización a interfaces para interactuar con los componentes del sistema, permitiendo la integración con aplicaciones externas y dispositivos móviles.
- **Controlador principal:** Gestiona la lógica de negocio de la aplicación, incluyendo la autenticación, autorización, registro de pacientes y administración de accesos.

Se plantea que la API REST y el controlador principal formen parte de un servicio, al igual que la base de datos auxiliar y la *blockchain* estén en servicios independientes y escalables, facilitando el desarrollo, mantenimiento y despliegue de la aplicación. Teniendo los componentes necesarios en microservicios aislados, estos se pueden almacenar en contenedores dockerizados, los cuales promueven la portabilidad, permitiendo una implementación consistente en diferentes entornos.

La arquitectura propuesta aprovecha las capacidades de escalabilidad horizontal de la tecnología Blockchain y los microservicios para agregar más nodos a la red *blockchain* o instancias de microservicios para manejar un mayor volumen de usuarios y transacciones, así como garantiza la continuidad del servicio incluso en caso de fallos en nodos o microservicios individuales, gracias a la redundancia y replicación de datos en la cadena de bloques y la implementación de mecanismos de alta disponibilidad para los microservicios.

Por último, para garantizar una experiencia de usuario fluida y accesible, se propone un cliente basado en React y Next JS, tecnologías front-end modernas que permiten desarrollar interfaces de usuario dinámicas y receptivas, junto con Electron y Capacitor, herramientas para empaquetar aplicaciones web en aplicaciones móviles nativas, proporcionando acceso desde dispositivos móviles y de escritorio.

5.2.2. Capas de la aplicación

La arquitectura de la aplicación se estructuró en tres capas:

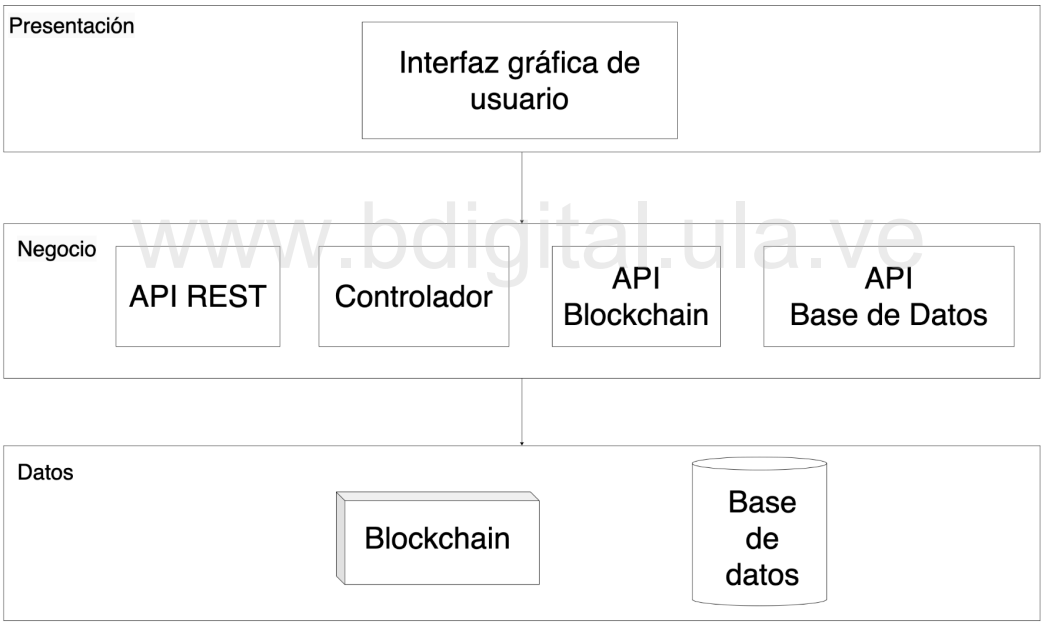


Figura 5.1: Diagrama arquitectónico de la aplicación

1. **Capa de Presentación:**

- **Responsabilidades:**

    - Interactuar con el usuario a través de la interfaz gráfica.
    - Presentar la información de manera clara y organizada.
    - Capturar y validar datos ingresados por el usuario.

- **Componentes:**

    - Interfaz de usuario web, móvil y de escritorio desarrollada con React, Next.js, Electron y Capacitor.

2. **Capa de Negocio:**

- **Responsabilidades:**

    - Implementar la lógica principal del sistema, incluyendo la gestión de identidades, autenticación, autorización y acceso a datos de pacientes.
    - Procesar las solicitudes recibidas desde la capa de presentación y comunicarse con la capa de datos.
    - Aplicar las reglas de negocio y validaciones necesarias para garantizar la integridad y seguridad de los datos.

- **Componentes:**

    - API REST que expone servicios RESTful y recibe las solicitudes del cliente para la interacción con la capa de negocio.
    - Controlador principal que coordina la interacción entre las solicitudes del cliente y la capa de datos mediante las API de la *blockchain* y la API de base de datos.
    - API de la *blockchain*, la cual expone un conjunto de métodos para crear billeteras e insertar datos en la cadena de bloques.
    - API de base de datos, que se encarga de insertar, obtener, modificar y borrar datos de la base de datos auxiliar, utilizando los modelos definidos para cada clase del negocio.

3. **Capa de Datos:**

- **Responsabilidades:**

- Almacenar y gestionar de forma segura la información de identidad de los pacientes.
- Proporcionar acceso a los datos de forma eficiente y confiable a la capa de negocio.
- Garantizar la integridad, consistencia y disponibilidad de los datos.

- **Componentes:**

  - Base de datos distribuida basada en tecnología *blockchain*, usando Hyper Ledger Fabric.
  - Base de datos auxiliar PostgreSQL para almacenar metadatos e índices de la información en la cadena de bloques.

La interacción entre estas capas se da de la siguiente manera:

1. El usuario interactúa con la capa de presentación a través de la interfaz gráfica.
2. La capa de presentación envía las solicitudes del usuario a la API REST.
3. La API REST valida la solicitud y la envía al controlador correspondiente en la capa de negocio.
4. El controlador procesa la solicitud, aplicando la lógica de negocio y accediendo a los datos en la capa de datos por medio de la API de la *blockchain* o la de la base de datos.
5. La capa de datos recupera la información solicitada de la base de datos *blockchain* o auxiliar.
6. Los llamados realizados hasta ahora empiezan a responder hasta llegar a la API REST, desde donde se envía la información procesada de vuelta a la capa de presentación.
7. La capa de presentación presenta la información al usuario en un formato adecuado.

Esta arquitectura en capas proporciona una estructura sólida y flexible para la aplicación de gestión de identidad de pacientes en *blockchain*, permitiendo un desarrollo

eficiente, un mantenimiento simplificado y una escalabilidad adecuada para satisfacer las demandas actuales del sistema.

5.3. Diseño de la interfaz gráfica de usuario

Luego de un proceso meticuloso de refinamiento y análisis de los requisitos relacionados con la interfaz gráfica de usuario (IGU), se procedió a analizar las funciones de los usuarios a través de casos de uso e historias de usuario. Esto permitió identificar con precisión las funcionalidades que la aplicación debía ofrecer a cada tipo de usuario. Con base en este análisis detallado, se diseñó la estructura jerárquica general de la IGU de toda la aplicación, así como la estructura general de las páginas web que la componen.

La estructura jerárquica de la IGU se define como la organización lógica de los elementos de la interfaz, estableciendo relaciones de importancia y prioridad entre ellos. Esta organización se refleja en la disposición visual de los elementos y en la forma en que los usuarios interactúan con ellos. Una estructura jerárquica bien diseñada guía al usuario de forma intuitiva a través de la aplicación, permitiéndole encontrar rápidamente las funciones que necesita y realizar las tareas deseadas con facilidad.

El análisis de las funciones de los usuarios fue fundamental para identificar las necesidades específicas de cada tipo de usuario. A través de casos de uso e historias de usuario, se documentaron las tareas que cada usuario debía realizar dentro de la aplicación. Esto permitió comprender los flujos de trabajo y las acciones que los usuarios debían ejecutar para alcanzar sus objetivos.

La combinación del análisis de requisitos, el análisis de funciones de usuario y la definición de la estructura jerárquica sentó las bases para el diseño de una IGU efectiva y centrada en el usuario. La estructura resultante asegura que la aplicación sea fácil de usar, intuitiva y satisfaga las necesidades de todos los tipos de usuarios.

5.3.1. Estructura jerárquica general de la interfaz gráfica de usuario de la aplicación

La identificación de las funciones que la aplicación ofrecería a cada tipo de usuario se realizó mediante la elaboración de la arquitectura de la información para cada uno, en la cual se usó la siguiente representación:

Figura 5.2: Notación de la arquitectura de la información de la interfaz gráfica de usuario de la aplicación

Médicos

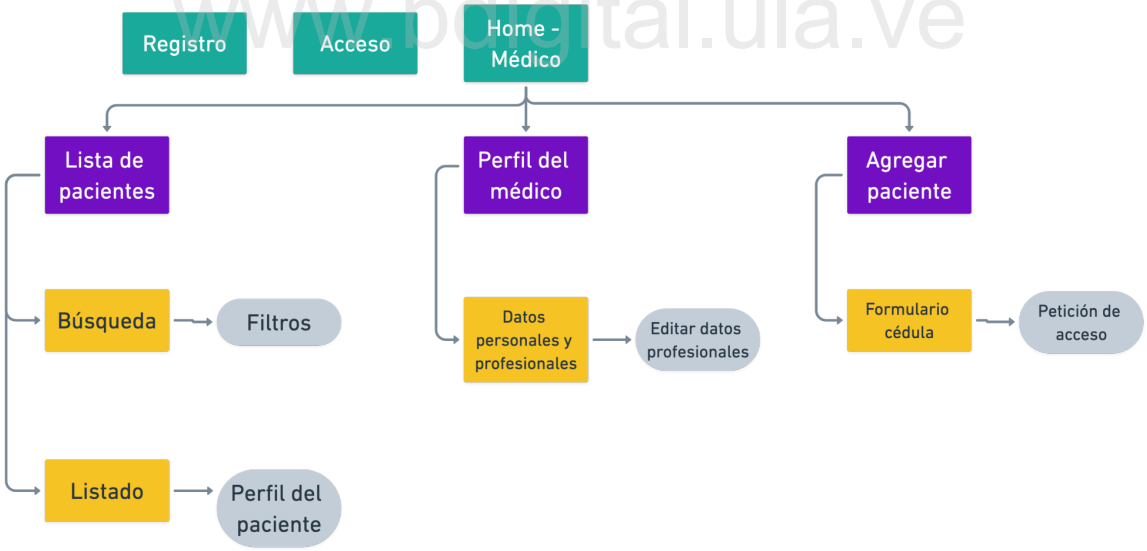


Figura 5.3: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para médicos

Enfermeros y enfermeras

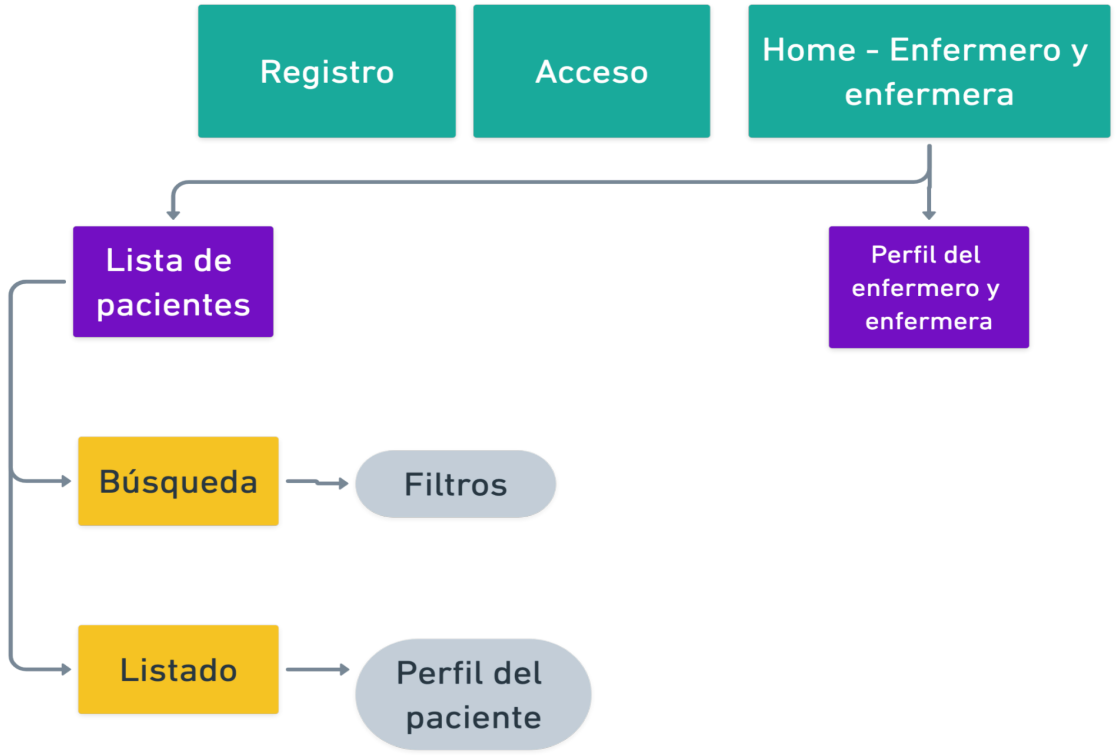


Figura 5.4: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para enfermeros y enfermeras

Pacientes

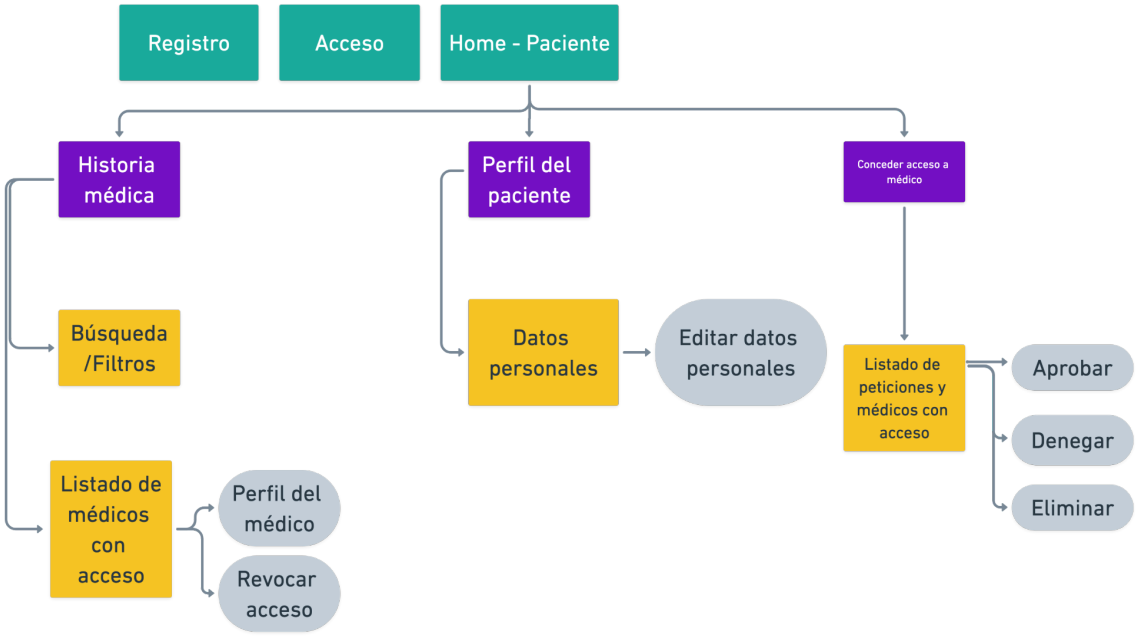


Figura 5.5: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para pacientes

Institución

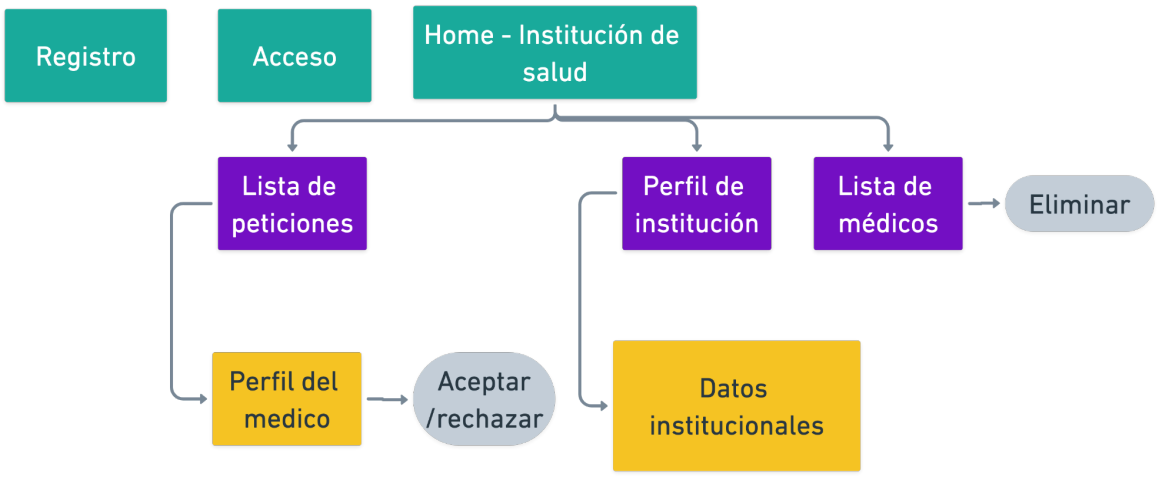


Figura 5.6: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para usuarios de instituciones

Ente gubernamental

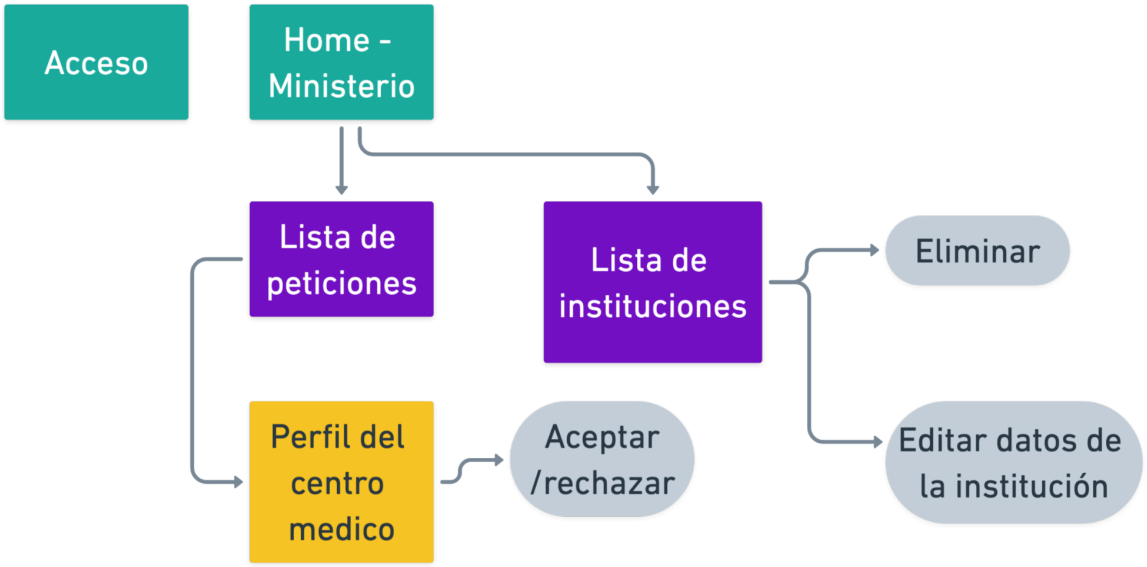


Figura 5.7: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para usuarios del ente gubernamental

Para la estructura jerárquica general de la interfaz gráfica de usuario de la aplicación se utilizaron los siguientes elementos para su descripción:

Figura 5.8: Notación de la estructura jerárquica de la interfaz gráfica de usuario de la aplicación

Médicos

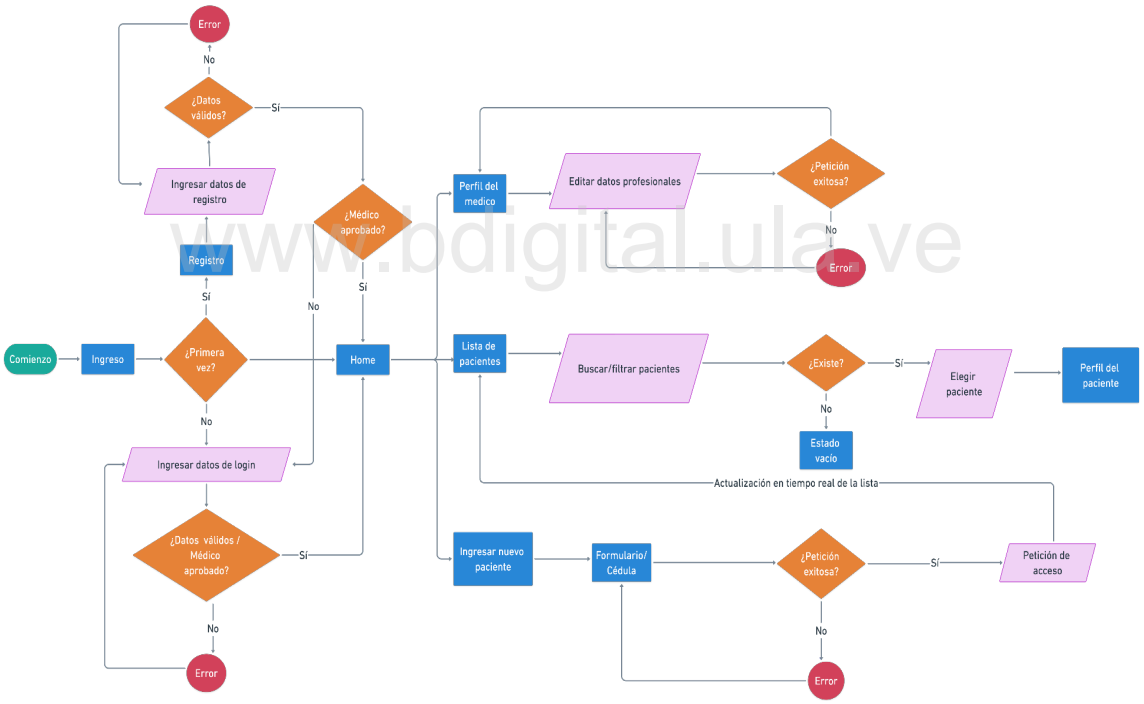


Figura 5.9: Estructura jerárquica de la interfaz gráfica de usuario para médicos

Enfermeros y enfermeras

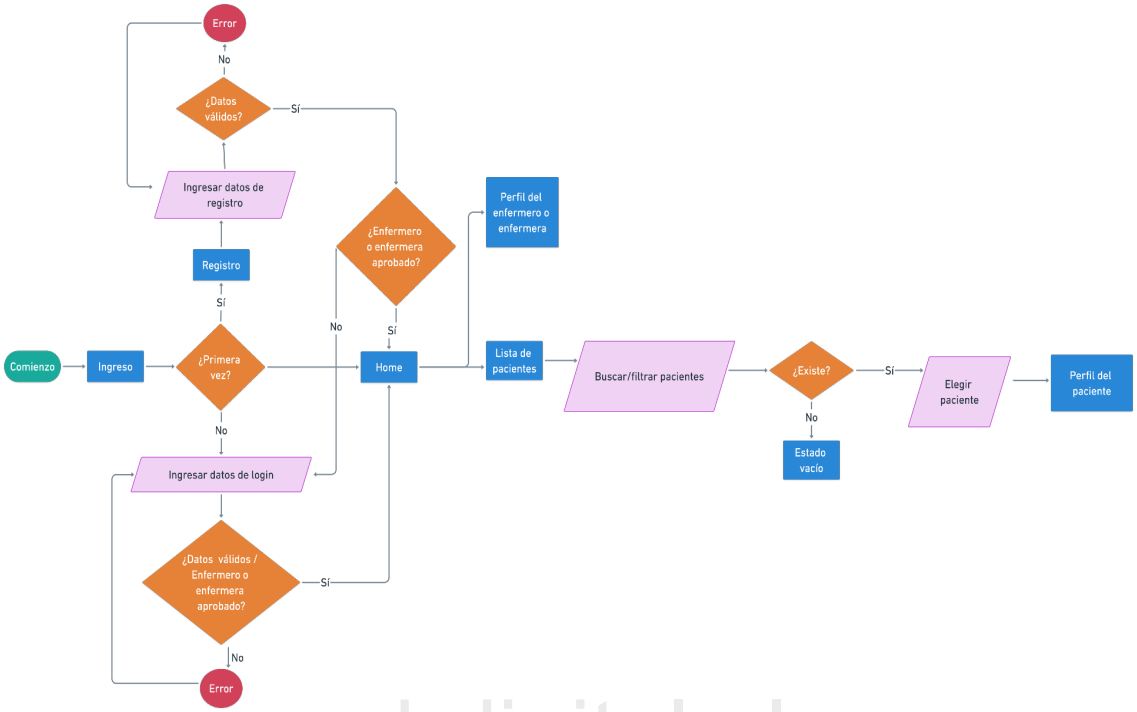


Figura 5.10: Estructura jerárquica de la interfaz gráfica de usuario para enfermeros y enfermeras

Pacientes

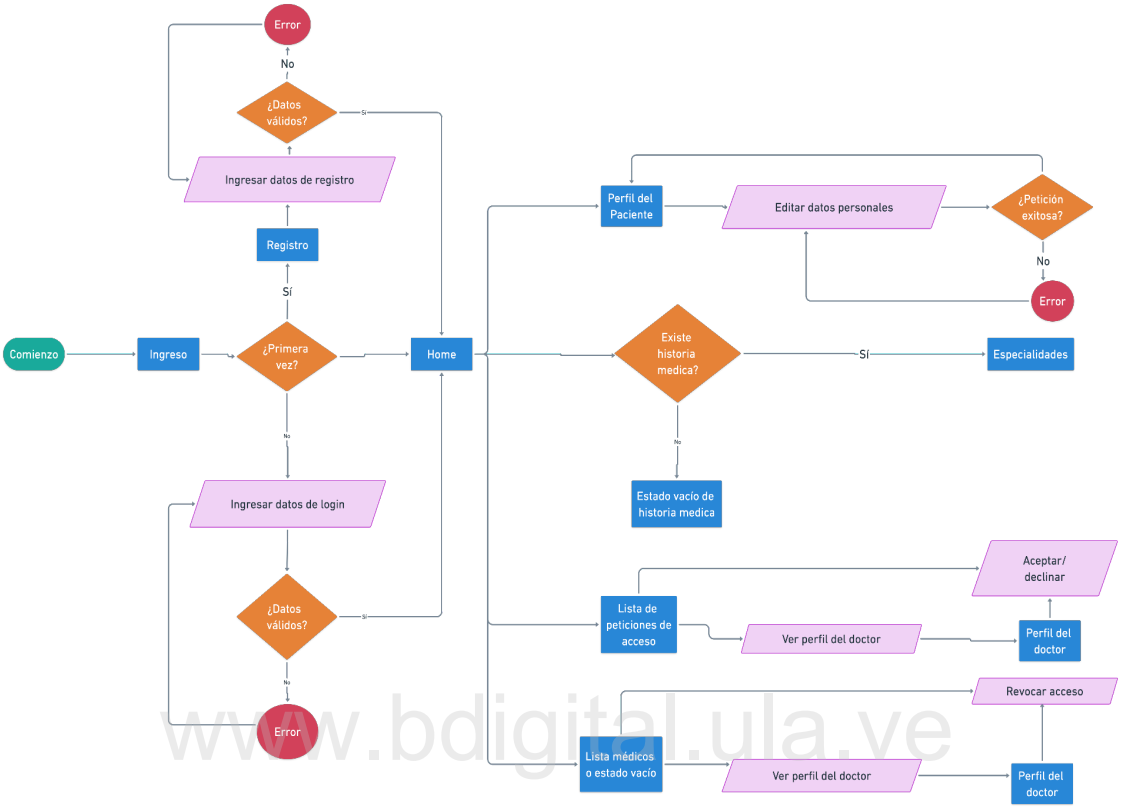


Figura 5.11: Estructura jerárquica de la interfaz gráfica de usuario para pacientes

Institución

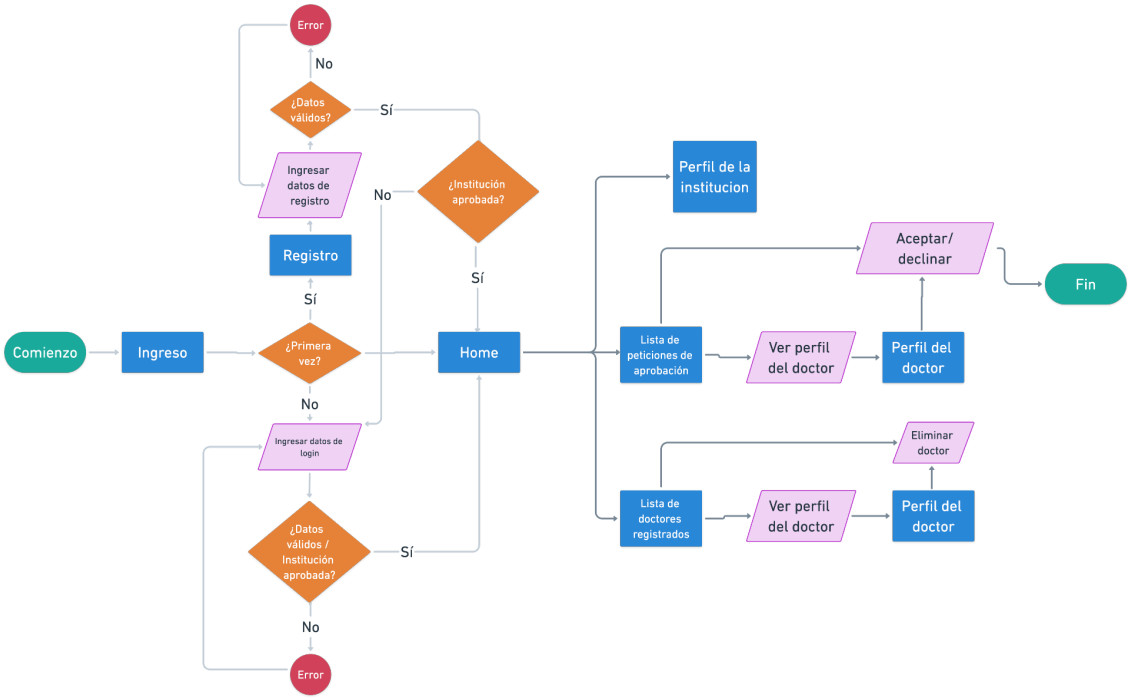


Figura 5.12: Estructura jerárquica de la interfaz gráfica de usuario para instituciones

Ente gubernamental

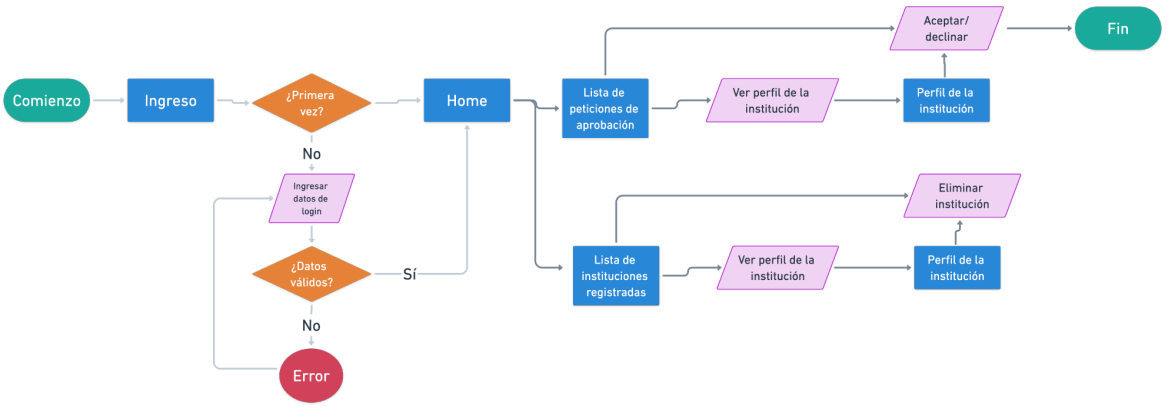


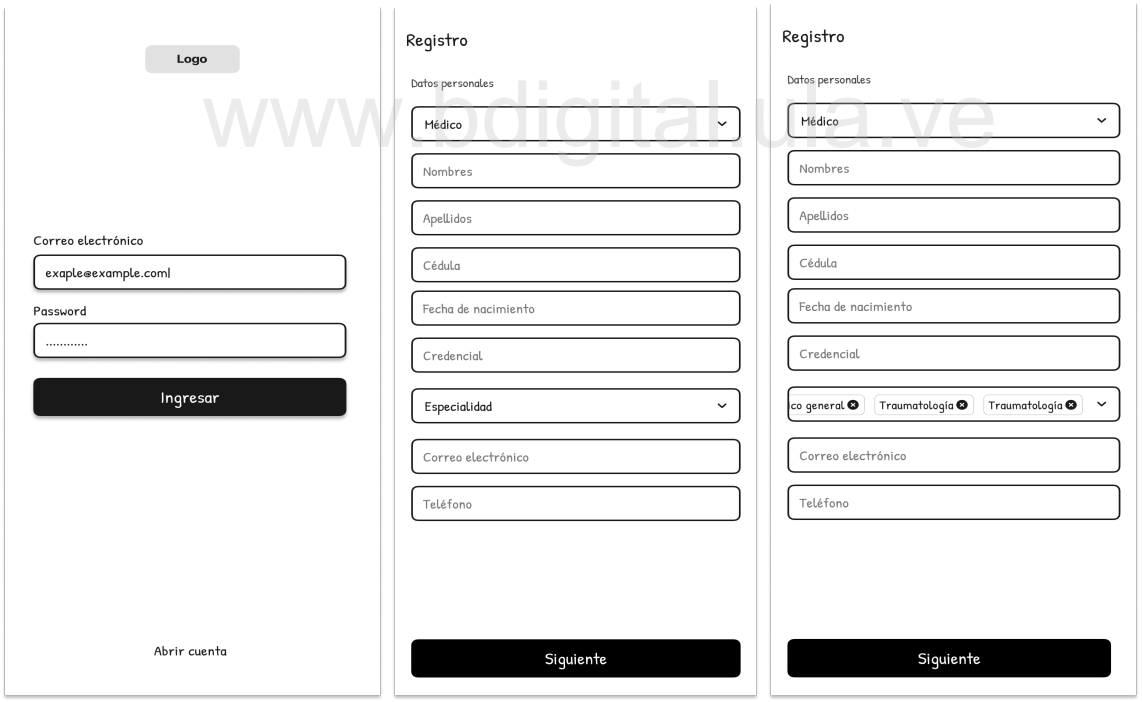
Figura 5.13: Estructura jerárquica de la interfaz gráfica de usuario para el ente gubernamental

5.3.2. Estructura general de las páginas web de la aplicación

La estructura general de las páginas web de la aplicación se diseñó de forma responsiva para que sea accesible desde dispositivos móviles y de escritorio. A continuación se ejemplifica la estructura de las páginas principales, de entrada y salida de datos, así como el menú, mensajes de error y estados vacíos.

Procesos de registro e inicio de sesión

Estas pantallas son las principales de la aplicación, las cuales permiten a los usuarios registrarse generando una identidad única e iniciar sesión en el sistema.



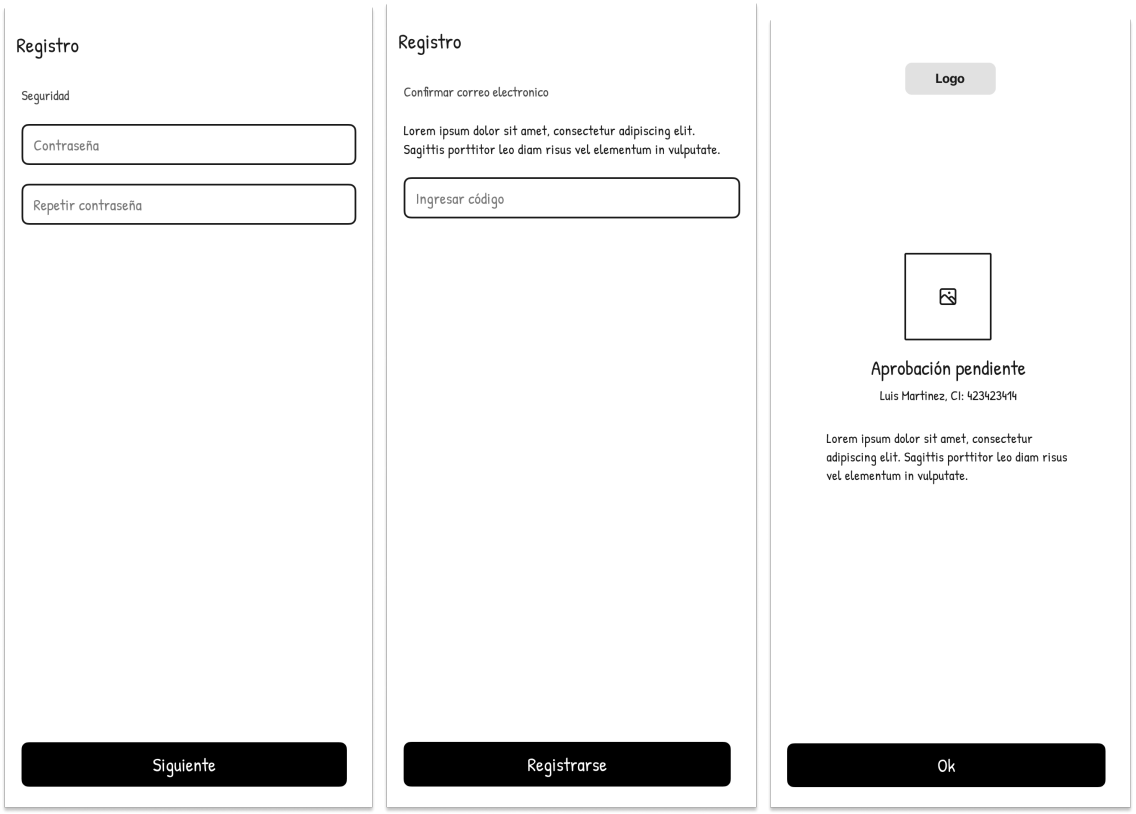


Figura 5.14: Interfaz gráfica de usuario de los procesos de registro e inicio de sesión

Menú en pantalla principal

El menú es el elemento que provee a las pantallas de la aplicación con la habilidad de navegar entre las pantallas principales dentro de los flujos de cada tipo de usuario.

Figura 5.15: Menú de navegación

Mensajes de éxito y error

Estos representan las respuestas visuales de la aplicación ante eventos generados por los usuarios, como la aplicación de filtros, guardado de datos, aprobación, denegación y revocación de accesos, etc.

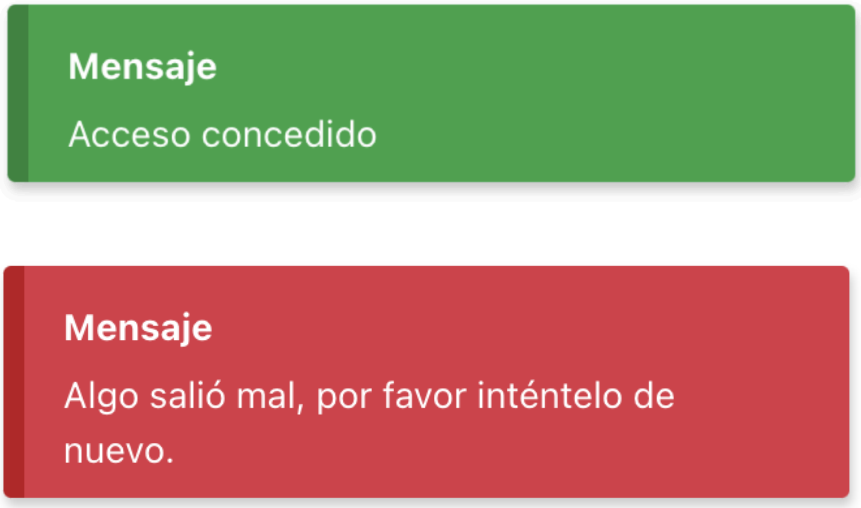


Figura 5.16: Mensajes de éxito y error

Estados vacíos

Pantallas que le informan a los usuarios que no hay datos actualmente para la funcionalidad que quieren consultar.

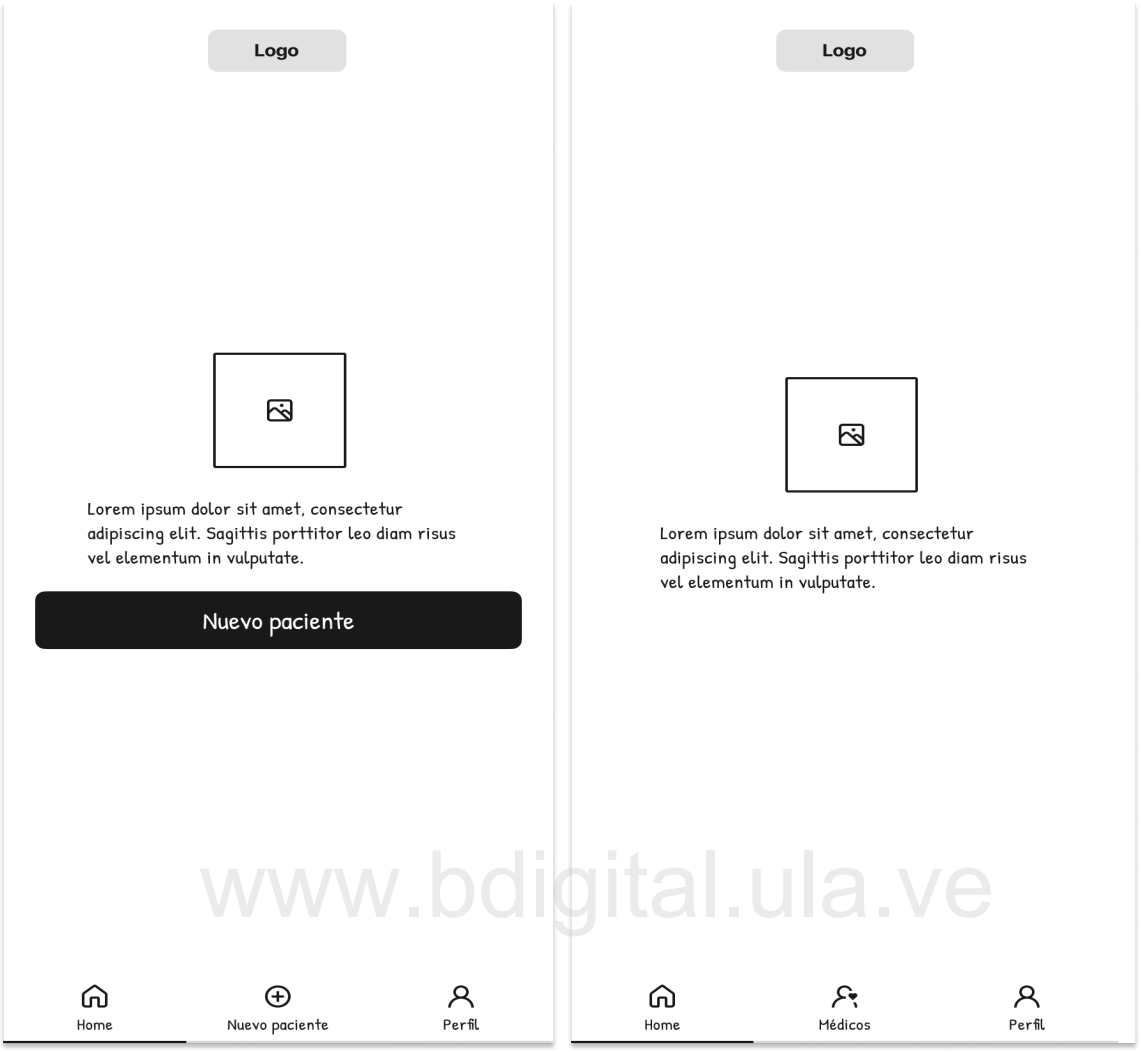


Figura 5.17: Estados vacíos

Filtros y búsqueda

Funcionalidades que le permiten a los usuarios buscar y filtrar información dentro de listados de datos para mayor facilidad de acceso a información específica.

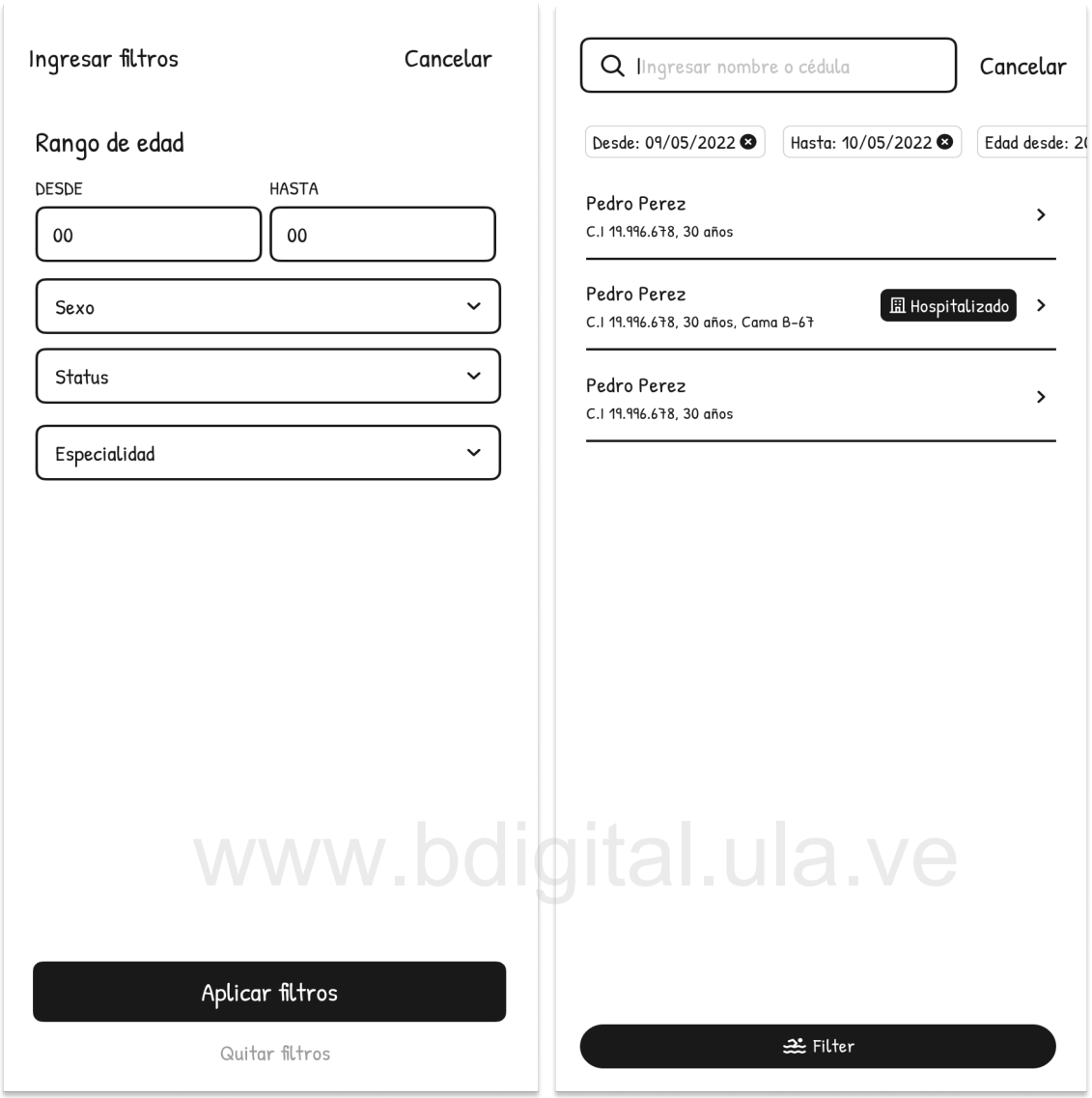


Figura 5.18: Filtros y búsqueda

Perfiles

Pantallas que muestran y permiten editar los datos de identidad de los usuarios a los cuales pertenecen, así como le brinda la visualización a los demás usuarios con acceso a ellos.

Figura 5.19: Perfiles

Vista de escritorio

Todas las pantallas de la aplicación cuentan con una versión web y de escritorio para mayor accesibilidad a los datos desde diferentes dispositivos.

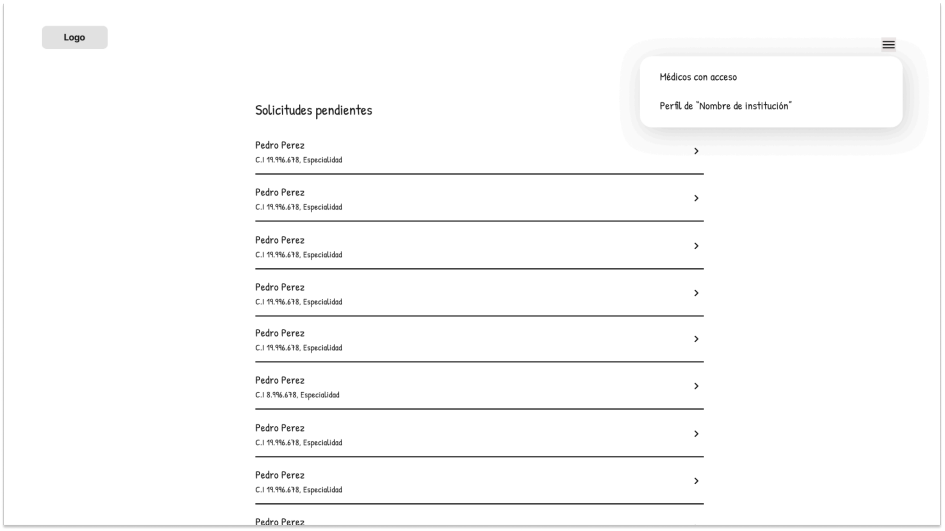


Figura 5.20: Vista de escritorio

5.3.3. Diseño de la base de datos

En la fase de diseño de la base de datos, se procedió a estructurar la información de manera que garantizara la integridad, seguridad y trazabilidad de los registros de identidad de los pacientes. En primer lugar, se llevó a cabo un detallado análisis de los requisitos, identificando los datos esenciales para el sistema y sus interrelaciones. A partir de este análisis, se elaboró un esquema conceptual que representó de forma abstracta la estructura de la base de datos, considerando los principios de normalización para minimizar la redundancia y maximizar la consistencia de los datos. Seguidamente, se definieron los lineamientos para la administración de la base de datos, estableciendo las reglas de acceso, seguridad y respaldo. Finalmente, se consolidó el diseño, dando lugar a un modelo de datos que servía como base para la implementación de la base de datos auxiliar.

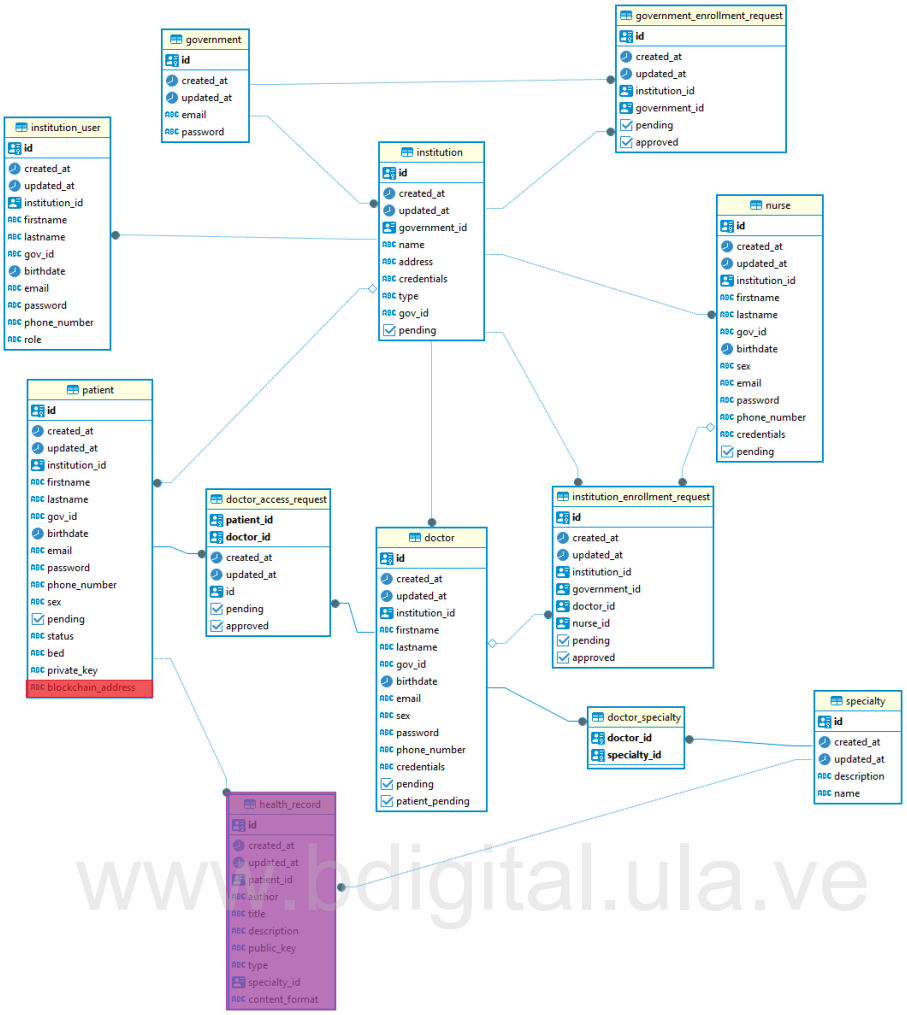


Figura 5.21: Diagrama de la base de datos

Capítulo 6
Implementación y pruebas

El presente capítulo se enfoca en la implementación y pruebas de la aplicación. El objetivo principal es definir el número de iteraciones necesarias para producir incrementos funcionales y entregables a los usuarios, así como establecer la duración de cada iteración. A través de un enfoque incremental y en ciclos iterativos, se busca desarrollar la aplicación de manera eficiente, asegurando que cada iteración aporte mejoras significativas al sistema y responda de manera óptima a las necesidades de los usuarios. Este capítulo detalla el proceso de planificación, desarrollo y evaluación continua que permitió la evolución progresiva del sistema, garantizando su fiabilidad y eficacia en el manejo seguro y eficiente de la identidad de los pacientes.

6.1. Planificación de entregas

La planificación de entregas es un proceso fundamental en el desarrollo de aplicaciones limitadas por un tiempo límite. Este proceso abarca desde la definición de criterios para descomponer el sistema en incrementos entregables hasta la estimación del esfuerzo requerido para cada incremento. A continuación, se detallan los pasos seguidos en esta planificación.

Establecimiento de criterios para descomponer el sistema en incrementos entregables

El criterio utilizado para descomponer el sistema fue el de descomposición por aspectos. Este enfoque consiste en extraer de los incrementos funcionales de la aplicación aquellas funciones que son transversales o comunes a varios de estos incrementos y

tratarlas como incrementos separados. Esto permite una gestión más eficiente y focalizada de las funcionalidades críticas y comunes del sistema, asegurando que se aborden de manera coherente y uniforme a lo largo del desarrollo.

División de la arquitectura del sistema en incrementos entregables

La arquitectura del sistema se dividió en un conjunto de incrementos funcionales que proporcionan valor al cliente. Estos incrementos son utilizables de manera parcial y temporal por los usuarios, lo que permite una implementación gradual y una validación continua de las funcionalidades desarrolladas. Cada incremento tiene asociado un conjunto de requisitos funcionales y no funcionales que se derivan de la Lista del Producto. Esta aproximación asegura que los incrementos entregables cumplan con el principio de modularidad, facilitando así su desarrollo, prueba e integración.

Establecimiento de la duración fija de las iteraciones (*sprints*)

Se determinó que la duración de cada iteración o *sprint* sería de dos semanas. Este periodo fue elegido para equilibrar la necesidad de realizar entregas frecuentes y la cantidad de trabajo que el equipo puede completar en un tiempo razonable. Las iteraciones de dos semanas permiten un ciclo de retroalimentación rápido y constante, lo que facilita la adaptación a cambios y mejoras continuas en el desarrollo del sistema.

Estimación del esfuerzo de cada incremento entregable

En esta actividad, el equipo de desarrollo estimó el esfuerzo en horas-hombre requerido para producir cada incremento entregable. Esta estimación se hizo en base a la cantidad y complejidad de los requisitos asociados a cada incremento, así como a la experiencia previa del equipo en el desarrollo de funciones similares. La duración de las iteraciones se ajustó para estar en consonancia con el esfuerzo promedio requerido para desarrollar los incrementos entregables, asegurando que cada *sprint* sea manejable y productivo. Esta planificación meticulosa garantiza que el equipo pueda cumplir con los plazos establecidos y entregar un producto de alta calidad de manera eficiente.

Plan de entregas e incrementos

Entrega 1

El primer incremento está compuesto de los componentes de las diferentes capas de la aplicación relacionados a los procesos de registro, inicio de sesión y visualización de perfiles de todos los tipos de usuarios de la aplicación, así como sus pruebas unitarias, de integración y de sistema en conjunto con la documentación relacionada. Este incremento le provee a los usuarios el primer acercamiento al uso de la aplicación donde pueden crear sus perfiles en la aplicación e ingresar a ella, teniendo acceso a sus datos almacenados en la aplicación.

Entrega 2

El segundo incremento contiene e integra al incremento anterior los componentes de las diferentes capas de la aplicación que permiten solicitar, visualizar, aprobar y denegar permisos a los respectivos usuarios, así como sus pruebas unitarias, de integración y de sistema en conjunto con la documentación relacionada. Este incremento le agrega a la aplicación las restricciones de seguridad necesarias para que los médicos, enfermeros y enfermeras e instituciones de salud posean la autorización necesaria antes de acceder a cualquier dato o funcionalidad relacionados a su rol.

Entrega 3

El tercer incremento incluye los componentes de las diferentes capas de la aplicación que hacen posible listar y filtrar los pacientes atendidos por un médico, médicos pertenecientes a alguna institución e instituciones pertenecientes a la red del ente gubernamental, incluyendo sus pruebas unitarias, de integración y de sistema en conjunto con la documentación relacionada. Integrándose a los incrementos anteriores, las funcionalidades de este incremento permiten tanto a los pacientes, médicos y enfermeros poder navegar hacia las historias clínicas y los datos a los que tienen acceso de los usuarios deseados.

Entrega 4

El cuarto incremento está constituido por los componentes de las diferentes capas de la aplicación que proporcionan las funcionalidades de crear, listar, modificar parcialmente y visualizar los datos de los usuarios limitados por las capacidades de su rol. Al igual que en los incrementos anteriores, dichos componentes incluyen sus pruebas unitarias, de integración y de sistema en conjunto con la documentación relacionada. Con este último incremento se cubren todas las necesidades de los usuarios especificadas en las historias de usuario y los requisitos no funcionales relacionados al dominio de la aplicación.

6.2. Desarrollo y operación de incrementos

Una vez definido el plan de entregas e incrementos, se procedió al desarrollo y operación de los mismos. Las siguientes etapas se repitieron para cada incremento y al final de cada uno se realizó una demostración del software al cliente para obtener retroalimentación y validar los avances. Esta retroalimentación se utilizó para mejorar la aplicación y planificar el siguiente incremento.

6.2.1. Planificación de las iteraciones

Al inicio de cada iteración, se revisó la Lista del Producto (*Product Backlog*) para incorporar requisitos nuevos o pendientes de la iteración anterior. Esto implicó analizar las necesidades del cliente, priorizar las funcionalidades y definir los criterios de aceptación para cada requisito.

A continuación se seleccionaron las historias de usuario relacionadas a las funcionalidades del incremento actual de la Lista del Producto para ser implementadas en la iteración. Estas historias se plasmaron en la Lista de Iteración (*Sprint Backlog*), la cual fue elaborada o actualizada en conjunto con el equipo de desarrollo.

Para cada historia de la Lista de Iteración, se definieron las tareas específicas que se debían realizar para completarla. Posteriormente, se estimó el esfuerzo y la duración de cada tarea, considerando la complejidad técnica y la experiencia del equipo. Las tareas de

la iteración se distribuyeron entre los miembros del equipo de desarrollo de acuerdo a sus habilidades y tomando en cuenta la carga de trabajo de cada miembro.

6.2.2. Refinamiento de requisitos de los incrementos

Planificada la iteración actual, se revisó y actualizó el diagrama de casos de uso para detallar los requisitos funcionales a implementar en el incremento. Esto implicó analizar las interacciones entre los usuarios y el sistema, definir los casos de uso principales y secundarios, y especificar los flujos de eventos alternativos y excepcionales.

De igual manera, se revisó y actualizó el diagrama de clases para representar la estructura del sistema y las relaciones entre las clases que implementan los requisitos del incremento. Para los casos de uso que lo requirieron, se actualizaron los diagramas de flujo de eventos para describir en detalle las interacciones entre los actores, el sistema y los datos. Estos diagramas permiten visualizar los pasos específicos que se ejecutan en cada caso de uso.

Por último, se revisaron y actualizaron los requisitos no funcionales de la iteración actual, como el rendimiento, la seguridad, la usabilidad y la confiabilidad para asegurar que estuvieran alineados con los objetivos del proyecto y las expectativas de los usuarios.

6.2.3. Diseño detallado de los incremento

Para esta etapa de las iteraciones, se revisó y actualizó la arquitectura de la aplicación para acomodar los cambios a introducir en el incremento actual. Esto implicó analizar los componentes de la Vista Arquitectónica asociados al incremento, definir las interfaces entre componentes, y especificar las tecnologías que se utilizarán para su implementación.

De igual modo, se revisó y optimizó el grafo de navegación de la IGU para facilitar la navegación de los usuarios por las diferentes funcionalidades del sistema. Se aseguraron flujos de navegación intuitivos y consistentes, al mismo tiempo que se refinaron los

mock-ups de cada página de la IGU del incremento en desarrollo. Esto implicó refinar la distribución de los elementos gráficos, la organización de la información y la aplicación de la paleta de colores definida.

A continuación, se diseñaron los elementos gráficos de la IGU asociada al incremento en desarrollo, como íconos, botones, menús y otros elementos visuales. Se consideró la estética general del software, la ergonomía y la accesibilidad, para luego actualizar el Diseño Detallado de la IGU, incluyendo las especificaciones técnicas necesarias.

El siguiente paso fue elaborar el esquema conceptual parcial del incremento en desarrollo, el cual representa la estructura lógica de los datos que se utilizaron en el incremento. Este esquema incluye las entidades, sus atributos y las relaciones entre ellas. Dicho esquema se integró al esquema conceptual general de la base de datos. Esto implicó identificar las entidades y relaciones existentes en el esquema conceptual general y mapearlas con las del esquema parcial. Posteriormente, se actualizó el esquema lógico de la base de datos, el cual representa la estructura de la base de datos en términos del modelo de datos relacional. Este esquema incluye tablas, columnas, índices y restricciones, con el cual se actualizó el esquema físico de la base de datos, el cual representa la forma en que los datos se almacenarán en el sistema de almacenamiento físico de la base de datos auxiliar y la cadena de bloques. Este esquema incluye la organización de las tablas en el espacio de almacenamiento, los tipos de datos utilizados y las configuraciones de rendimiento.

Contando con la estructura visual y conceptual de la aplicación, se procedió al diseño de los programas del incremento, para lo cual seleccionó una función específica (caso de uso o historia de usuario) de la Lista de Iteración actual para realizar un análisis y diseño detallado. Por cada función elegida, se elaboró una Realización de Caso de Uso, siguiendo un enfoque independiente de la plataforma hardware o software. Este análisis incluyó los siguientes pasos:

1. Identificación de clases de negocio: Se identificaron las clases de negocio que participan en la función seleccionada. Estas clases representan los objetos del dominio del problema y encapsulan sus atributos, relaciones y comportamiento.

2. Definición de atributos, relaciones y responsabilidades: Para cada clase de negocio identificada, se definieron sus principales atributos, relaciones y responsabilidades. Los atributos representan las características de los objetos, las relaciones representan las conexiones entre ellos y las responsabilidades representan las acciones que pueden realizar.

3. Elaboración del diagrama de clases participantes: Se elaboró un diagrama de clases participantes para visualizar las clases de negocio identificadas, sus atributos, relaciones y responsabilidades. Este diagrama representa la estructura estática del caso de uso. Se identificaron los servicios (operaciones) que cada clase participante debe ofrecer para cumplir con los requisitos del caso de uso. Las operaciones representan las acciones que las clases pueden realizar sobre sus atributos y las relaciones entre ellas.

Al finalizar estos pasos se procedió a actualizar el comportamiento (conjunto de operaciones) de las clases participantes para incluir las operaciones identificadas en los pasos anteriores. Y por último, se elaboró un Diagrama de Secuencias para describir el comportamiento de la función seleccionada. Este diagrama muestra la interacción entre las clases participantes y las operaciones que se invocan en cada paso del caso de uso.

Teniendo la realización del Caso de Uso elaborada previamente, el siguiente paso fue refinarla tomando en cuenta la plataforma de software elegida y los requisitos arquitectónicos del proyecto. Este refinamiento incluyó los siguientes pasos:

1. Análisis de requisitos arquitectónicos: Se analizaron los requisitos arquitectónicos del incremento asociados al caso de uso seleccionado. Estos requisitos se refieren a la forma en que el software se estructura, organiza y comunica con los elementos externos.

2. Identificación de mecanismos de implementación: Se identificaron los mecanismos de implementación de los requisitos arquitectónicos del caso de uso. Estos mecanismos definen cómo se traducirán los requisitos arquitectónicos en código fuente.

3. Identificación de patrones de diseño: Se identificaron patrones de diseño que pueden ser aplicados para diseñar el programa. Los patrones de diseño son soluciones probadas y reutilizables a problemas comunes de diseño de software.

Posteriormente, se identificaron las clases de interfaz que se requerían para manejar la interacción usuario-programa, ya que son las que definen la forma en que el usuario interactúa con el software y cómo el software responde a las acciones del usuario. Luego, se identificaron las clases de control que se necesitaban para manejar el flujo de control del programa, para así implementar la lógica del negocio y la secuencia de pasos que se ejecutan para cumplir con los requisitos del caso de uso. De la misma forma, se identificaron las clases que manejan los aspectos de persistencia, seguridad, manipulación de errores, etc, para proporcionar servicios de apoyo al funcionamiento del programa principal.

Finalmente, se refinó el Diagrama de Secuencias de la Realización del Caso de Uso incorporando las nuevas clases identificadas en los pasos anteriores. Este diagrama ahora muestra una visión más completa de la interacción entre las clases y las operaciones que se invocan en cada paso del caso de uso.

6.2.4. Codificación, pruebas e integración de los incrementos

En cada iteración (i-ésima), se seleccionaron las RCU que se implementarían de acuerdo con el plan de desarrollo y las prioridades del proyecto. La selección se basó en la complejidad de las RCU, la disponibilidad de recursos y las dependencias entre ellas. Para cada RCU seleccionada, se revisaron y actualizaron los diagramas de secuencias y de clases. Los diagramas reflejaron los últimos cambios en el diseño y garantizaron una representación precisa de la implementación.

A continuación, se diseñaron pruebas de unidad e integración para cada RCU. Las pruebas de unidad verificaron el correcto funcionamiento de cada módulo de código, mientras que las pruebas de integración garantizaron la interacción correcta entre estos módulos. Los programas del incremento se codificaron utilizando técnicas de programación convencionales, empezando por codificar manualmente el esqueleto del programa que implementa la RCU para facilitar la visualización de su estructura y la codificación posterior, para luego definir el orden en que se codificarían, integrarían y probarían los métodos de las clases del programa, considerando las dependencias entre ellos para asegurar una integración fluida.

El diseño de las pruebas de métodos se revisó y actualizó para todas las clases del programa, cubriendo todos los escenarios posibles y garantizando la calidad del código. Se codificaron los métodos de cada clase siguiendo el orden establecido, utilizando técnicas de programación adecuadas y buenas prácticas de desarrollo de software. Posteriormente, se prepararon y ejecutaron las pruebas de unidad para cada clase, corrigiendo los errores encontrados.

La calidad del código se verificó utilizando herramientas de análisis estático y revisiones de código, identificando y corrigiendo errores, problemas de estilo y posibles vulnerabilidades. Además, se refactorizó el código fuente de los programas que lo requirieron, mejorando su legibilidad, mantenibilidad y extensibilidad. Una vez codificados y verificados, los programas del incremento se integraron y se realizaron pruebas de integración para asegurar la correcta interacción entre ellos. Los errores encontrados se corrigieron y las pruebas se repitieron hasta obtener resultados satisfactorios.

Finalmente, se llevaron a cabo pruebas a nivel de sistema para verificar el correcto funcionamiento del incremento en conjunto con el resto del sistema. Estas pruebas cubrieron todos los escenarios posibles dentro del alcance del proyecto y aseguraron la calidad del sistema en su totalidad.

6.2.5. Verificación y validación de los incrementos

Una vez finalizada la integración y las pruebas del incremento, se programó una demostración del incremento integrado para mostrar a los interesados las nuevas funcionalidades y mejoras implementadas. La demostración incluyó una presentación de las características del incremento, así como una demostración práctica de su funcionamiento.

Los defectos encontrados durante la revisión y las pruebas se corrigieron cuidadosamente. Las correcciones se integraron con el código funcional de las iteraciones anteriores y se realizaron pruebas de regresión para garantizar que no se introdujera nuevos errores.

Al final de cada iteración se llevó a cabo una reunión de retrospectiva para revisar el proceso de desarrollo del incremento. La reunión tuvo como objetivo identificar las lecciones aprendidas, las buenas prácticas y las áreas de mejora para las próximas iteraciones.

6.2.6. Cierre de proyecto

De acuerdo con la metodología Blue Watch, esta etapa del proyecto se encarga de la recopilación de información del proyecto luego de ser puesto en operación, así como de la evaluación de la ejecución técnica del mismo y más. En nuestro caso por ser el producto resultante un prototipo funcional, no fue puesto en operación, por lo que en el cierre del proyecto lo que se hizo fue evaluar los resultados del proyecto y de las pruebas del sistema para redactar las siguientes conclusiones y recomendaciones finales.

Capítulo 7
Conclusiones y recomendaciones

El desarrollo del sistema de gestión de identidad de pacientes basado en tecnología blockchain logró alcanzar los objetivos propuestos, contribuyendo significativamente a una mejora de la calidad de la atención médica, fomentando la interoperabilidad entre instituciones de salud. A través de una exhaustiva revisión bibliográfica y entrevistas a los usuarios finales del sistema, se identificaron las principales deficiencias y problemáticas en la gestión de identidades de pacientes. Estos desafíos incluyen la falta de precisión en la identificación, problemas de duplicación de registros y vulnerabilidades en la protección de datos personales.

Además, la investigación demostró que la tecnología blockchain es altamente idónea para la gestión de identidades de pacientes, gracias a sus características de descentralización, inmutabilidad y alta seguridad. También se identificaron casos de uso relevantes que sirvieron de referencia para el desarrollo del sistema. El diseño conceptual y arquitectónico del sistema abordó eficazmente los desafíos identificados, proponiendo una arquitectura basada en cadena de bloques que incluye mecanismos para asegurar la privacidad y seguridad de los datos de los pacientes, así como para facilitar la interoperabilidad entre diferentes instituciones de salud.

Asimismo, se logró desarrollar un prototipo funcional del sistema, demostrando la viabilidad técnica del enfoque propuesto. El prototipo implementa tecnología *blockchain* para garantizar la integridad y privacidad de los datos, y ofrece funcionalidades que mejoran la precisión en la identificación de pacientes. Las pruebas exhaustivas y análisis de vulnerabilidades realizados confirmaron que el sistema es eficiente y seguro,

demostrando una alta capacidad para prevenir errores de identificación y proteger la información confidencial de los pacientes.

Por otro lado, las pruebas de integración y análisis de compatibilidad demostraron que el sistema facilita el intercambio seguro de datos médicos entre diferentes instituciones de salud, mejorando así la colaboración y la continuidad de la atención médica. Se recomienda una implementación gradual del sistema en las instituciones de salud, comenzando con proyectos piloto en entornos controlados para ajustar y mejorar el sistema basado en los comentarios de los usuarios finales. Además, es fundamental realizar programas de capacitación y concienciación para los profesionales de la salud sobre el uso del sistema y la importancia de la tecnología blockchain en la protección de datos y mejora de la atención médica.

Dado el rápido avance de la tecnología blockchain, se recomienda mantener una actualización continua del sistema para incorporar nuevas funcionalidades y mejoras en la seguridad y eficiencia. También es esencial colaborar con entidades regulatorias para establecer políticas y normativas claras que respalden el uso de tecnología *blockchain* en la gestión de identidades de pacientes, asegurando así el cumplimiento legal y ético.

Es necesario destacar que el producto de este proyecto de grado cumplió con todos los requisitos identificados en la etapa de análisis, sin embargo, por las limitantes de tiempo, personal y presupuesto, hay varios aspectos que se pueden mejorar en el producto. Uno de ellos sería el uso de certificados de capa de sockets seguros (SSL por su siglas en inglés) para todas las conexiones de la aplicación que manejen este protocolo y así encriptar la transferencia de datos entre pares. También debería evitarse el almacenamiento de la llave privada de la identidad del paciente en la base de datos y delegarle su almacenamiento únicamente a los usuarios, afianzando así su propiedad sobre los datos. Del mismo modo, una de las mejoras que contribuiría a una mayor descentralización y democratización de la información sería el uso de una cadena de bloques más popular como Ethereum, por ejemplo. Y por último, con el fin de mejorar la escalabilidad del sistema, se recomienda la orquestación de sus contenedores con alguna herramienta como Kubernetes para garantizar una utilización y un consumo más eficientes de los recursos.

En conclusión, el sistema de gestión de identidad de pacientes basado en tecnología blockchain ofrece una solución innovadora y efectiva para los desafíos actuales en la identificación y protección de datos de pacientes, marcando un avance significativo en la calidad y seguridad de la atención médica.

Anexos

Entrevistas a los médicos

Las preguntas principales en las entrevistas a los profesionales de la salud que conformaron la muestra de este proyecto fueron las siguientes:

- ¿Cuál es tu experiencia con el manejo de la identificación de pacientes?
- ¿Alguna vez has tenido problemas de identificación con algún paciente?
- ¿Qué te parece que está bien en la forma actual en que manejas la identificación de pacientes?
- ¿Qué te parece que está mal en la forma actual en que manejas la identificación de pacientes?
- ¿Qué le mejorarías a la forma actual en que manejas la identificación de pacientes?
- ¿Te sería factible una aplicación de escritorio, tablet o móvil para manejar la identidad de tus pacientes?

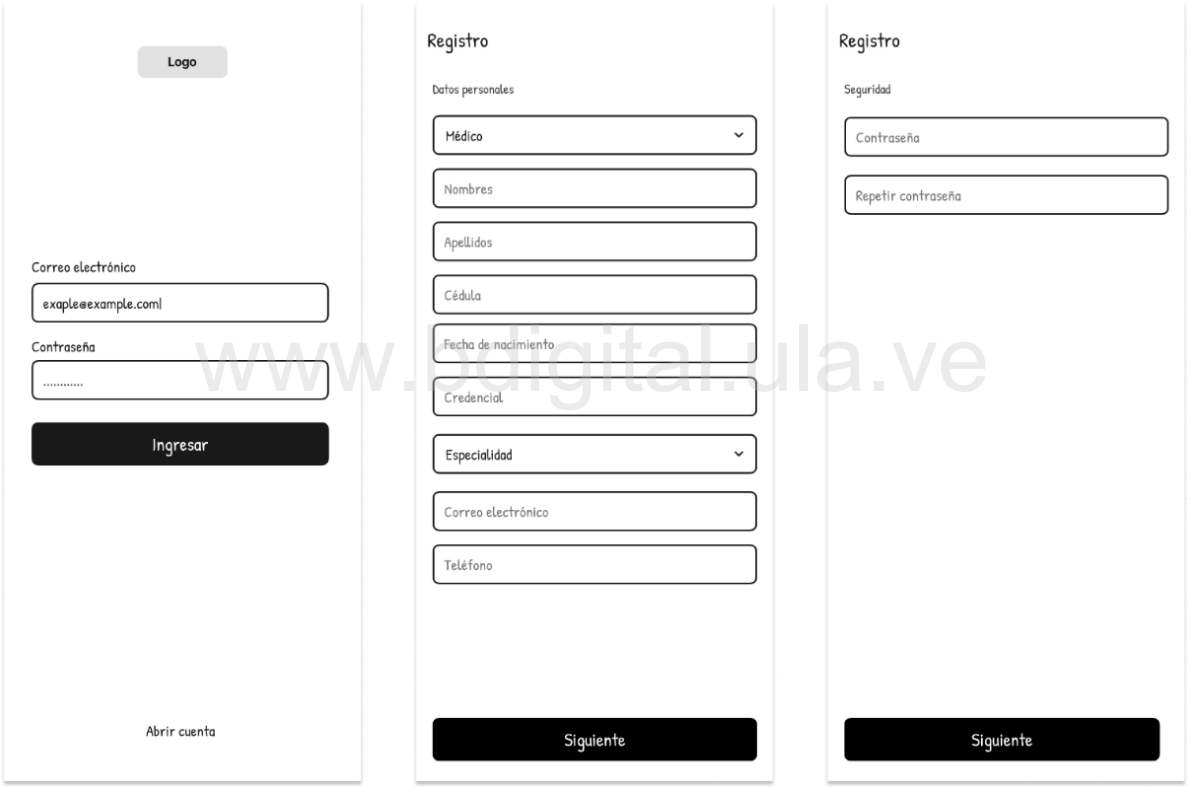
Entrevistas a los pacientes

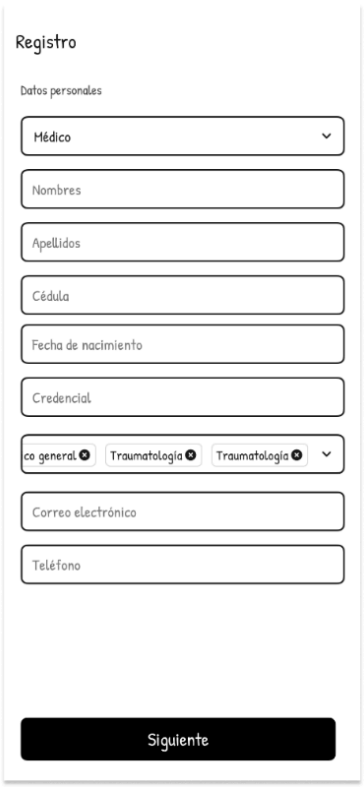
Las preguntas principales en las entrevistas a los pacientes regulares del IAHULA que conformaron la muestra de este proyecto fueron las siguientes:

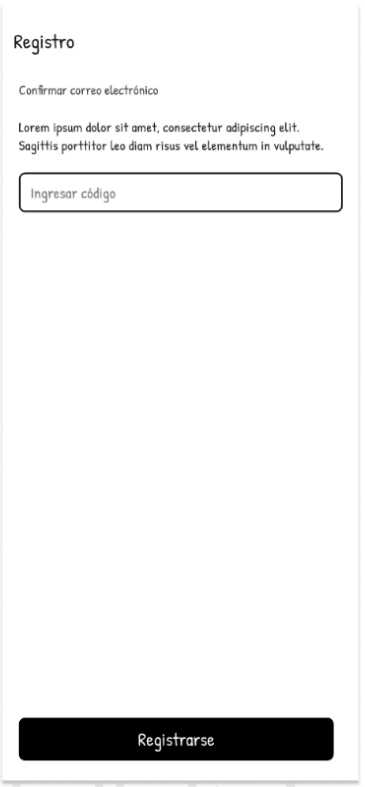
- ¿De qué forma te identificas cuando vas al médico? Es decir, ¿Qué información personal te solicita el médico para sus registros?
- ¿Alguna vez has tenido problemas con tu identidad en alguna institución de salud?
- ¿Qué mejorarías de la forma en que te identificas en las instituciones de salud?

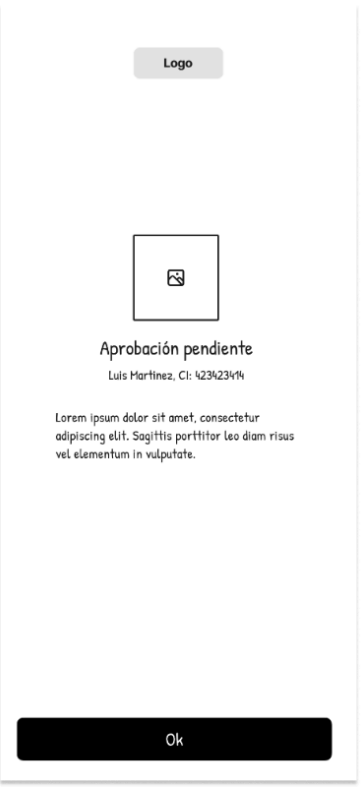
- ¿Cuando necesitas cambiar de institución médica cómo te identificas en la nueva institución?
- ¿Te sería factible una aplicación de escritorio, tablet o móvil para manejar tu identificación en las instituciones de salud?

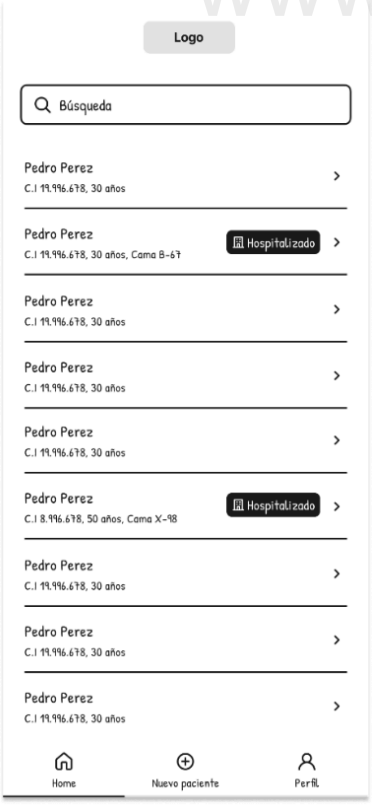
Interfaz gráfica

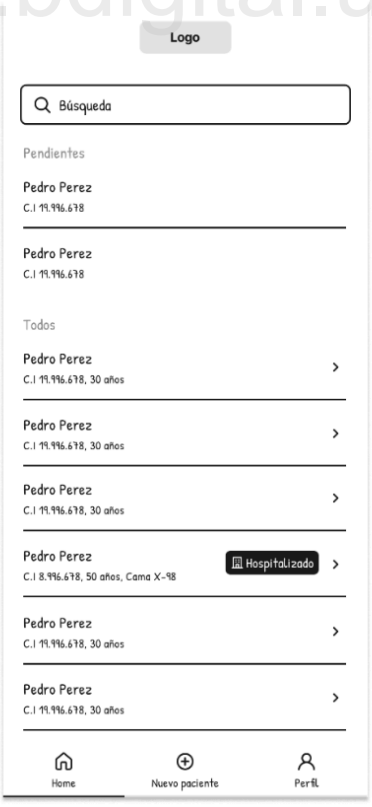


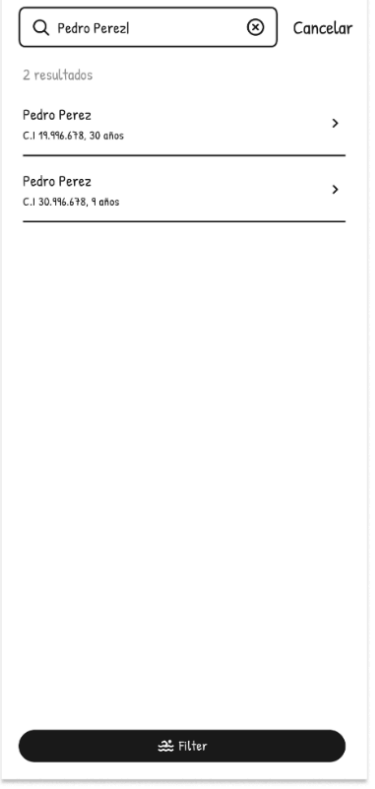


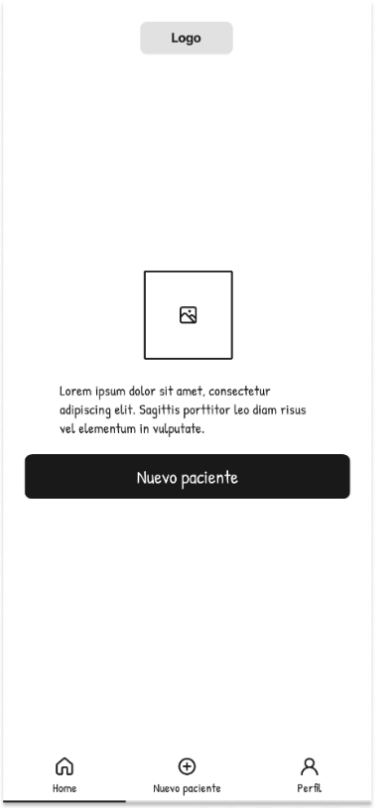


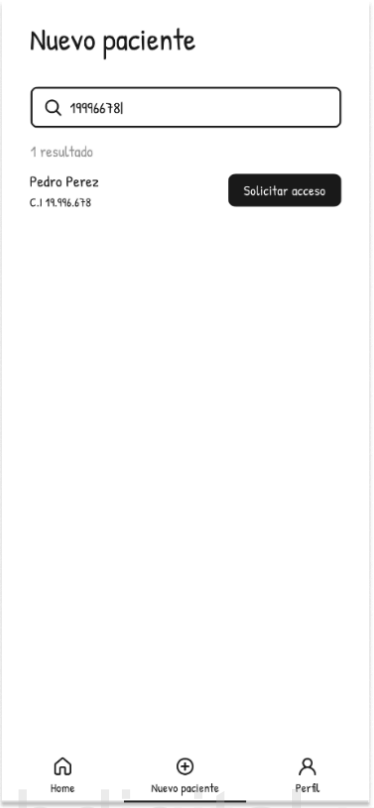


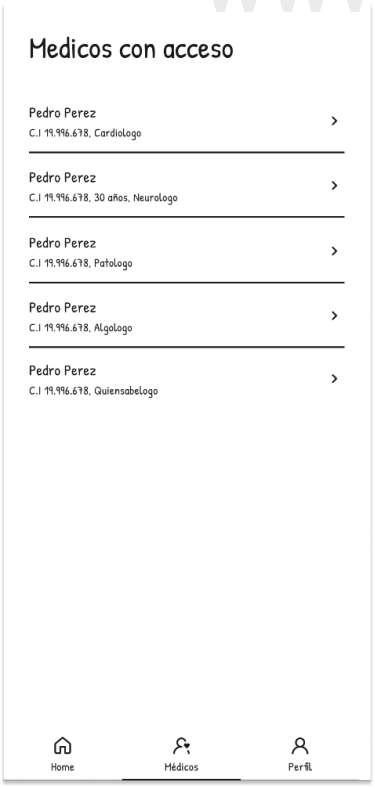


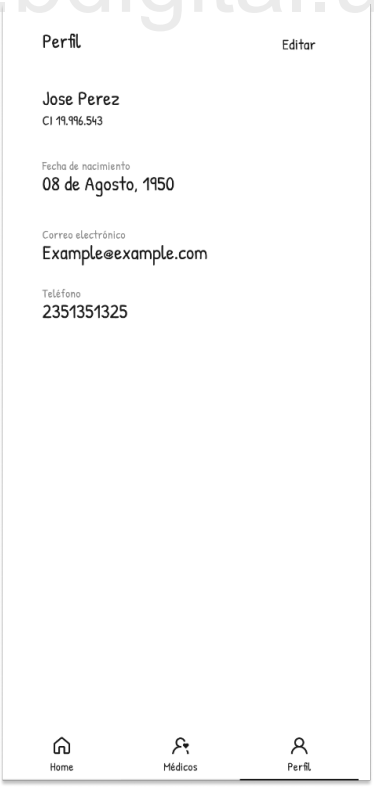


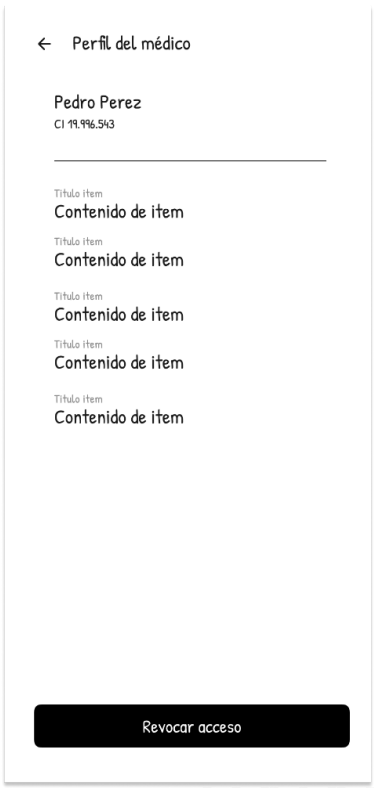


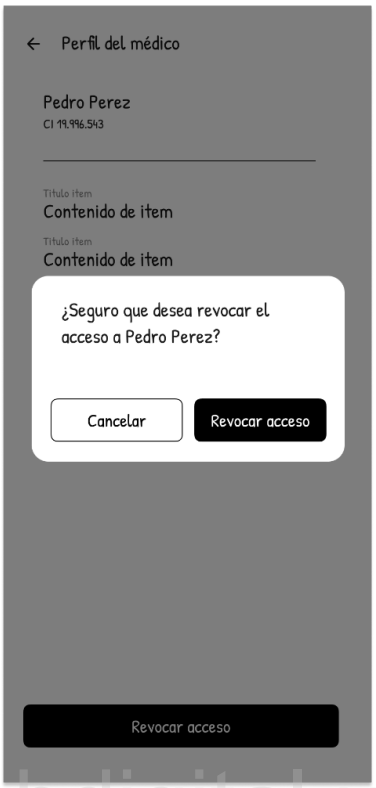


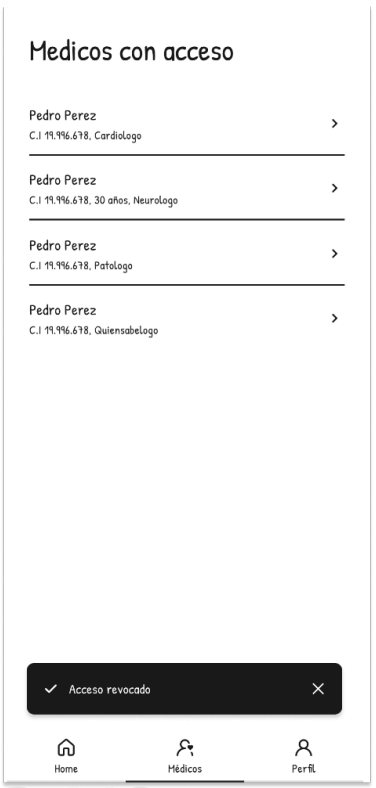


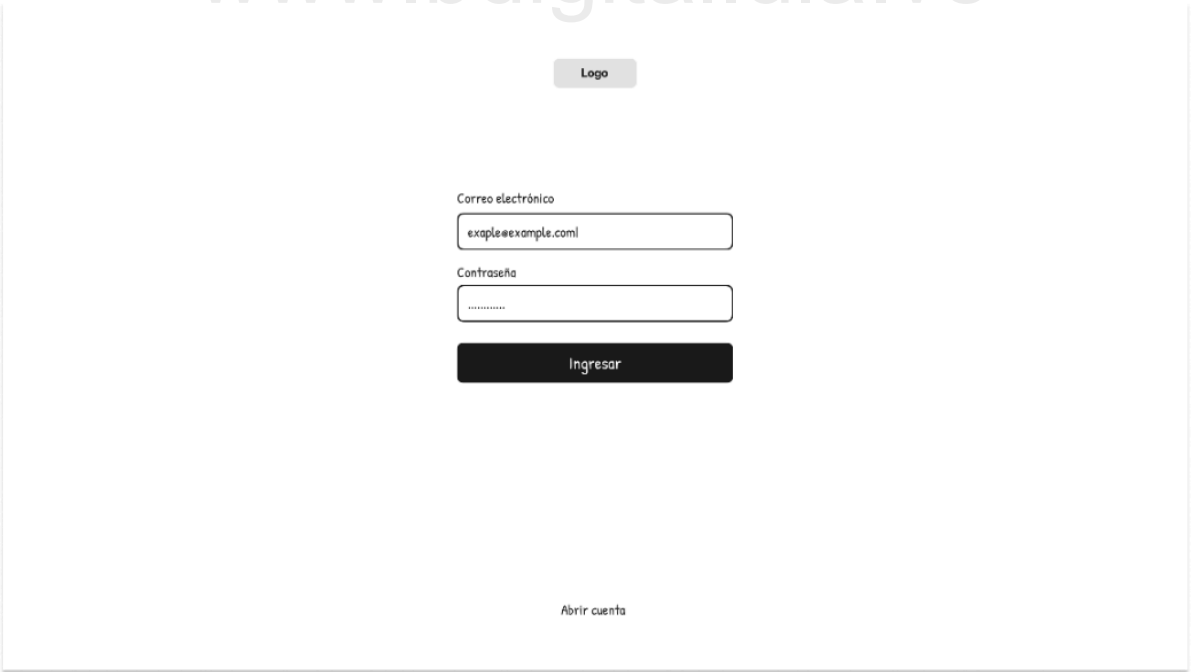


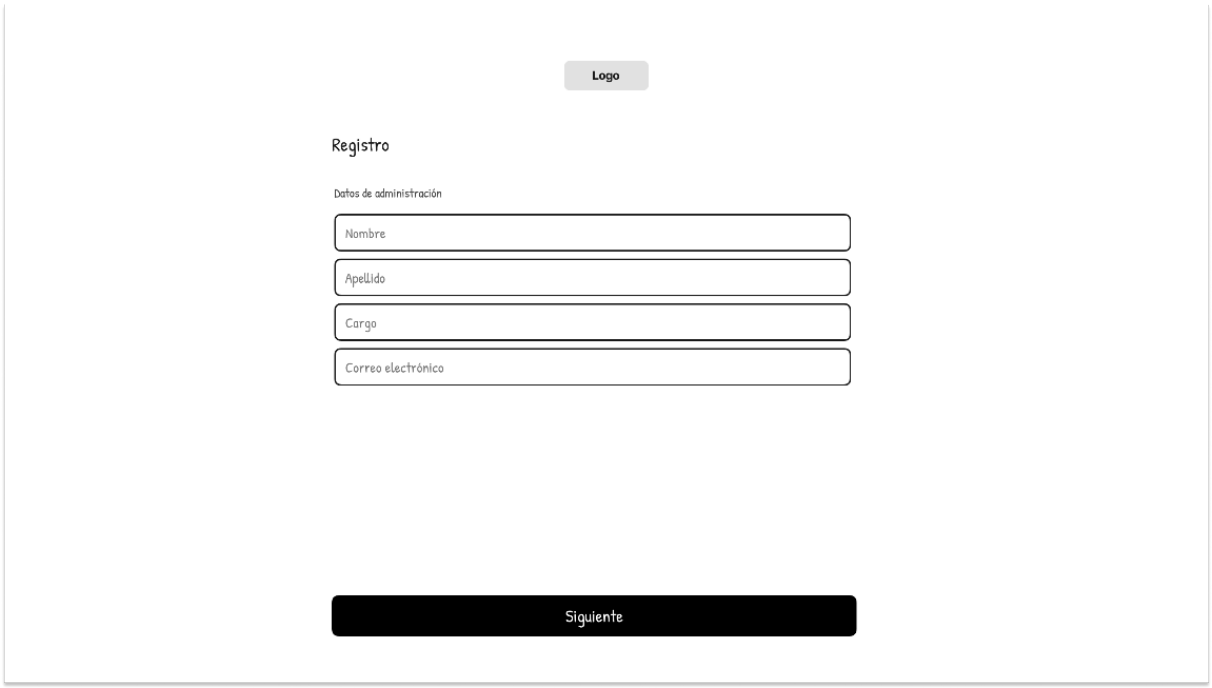


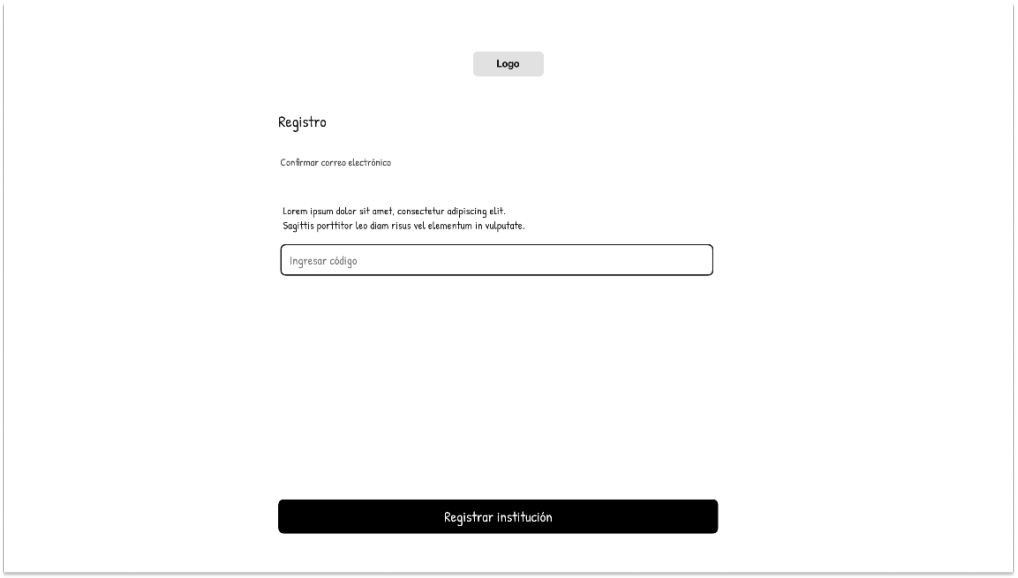


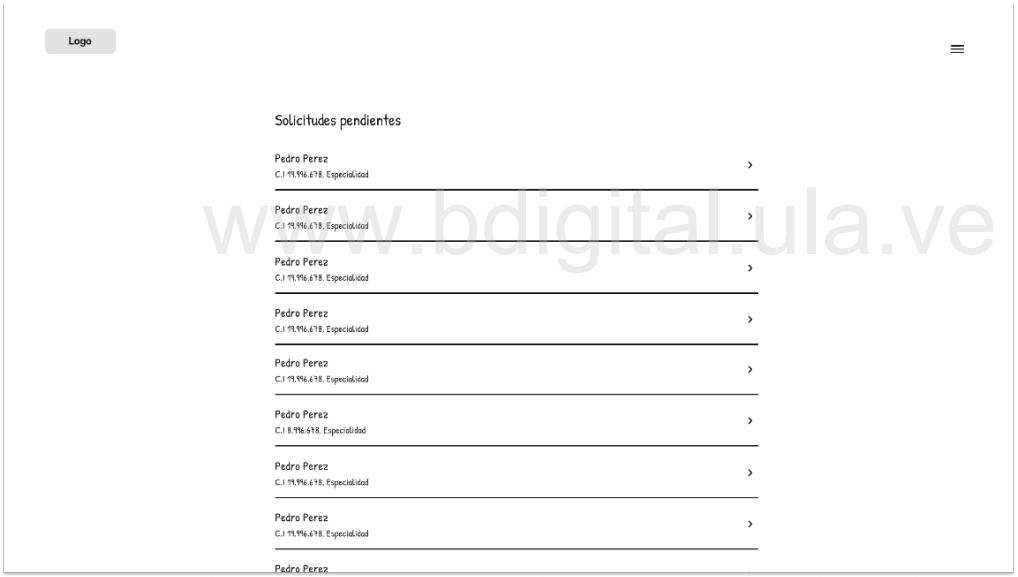


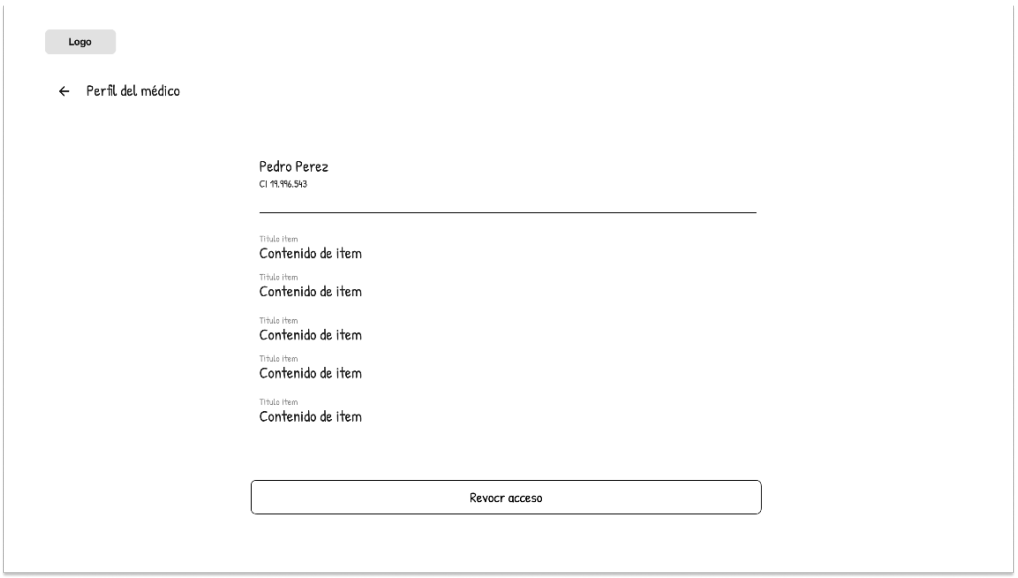


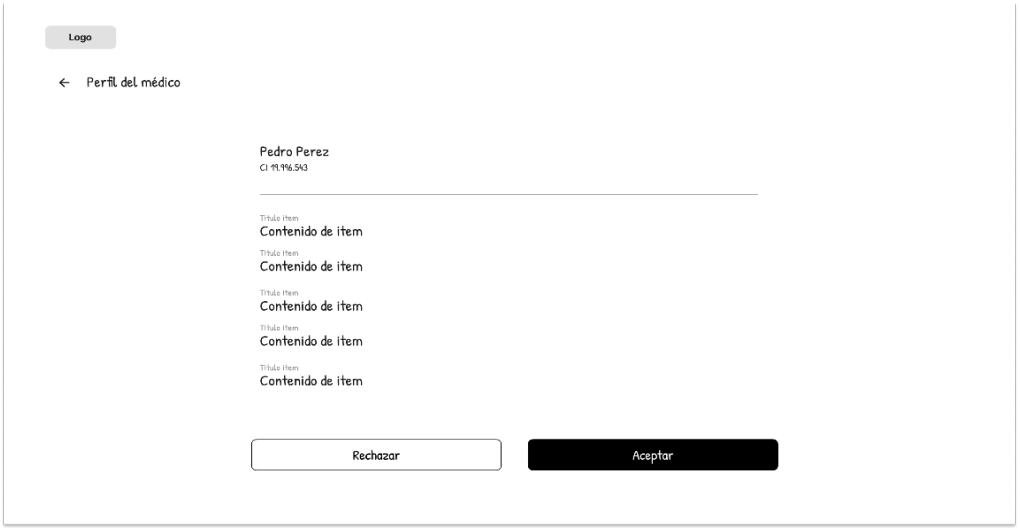


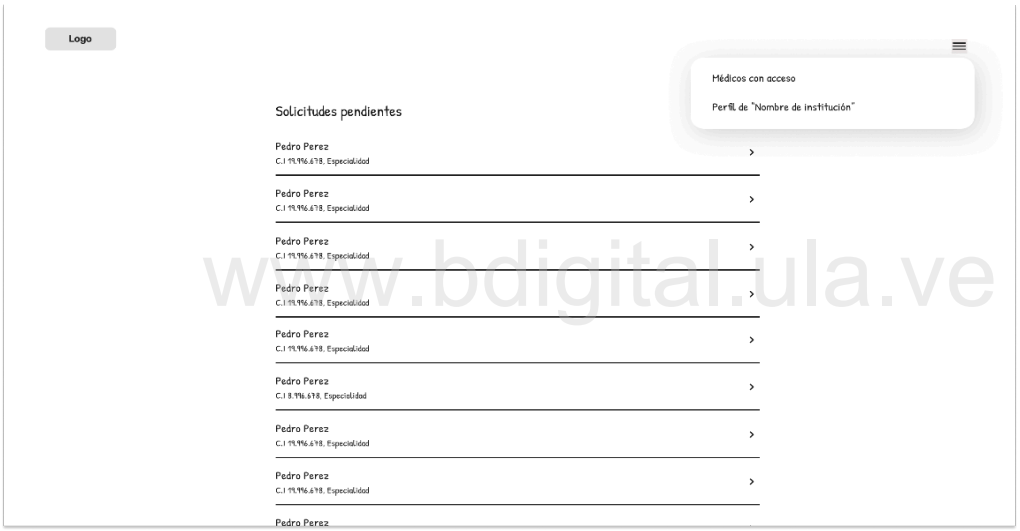


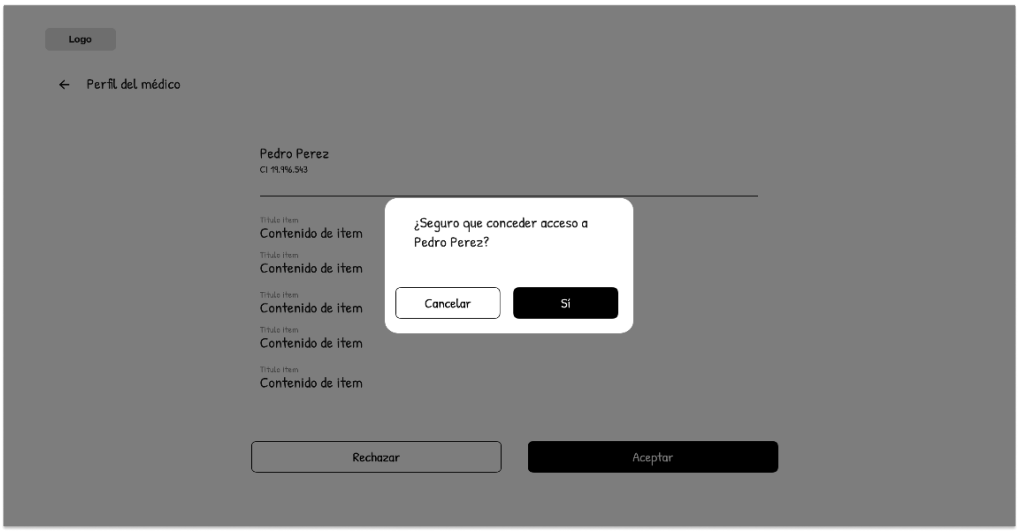












Bibliografía

Allamde Cusso, R., Macías Seda, J., & Porcel Gálvez, A. M. (2019). La relación enfermera-paciente: Identidad histórica, metodológica y terapéutica en los cuidados de enfermería. Cultura de los cuidados, 23(55), 78. https://doi.org/10.14198/cuid.2019.55.08

Anderson, G. W. (2022). Design Thinking for Tech: Solving Problems and Realizing Value in 24 Hours. Pearson.

Cáceres, E. A. (2014). Análisis y diseño de sistemas de información. Rivadavia: Facultad de Ciencias Sociales Universidad de San Juán.

Camburn, B., Dunlap, B., Gurjar, T., Hamon, C., Green, M., Jensen, D., Crawford, R., Otto, K., & Wood, K. (2015). A systematic method for design Prototyping. Journal of Mechanical Design, 137(8). https://doi.org/10.1115/1.4030331

Chen, Y., Ding, S., Xu, Z., Zheng, H., & Yang, S. (2018). Blockchain-Based Medical Records Secure Storage and Medical Service Framework. Journal of Medical Systems, 43(1), Article 1. https://doi.org/10.1007/s10916-018-1121-4

Contreras P., A. F. (2020). Marco normativo de la historia clínica electrónica y su incidencia en el ámbito de la protección de datos personales en Colombia. Revista La Propiedad Inmaterial, (29), 95-116. https://doi.org/10.18601/16571959.n29.04

Dennis, A., Wixom, B., & Tegarden, D. (2015). Systems analysis and design: An object-oriented approach with UML. John wiley & sons.

Díaz, J. A., Ocampo, M. C., Pantoja, J. E., & Román, E. A. (2015). Administración de Hospitales Y Servicios de Salud. Alpha Editorial.

Evans, R. S. (2016). Electronic health records: Then, now, and in the future. Yearbook of Medical Informatics, 25(S 01), S48-S61. https://doi.org/10.15265/iys-2016-s006

Fretheim, E., & Deschene, M. (2023). Secure software systems: Design and development. Jones & Bartlett Learning.

Gaynor, M., Yu, F., Andrus, C. H., Bradner, S., & Rawn, J. (2014). A general framework for interoperability with applications to healthcare. Health Policy and Technology, 3(1), 3-12. https://doi.org/10.1016/j.hlpt.2013.09.004

Joyanes, L. (2015). Sistemas de Información en la empresa. Alpha Editorial. https://books.google.co.ve/books?id=oHNxEAAAQBAJ

Khorikov, V. (2020). Unit testing principles, practices, and patterns. Manning Publications Co.

Kothari, C. R. (2004). Research methodology: Methods and techniques [Metodología de la Investigación: Métodos y técnicas]. (2.ª ed.). New Age International.

León-Castañeda, C. D. de. (2018). Revisión de temas fundamentales en sistemas de salud. Revista Médica Del Instituto Mexicano Del Seguro Social, 56(3), 295–304. https://www.redalyc.org/journal/4577/457757174018/html/

Ley de Reforma de la ley de Ejercicio de la Medicina. (2011, 1). https://www.asambleanacional.gob.ve/leyes/sancionadas/ley-de-reforma-de-la-ley-de-ejercicio-de-la-medicina

Martin, R. C. (2017). Clean architecture: A craftsman's guide to software structure and design. Prentice Hall.

Montilva, J. C., & Barrios, J. A. (2021). Ingeniería del Software. Un enfoque basado en procesos (1.ª ed.). Universidad de los Andes.

Nakamoto, Satoshi. "Bitcoin: A Peer-To-Peer Electronic Cash System." 31 Oct. 2008.

Paul, P., Aithal, P. S., Saavedra, R., & Ghosh, S. (2021). Blockchain technology and its types—a short review. International Journal of Applied Science and Engineering (IJASE), 9(2), 189-200.

Ramos, D., Noriega, R., Laínez, J. R., Durango, A., & Academy, I. T. C. (2017). Curso de Ingeniería de Software: 2ª Edición. CreateSpace Independent Publishing Platform. https://books.google.co.ve/books?id=G2Q4DgAAQBAJ

Sharma, P., Jindal, R., & Borah, M. D. (2020). Blockchain Technology for Cloud Storage: A Systematic Literature Review. ACM Computing Surveys, 53(4), Article 4. https://doi.org/10.1145/3403954

Silberschatz, A., Korth, H. F., & S Sudarshan. (2020). Database system concepts. Mcgraw-Hill Education.

Swan, M. (2015). Blockchain: Blueprint for a new economy. O'Reilly Media.

Vazirani, A. A., Brindley, D., & Meinert, E. (2020). Blockchain vehicles for efficient Medical Record management. *Npj Digital Medicine*, *3*(1), 1-5. https://doi.org/10.1038/s41746-019-0211-0