UNIVERSIDAD DE LOS ANDES
NÚCLEO "PEDRO RINCÓN GUTIÉRREZ"
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS

DESARROLLO DE UN SISTEMA DE GESTIÓN DE HISTORIAS CLÍNICAS EN INSTITUCIONES DE SALUD MEDIANTE TECNOLOGÍA DE CADENA DE BLOQUES

Autor: Br. Jorge L. Rodríguez G.

Tutor: Dr. Jonás Montilva

Julio 2024

Resumen

La gestión de historias clínicas es uno de los componentes más fundamentales en el proceso de atención médica. Este proyecto buscó, utilizando la tecnología de cadena de bloques, crear un sistema público, anónimo, seguro y descentralizado de gestión de historias clínicas que busca mejorar la interoperabilidad entre instituciones de salud y el control de los pacientes sobre su información. Para esto se diseñó e implementó un producto mínimo viable utilizando la metodología de gestión de proyecto Scrum y la metodología para el desarrollo de aplicaciones empresariales de mediana o pequeña complejidad y tamaño Blue Watch, realizando las pruebas unitarias acordes para poder verificar su funcionamiento así como las pruebas de integración necesarias para validar los estándares de interoperabilidad entre instituciones de salud definidos.

El proyecto logró alcanzar sus objetivos planteados, con un sistema que permite la gestión de historias clínicas utilizando la tecnología de cadena de bloques.

Palabras clave: Gestión de historias clínicas, cadena de bloques, institución de salud, interoperabilidad.

Índice

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La gestión de historias clínicas es un aspecto fundamental de la atención médica, ya que proporcionan un registro del estado de salud de un paciente a lo largo del tiempo y es esencial para la toma de decisiones informadas por parte de profesionales de la salud. Sin embargo, la gestión de historias clínicas ha enfrentado desafíos críticos en cuanto a la seguridad, la privacidad y la integridad de los datos, lo que ha generado desconfianza respecto a los métodos tradicionales de almacenamiento de historias clínicas. La tecnología de cadena de bloques o *blockchain*, como se le denomina en inglés, se destaca por su capacidad para garantizar la inmutabilidad, la seguridad y en ciertos aspectos la privacidad de los datos almacenados, de acuerdo con Hasselgren et al. (2020), ha emergido como una solución prometedora para abordar estos desafíos en el campo de la gestión de historias clínicas.

Haciendo implementación de la cadena de bloques en el contexto de la atención médica, tiene el potencial de transformar la forma en que se almacenan, acceden y comparten las historias clínicas de los pacientes, mejorando significativamente la seguridad de los datos, la privacidad del paciente y la posibilidad de interoperabilidad entre diferentes instituciones de atención

médica, que son características clave, de acuerdo con Nakamoto (2009), a las soluciones que proveen las cadenas de bloques.

Planteamiento del problema

Los sistemas de gestión de historias clínicas en las instituciones de salud enfrentan desafíos cruciales en términos de seguridad y privacidad de los datos, así como en interoperabilidad entre instituciones de salud, que han generado desconfianza en las instituciones de salud que utilizan métodos convencionales como documentos en papel o sistemas de almacenamiento de registros médicos electrónicas.

Con el constante descubrimiento de vulnerabilidades en sistemas computacionales aumenta el riesgo de la manipulación o pérdida de datos debido a fallas de seguridad, esto también demostraría una brecha de privacidad y podría resultar en la divulgación no autorizada de información sensible del paciente.

La falta de interoperabilidad entre diferentes instituciones de salud puede llevar a retrasos en el tratamiento de pacientes y aumentar los costos de atención médica debido a la necesidad de volver a realizar estudios que están almacenados en instituciones diferentes.

Estos desafíos aluden a que existe un vacío en el ámbito de la gestión de historias clínicas por una solución que los pueda mitigar. La tecnología de cadena de bloques, aparenta ser la mejor solución.

1.1. Delimitación del problema

- ***Enfoque en la gestión de historias clínicas***: Este proyecto de grado se centrará exclusivamente en la gestión de historias clínicas dentro del ámbito de la atención médica. Aunque la tecnología de cadena de bloques tiene aplicaciones en otros aspectos de la atención médica, como la gestión de récipes médicos o la identificación de pacientes (estudiado en el proyecto acompañante a este), se hará el enfoque solamente en la gestión integral de las historias clínicas de los pacientes.

- ***Ámbito geográfico específico***: La implementación y las regulaciones relacionadas con la atención médica pueden variar significativamente según la región y el país. Este proyecto de grado se llevará a cabo en un contexto específico: Mérida, Venezuela, teniendo en cuenta las regulaciones y requisitos legales en una ubicación geográfica específica, lo que puede limitar la aplicabilidad de los resultados a otras jurisdicciones.

- ***Tamaño de la muestra y alcance de la implementación***: La implementación real del sistema de cadena de bloques se llevará a cabo en un entorno limitado o con una muestra representativa. La escalabilidad y los desafíos prácticos de implementación a gran escala no serán el foco principal de esta investigación.

- ***Consideración ética y legal***: Aunque se abordarán cuestiones éticas y legales en la gestión de historias clínicas con tecnología de cadena de bloques, el proyecto de grado no proporcionará asesoramiento legal específico ni abordará exhaustivamente todas las implicaciones éticas en todas las situaciones posibles. En su lugar, se buscará ofrecer un marco general y recomendaciones.

- ***Limitaciones tecnológicas***: Aunque la tecnología de cadena de bloques se considera prometedora, este proyecto de grado reconocerá las limitaciones tecnológicas actuales en términos de escalabilidad, costos, velocidad de procesamiento y seguridad. No se asumirá que la cadena de bloques es la solución perfecta para todos los aspectos de la gestión de historias clínicas.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un sistema de gestión de historias clínicas que implemente mecanismos de descentralización, privacidad y seguridad, que mejore la calidad de la atención médica y fomente la interoperabilidad entre las instituciones de salud en beneficio de pacientes y profesionales médicos utilizando la tecnología de cadena de bloques.

1.2.2. Objetivos específicos

- ***Analizar desafíos actuales en la gestión de historias clínicas***: Realizar una revisión exhaustiva de los problemas de seguridad, privacidad y falta de interoperabilidad en la gestión de historias clínicas, junto con las regulaciones de salud y protección de datos, mediante análisis bibliográficos y entrevistas a personal de la salud y pacientes.

.

- ***Explorar la tecnología de cadena de bloques en el contexto de la gestión de historias clínicas***: Investigar las capacidades y aplicaciones de la tecnología de cadena de bloques, con un enfoque en su potencial para abordar los desafíos específicos de la gestión de historias clínicas.

- ***Diseñar un sistema de gestión de historias clínicas basado en la cadena de bloques***: Desarrollar un diseño detallado para un sistema que utilice la tecnología de cadena de bloques para la gestión segura y eficiente de historias clínicas, considerando aspectos técnicos y de seguridad.

- ***Crear un prototipo funcional del sistema***: Implementar un prototipo práctico del sistema propuesto, demostrando su viabilidad en la gestión de historias clínicas y su capacidad para garantizar la integridad de los datos mediante la implementación de la tecnología de cadena de bloques.

- ***Evaluar la seguridad y efectividad del sistema***: Realizar pruebas exhaustivas para evaluar la seguridad y efectividad del sistema de cadena de bloques en la gestión de historias clínicas, incluyendo su capacidad para cumplir con regulaciones de protección de datos y garantizar la privacidad del paciente, mediante distintas pruebas técnicas realizadas sobre el prototipo.

- ***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 la aplicación

Abarca la intersección entre la tecnología de cadena de bloques y la atención médica, con un enfoque en cómo la cadena de bloques puede ser una solución

eficiente y segura para la administración de historias clínicas y la mejora de la atención médica en general.

1.4. Justificación

La implementación de la cadena de bloques en la gestión de historias clínicas tiene un potencial significativo para abordar desafíos críticos en la atención médica. Este proyecto busca mejorar la seguridad, la privacidad y la resistencia a la manipulación no autorizada de historias clínicas y al mismo tiempo proporcionando a los pacientes un mayor control sobre su información y facilitando el intercambio de datos entre sistemas de salud.

1.5. Antecedentes

Aunque previos estudios han analizado aplicaciones de cadena de bloques en salud desde diversas perspectivas, listadas por Hasselgren et al. (2020), como privacidad, seguridad, interoperabilidad y democratización del manejo de historias clínicas, pocos se han centrado en escenarios de uso reales en el campo médico.

El sistema propuesto por Zhou et al. (2018) es un sistema de almacenamiento para seguros médicos basado en cadena de bloques, utiliza la propiedad de resistencia a la manipulación de la tecnología de cadena de bloques para proporcionar alta credibilidad a los usuarios.

La tecnología de cadena de bloques posee todas las características necesarias como para convertirse en una herramienta poderosa en la gestión de historias clínicas, abordando problemas de seguridad, interoperabilidad, privacidad, democratización de datos, resistencia a la manipulación, entre otros

Capítulo 2

Marco Teórico

Con el propósito de adquirir una comprensión más completa de este proyecto de grado, a continuación, se expone la fundamentación teórica. Este apartado abarca distintos conceptos y sus respectivas definiciones en los ámbitos de salud, datos, almacenamiento, sistemas, y tecnología.

2.1. Gestión de historias clínicas

Una de las actividades más importantes dentro de una institución de salud es la gestión de historias clínicas de los pacientes, en conjunto con estas los médicos son capaces de proveer una atención de calidad, tomando en cuenta no solo sintomatologías actuales sino históricas descritas a través de exámenes médicos, análisis, prescripciones, diagnósticos, etc., que pueden permitir llegar más eficientemente a un diagnóstico.

2.1.1. Sistemas de salud

Los sistemas de salud son estructuras complejas diseñadas para satisfacer las necesidades de salud de la población, cumpliendo funciones básicas como gobernanza, finanzas, generación de recursos y prestación de servicios, con el objetivo de mejorar la salud y brindar capacidad de respuesta adecuada frente a

eventualidades (Díaz, 2015). Las instituciones de salud, al igual que las empresas, emplean recursos para atender demandas, pudiendo tener fines lucrativos o de provisión de servicios de calidad. La atención médica abarca diagnósticas, tratamientos y prevención de enfermedades físicas y mentales, buscando mejorar el bienestar y recuperación de los pacientes.

2.1.2. Interoperabilidad de sistemas de salud

Gaynor (2014) define la interoperabilidad de los sistemas de salud como la habilidad de distintas aplicaciones, sistemas y dispositivos de atención médica para compartir y usar datos eficientemente de forma coordinada. Requiere la combinación de varios estándares, tecnologías y protocolos para asegurar que la información de salud relevante pueda transmitirse y entenderse en diferentes plataformas y organizaciones dentro del sector de la salud. Conseguir la interoperabilidad de los sistemas de salud es esencial para mejorar la atención al paciente, mejorar el juicio clínico y optimizar los procesos de atención médica dentro de una región o localidad.

2.1.3. 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.

2.1.4. 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 únicamente 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.

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.

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.

2.1.7. Historias clínicas

Las historias clínicas son registros digitales o físicos de los datos de salud de un individuo, según Evans (2016). Estos registros son accesibles, confidenciales, seguros y aceptables tanto para los médicos como para los pacientes. Además, están integrados con otra información no específica del paciente. Las historias clínicas electrónicas prometen resolver los problemas asociados con la conservación de registros en papel y tienen un gran potencial en áreas como los ensayos clínicos y el apoyo a la toma de decisiones. Sin embargo, para que los sistemas de historias clínicas se integren más allá de los entornos intranet dentro de sistemas o prácticas de salud individuales y se conecten con recursos regionales, nacionales e internacionales a través de Internet, hay ciertos temas que deben abordarse y requisitos que deben cumplirse.

2.2. Cadena de bloques

2.2.1 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.

2.2.2 Cadena de bloques

Según Nakamoto (2009), la cadena de bloques, conocida en inglés como *blockchain*, es una estructura de datos que se utiliza para almacenar información de manera segura y descentralizada. La información se agrupa en conjuntos llamados bloques, a los que se les añade metadatos relativos a otro bloque anterior de la cadena relativo a una línea temporal que permite el orden inmutable dentro de una cadena a través de grandes cálculos criptográficos. Debido a esto la información contenida en un bloque solo puede ser repudiada o editada modificando todos los bloques anteriores, permitiendo su aplicación en un entorno distribuido de manera que la estructura de datos de la cadena de bloques puede funcionar como una base de datos pública no relacional que contenga un histórico irrefutable de información.

Refiriéndose a la especificación original de la cadena de bloques de Bitcoin, su funcionamiento se describe de la siguiente manera:

- Los activos digitales almacenados y transferidos en la cadena de bloques se representan como una cadena de firmas digitales. Cada propietario puede transferir sus activos a otro participante de la red firmando un *hash* de la transacción previa junto con la clave pública del nuevo propietario, añadiéndolos al final del activo. El receptor puede verificar las firmas para confirmar la cadena de propiedad. El problema de los gastos dobles, que típicamente se resuelve con una autoridad central, aquí se soluciona

haciendo públicas todas las transacciones, permitiendo que todos los participantes acuerden un único historial del orden de las transacciones, validando la primera en ser registrada.

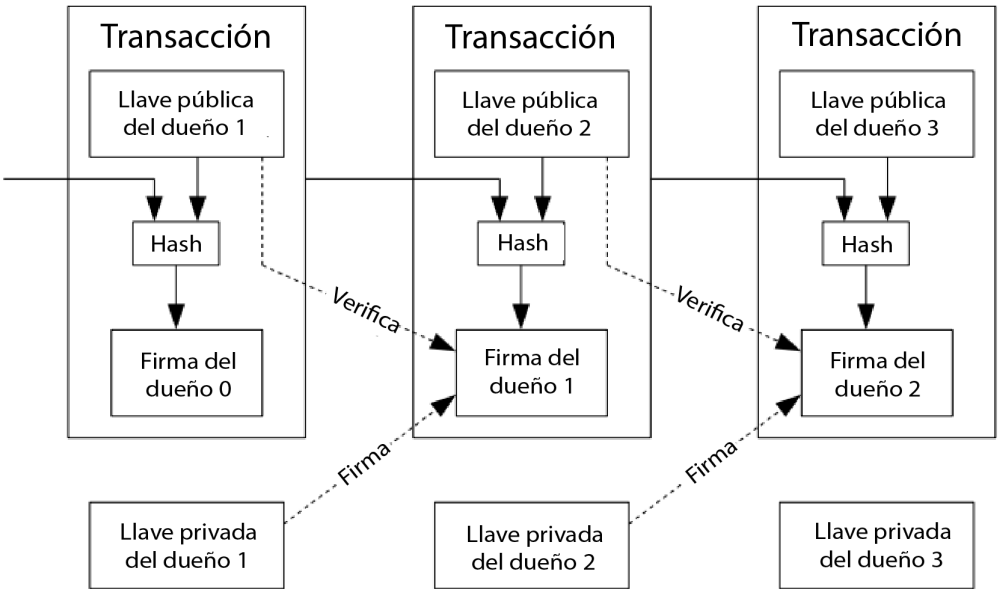


Figura 2.1: Transacciones en cadena de bloques.

Para implementar esta solución se necesita un servidor de marcas de tiempo, que toma el *hash* de un bloque y lo publica ampliamente a todos los participantes. La marca de tiempo demuestra que los datos debían existir en ese momento para ser asociados con el *hash* del bloque. Cada marca de tiempo incluye la anterior en su *hash*, creando una cadena que refuerza las marcas de tiempo previas.

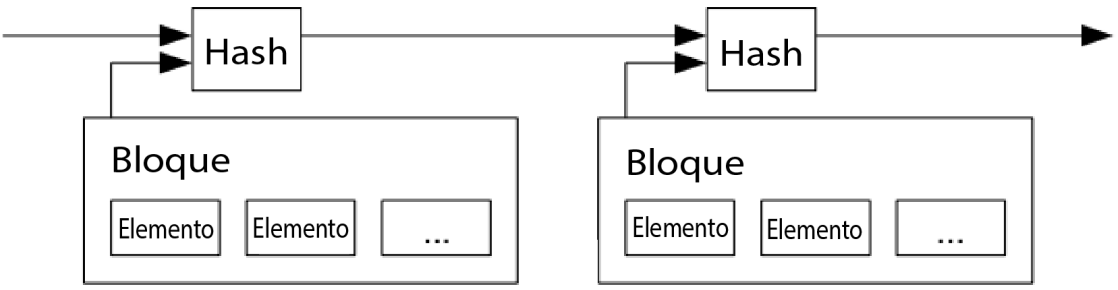


Figura 2.2: Servidor de marcas de tiempo.

En un entorno distribuido con participantes iguales, se utiliza una prueba de trabajo para determinar quién aprueba el siguiente bloque de la cadena. La prueba de trabajo implica encontrar un valor que, al ser "hasheado" con un algoritmo como el SHA-256, produzca un *hash* que comience con un cierto número de bits en cero. El trabajo necesario es exponencial con relación al número de bits en cero requeridos y puede ser verificado con un solo *hash*.

En esta red de marcas de tiempo, la prueba de trabajo se implementa incrementando un *nonce* (número usado una vez) en el bloque hasta encontrar un valor que haga que el *hash* del bloque tenga los bits en cero necesarios. Una vez que la CPU ha realizado el esfuerzo para satisfacer la prueba de trabajo, el bloque no puede ser modificado sin rehacer todo el trabajo. A medida que se añaden bloques, cambiar uno implicaría rehacer todos los bloques anteriores.

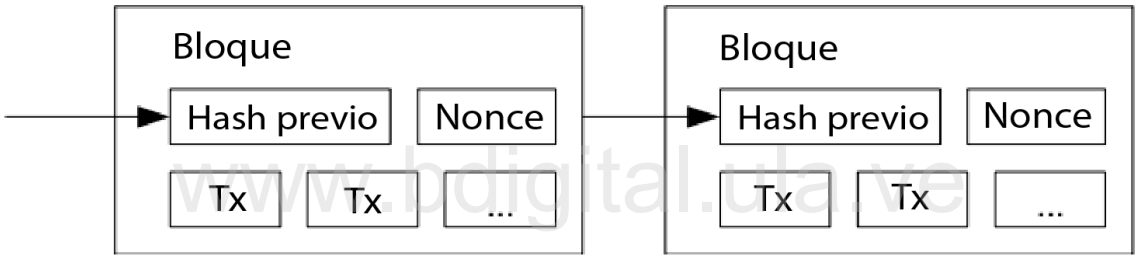


Figura 2.3: Cadena de bloques.

Al resumir esto se puede ver que estas redes cumplen con que:

- Cada nodo recopila nuevas transacciones en un bloque.
- Cuando ocurre una nueva transacción en un bloque, esta se transmite a todos los nodos.
- Cada nodo trabaja para encontrar una prueba de trabajo difícil para su bloque.
- Cuando un nodo encuentra una prueba de trabajo, transmite el bloque a todos los nodos.
- Al recibir múltiples bloques simultáneamente, el nodo siempre considera la cadena más larga como la correcta y sigue trabajando en extenderla.
- Si dos nodos envían diferentes versiones del siguiente bloque, el nodo receptor sigue trabajando en el primero recibido, pero guarda el otro en

caso de que la cadena se extienda sobre él, haciendo que los nodos que estaban trabajando en el otro bloque cambien a la cadena más larga.
- Los nodos aceptan el bloque sólo si todas las transacciones que contiene son válidas y no se han gastado previamente.
- Los nodos indican su aceptación del bloque trabajando en la creación del siguiente bloque en esa cadena, utilizando el *hash* del bloque aceptado como *hash* previo.

Se conocen tres tipos, según Paul (2021), de cadenas de bloques:
- ***Cadena de bloques pública***: En este tipo de cadena de bloques, cualquier interesado tiene la capacidad de leer y enviar transacciones.
- ***Cadena de bloques privada***: En una cadena de bloques privada, solo una organización o las organizaciones subsidiarias dentro del mismo grupo pueden leer y enviar transacciones.
- ***Cadena de bloques híbrida***: Esta categoría combina características de las cadenas de bloques públicas y privadas para facilitar las transacciones. Su principal ventaja radica en su alta personalización.
- ***Cadena de bloques comunitaria o de consorcio***: En este tipo de cadena de bloques, varios grupos de organizaciones forman un consorcio. Los miembros de dicho consorcio tienen la capacidad de enviar transacciones y leer datos transaccionales.

2.3. Antecedentes

2.3.1. Cadena de bloques en la atención médica y las ciencias de la salud: una revisión de alcance

La investigación realizada por Hasselgren et al. (2020) hace una revisión sistemática de la utilización de la tecnología cadena de bloques en el ámbito de la salud, con el objetivo de evaluar y sintetizar estudios publicados que han implementado o propuesto el uso de cadena de bloques para mejorar procesos y servicios en la atención médica, las ciencias de la salud y la educación sanitaria. A través de una búsqueda estructurada en bases de datos bibliográficas relevantes, se identificaron 39 publicaciones que cumplen con los criterios de inclusión. Los resultados indican que las áreas más enfocadas en el uso de cadena de bloques son los registros electrónicos de salud y los registros personales de salud, *EHR* y *PHR* según sus siglas en inglés. Los problemas de

control de acceso, interoperabilidad, procedencia e integridad de los datos son abordados con la implementación de cadena de bloques, con Ethereum y Hyperledger Fabric siendo las plataformas más utilizadas en este campo.

La investigación demuestra que el uso de cadena de bloques en la salud se está consolidando como un campo académico con un aumento significativo en la cantidad y calidad de las publicaciones. Aunque se han explorado casos de uso en *EHR*, *PHR* y sistemas de ensayos clínicos, otros dominios de los sistemas de información sanitaria, como infraestructuras de conocimiento, sistemas de archivo y comunicación de imágenes, servicios de diagnóstico automatizados, sistemas administrativos, gestión de la salud poblacional y cadenas de suministro farmacéuticas, están subexplotados. El estudio recomienda que la agenda de investigación debe ampliarse para abordar estas áreas específicas y desarrollar soluciones basadas en cadenas de bloques que preserven la confianza y mitiguen amenazas tanto internas como externas en el sector de la salud.

2.3.2. MIStore: un sistema de almacenamiento de seguros médicos basado en blockchain

Realizado por Zhou et al. (2018) este es un sistema de almacenamiento de seguros médicos basado en cadenas de bloques. *MIStore* utiliza la resistencia a la manipulación de cadenas de bloques para proporcionar alta credibilidad y seguridad a los usuarios. En el sistema básico, participan un hospital, un paciente, una compañía de seguros y múltiples servidores. El hospital almacena los datos de gastos del paciente en la cadena de bloques, protegidos por servidores que pueden realizar cálculos homomórficos sobre los datos sin aprender nada de ellos. El sistema permite a las compañías de seguros obtener sumas de datos de gastos del paciente con la ayuda de una cantidad mínima de servidores honestos. *MIStore* también implementa el *Protocolo de Umbral(t,n)*, donde al menos *t* servidores deben colaborar para que la compañía de seguros obtenga los datos necesarios, manteniendo la confidencialidad y veracidad de los datos.

La implementación de *MIStore* utilizó red de cadena de bloques de Ethereum y presenta una evaluación de rendimiento. El sistema aprovecha las propiedades de la descentralización, resistencia a la manipulación y verificación eficiente de datos. Las transacciones son verificadas por nodos de registro antes de ser

incluidas en la cadena de bloques, reduciendo significativamente la carga de verificación para los usuarios. La combinación de cálculos homomórficos eficientes y la propiedad de umbral garantiza que los datos del paciente permanezcan confidenciales mientras permiten a las compañías de seguros realizar las consultas necesarias.

Capítulo 3

Marco Metodológico

De acuerdo con Kothari (2004), existen 8 tipos básicos de metodología de investigación: Descriptiva, Analítica, Aplicada, Fundamental, Cuantitativa, Cualitativa, Conceptual y Empírica. Dentro de este proyecto de grado vamos a aplicar un tipo de investigación aplicada, ya que esta busca soluciones inmediatas para problemas concretos en la sociedad o en organizaciones.

La investigación adopta un enfoque explicativo al centrarse en la elucidación de los elementos más significativos del tema, partiendo de una concepción general. Utiliza la investigación como una herramienta para orientar posibles temas futuros, sin buscar evidencia concluyente, sino más bien buscando comprender de manera más eficaz el problema en cuestión.

3.1. Metodología de investigación

Las etapas de esta metodología se detallan a continuación:

- ***Empatizar*** - Comprender las necesidades de los usuarios:
  En esta fase, el objetivo es adquirir un entendimiento profundo de las necesidades y deseos de los usuarios en relación con el sistema de gestión de historias clínicas de los pacientes en instituciones de salud.
  Para lograrlo, se debe:
  - Llevar a cabo entrevistas con pacientes y profesionales de la salud para comprender sus necesidades, desafíos actuales y expectativas en relación con la gestión de historias clínicas de los pacientes.
  - Observar los entornos en los cuales los usuarios interactúan con el sistema.

- ***Definir*** - Identificar los problemas a resolver:
  En esta fase, se definen los problemas que el sistema debe abordar, lo cual implica:
  - Analizar los datos recopilados durante la fase de Empatía con el fin de identificar los problemas clave que el sistema debe resolver.

- ***Idear*** - Generar soluciones innovadoras:
  Esta fase fomenta la generación de ideas creativas para resolver los problemas identificados. Las actividades comprenden:
  - Realizar sesiones de lluvia de ideas con un equipo interdisciplinario para concebir soluciones innovadoras.
  - Evaluar las ideas generadas para seleccionar las más adecuadas y prometedoras.

- ***Prototipar*** - Desarrollar prototipos y obtener retroalimentación:
  En esta etapa, se confeccionan prototipos de las soluciones elegidas en la fase de “ideación”:
  - Elaborar prototipos de las soluciones identificadas como las más prometedoras.
  - Probar los prototipos con los usuarios para obtener retroalimentación.

- Iterar los prototipos con base en la retroalimentación de los usuarios.

- ***Poner a prueba*** - Implementación en el mundo real y retroalimentación: En esta fase, se procede a implementar las soluciones en el entorno real de las instituciones de salud:
  - Implementar las soluciones en un entorno de atención médica real.
  - Evaluar las soluciones en el entorno real con la participación de los usuarios.
  - Recolectar los comentarios y la retroalimentación de los usuarios acerca de la eficacia y utilidad de las soluciones implementadas.

El marco metodológico presentado se adaptará según las necesidades específicas del proyecto y contribuirá a mejorar la calidad de la atención médica, con un enfoque en la privacidad de los pacientes y la promoción de la interoperabilidad de los sistemas de salud a través de la tecnología de cadena de bloques. La aplicación del Enfoque de Diseño se enfoca en la resolución de problemas complejos y en la satisfacción de las necesidades de los usuarios, involucrándose en todas las etapas del proceso.

3.1.1. Población

Al aludir a la población, nos referimos al conjunto de individuos dentro del ámbito abordado en el problema de investigación. Según Kothari (2004), una población se caracteriza por sus atributos definitorios; en consecuencia, el grupo de elementos que comparte dichos atributos se identifica como población o universo. En este proyecto, la población a tener en cuenta será el Instituto Autónomo Hospital Universitario de Los Andes, más conocido como IAHULA.

3.1.2. Muestra

Una muestra se concibe como "una porción o fragmento del total de unidades de observación o análisis que ha sido seleccionado". En este estudio, la muestra estará compuesta por tres profesionales de la salud, tres pacientes y los procesos clínicos y administrativos asociados al IAHULA.

3.1.3. Tipo de Muestra

En este proyecto, se empleará un tipo de muestra intencional, donde los elementos son seleccionados con base en criterios o juicios predefinidos por el investigador, siendo estos profesionales de la salud vinculados al IAHULA.

3.2. Método de recolección de datos

Una vez definido el problema de investigación, se inicia la recopilación de datos con el objetivo de obtener información sobre el fenómeno o problema de interés. En este proyecto se empleará el método de entrevistas, el cual se basa en estímulos orales-verbales, y las respuestas se expresan en términos de respuestas orales-verbales. Este método puede implementarse a través de entrevistas personales y, de ser posible, mediante entrevistas telefónicas o videoconferencias, dependiendo de la disponibilidad del individuo entrevistado.

3.3. Procesamiento y codificación de datos

Una vez recopilados, los datos deben procesarse y analizarse de acuerdo con el esquema establecido durante la elaboración del plan de investigación. Este proceso es crucial para un estudio científico y garantiza la disponibilidad de datos relevantes para realizar comparaciones y análisis contemplados. Técnicamente, el procesamiento implica 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 búsqueda de patrones de relación entre los grupos de datos. 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 diferencia entre procesamiento y análisis, aunque algunos expertos opinan que ambas actividades están estrechamente relacionadas.

3.4. Metodología para el desarrollo de la solución

Una sólida metodología ofrece una estructura y un enfoque sistemático para enfrentar los desafíos complejos vinculados a la gestión de historias clínicas en el ámbito de la salud. Así mismo, dirige la recolección y el análisis de información, la creación de soluciones innovadoras y la valoración de su

eficacia. La atención médica se trata de un servicio vital en el cual los pacientes deben experimentar una atención eficaz y comprensiva. Por esta razón, el Enfoque de Diseño se presenta como una metodología que se adapta a los objetivos de este proyecto, ya que pone por delante la relevancia de la empatía hacia los usuarios, la originalidad y la constante iteración, tal como se expone en Anderson (2022). Esto garantiza que la solución final sea auténticamente efectiva y responda a las necesidades de los pacientes y profesionales de la salud, contribuyendo, de este modo, a mejorar la atención médica y la confidencialidad de los pacientes.

La metodología de gestión de proyecto empleada fue Scrum, que según Schwaber (2015), es un marco de referencia para el desarrollo y mantenimiento de productos que facilita la estructuración y gestión del trabajo a través de valores, principios y prácticas. Scrum fomenta que los equipos aprendan de sus experiencias y se autoorganicen al abordar problemas. Aunque se clasifica como un marco ágil de gestión de proyectos, Scrum integra reuniones, herramientas y roles coordinados para estructurar y gestionar el trabajo. Este enfoque desempeñó un papel crucial en la elección para el desarrollo del proyecto de grado.

Las fases aplicadas de esta metodología fueron:

3.4.1. Planificación

En esta fase, se identifican las tareas clave y se recopila información detallada sobre el proyecto. Un representante del cliente elabora el *product backlog*, 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.

3.4.2. Ejecución

La fase de ejecución, denominado 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 el caso de este proyecto de grado, esta fase incluyó la implementación de las funcionalidades.

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 manifestó cada vez que se completaba una funcionalidad, mediante reuniones que discutían avances, mejoras y ajustes de manera iterativa.

3.4.4. Metodología Ágil y Scrum

Según Montilva et al. (2021), una metodología ágil es un método para el desarrollo de software que se fundamenta en iteración, cooperación y adaptación. Las metodologías ágiles se distinguen por los siguientes principios:

- Personas y comunicación sobre procesos y herramientas.
- Software operativo sobre documentación detallada.
- Colaboración con el cliente sobre contrato fijo.
- Adaptación al cambio sobre cumplir un plan.
- Entrega rápida y frecuente sobre un lanzamiento demorado.

Las metodologías ágiles implican una serie de elementos que garantizan la comunicación, interacción y productividad, de modo que el producto funcional se logre lo más pronto posible.

Scrum, según Schwaber (2015), es un marco de trabajo ágil que se puede utilizar 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. En Scrum hay tres roles principales:

- ***Dueño del producto**:* Responsable de definir el producto y las prioridades del desarrollo.
- ***Maestro del Scrum***: Responsable de facilitar el proceso de Scrum y de garantizar que el equipo cumpla con los objetivos.
- ***Equipo de desarrollo***: Responsable de desarrollar el software.

De igual manera, en Scrum hay cinco eventos clave:

- ***Planificación del sprint***: Definición del trabajo a realizar en el sprint.
- ***Scrum diario***: Reunión diaria para discutir el progreso y los obstáculos.
- ***Revisión de sprint***: Presentación del trabajo completado al product owner y a los interesados.
- ***Retrospectiva del sprint***: Reflexión sobre el sprint y mejoras para el siguiente.
- ***Meta del sprint***: Objetivo del sprint.

En Scrum hay tres artefactos principales:

- ***Backlog del producto***: Lista de todos los requisitos del producto.
- ***Backlog del sprint***: Lista de tareas que se realizarán en el sprint.
- ***Incremento***: Código, documentación y otros artefactos completados en el sprint.

Scrum ofrece una serie de ventajas, incluyendo mayor flexibilidad y adaptabilidad, mejor colaboración entre las partes interesadas, entrega de software de mayor calidad y reducción de los costos de desarrollo. Sin embargo, también presenta algunas desventajas, como la dificultad de implementación comparada con metodologías tradicionales, mayor compromiso requerido de los miembros del equipo y menor adecuación para proyectos complejos o de gran envergadura.

Para la ejecución de este proyecto, se adoptará un enfoque ágil utilizando la metodología Scrum. Esto implica el uso de sprints o incrementos de dos semanas de duración. La elección de esta metodología responde a la necesidad de mantener un ritmo constante de desarrollo y adaptación. La duración bimensual de los sprints 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. Las actividades de los sprints serán las siguientes:

Sprint 1

- Elaboración del *backlog* del producto.
- Planeamiento del sprint.
- Recolección de datos.

- Análisis de datos.
- Selección de la metodología de desarrollo de software.
- Revisión bibliográfica sobre la metodología de desarrollo de software.
- Revisión del sprint.
- Retrospectiva del sprint.

Sprint 2

- Planeamiento del sprint.
- Gestión inicial del proyecto según la metodología de desarrollo de software.
- Redacción de la lista de interesados.
- Elaboración de la descripción del problema.
- Elaboración de la visión del producto.
- Planificación inicial del proyecto de desarrollo.
- Análisis y especificación inicial de requisitos.
- Elaboración del informe de descripción del contexto del sistema.
- Elaboración de la lista del producto (product backlog) con requisitos e historias de usuarios.
- Documentación de la definición de requisitos.
- Revisión del sprint.
- Retrospectiva del sprint.

Sprint 3

- Planeamiento del sprint.
- Modelado funcional y estructural de los requisitos.
- Elaboración del documento de requisitos.
- Diseño arquitectónico inicial de la aplicación.
- Elaboración de la lista de requisitos arquitectónicos.
- Elaboración del diagrama estructural de la arquitectura inicial de la aplicación.
- Revisión del sprint.
- Retrospectiva del sprint.

Sprint 4

- Planeamiento del sprint.
- Diseño de la interfaz gráfica de usuario.

- Elaboración del documento inicial de diseño.
- Elaboración del plan de entregas e incrementos.
- Actualización del plan del proyecto.
- Desarrollo y pruebas del incremento 1.
- Entrega del incremento 1.
- Revisión del sprint.
- Retrospectiva del sprint.

Sprint 5

- Planeamiento del sprint.
- Desarrollo y pruebas del incremento 2.
- Entrega del incremento 2.
- Revisión del sprint.
- Retrospectiva del sprint.

Sprint 6

- Planeamiento del sprint.
- Desarrollo y pruebas del incremento 3.
- Entrega del incremento 3.
- Revisión del sprint.
- Retrospectiva del sprint.

Sprint 7

- Planeamiento del sprint.
- Desarrollo y pruebas del incremento 4.
- Entrega del incremento 4.
- Revisión del sprint.
- Refinamiento de los productos de la metodología de desarrollo de software.

Este enfoque permitirá cumplir con los objetivos del proyecto de manera eficiente y estructurada, asegurando la calidad y el cumplimiento de los plazos establecidos.

3.5. Metodología para el desarrollo de software

Según Montilva et al. (2021), Blue Watch es un método para el desarrollo y operación de sistemas de software. Combina diferentes aspectos de los enfoques ágiles y disciplinados de la Ingeniería del Software, proporcionando una guía práctica adaptable a las necesidades particulares de sus usuarios. Este método actúa como un marco metodológico que proporciona una estructura conceptual y de trabajo para resolver problemas, desarrollar soluciones o prestar servicios, ayudando a los equipos de desarrollo y operación de software a crear sus propios métodos mediante la adaptación de un marco completo. El método se compone de 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. Contiene plantillas que facilitan la documentación de informes técnicos y de gestión necesarios para desarrollar el sistema.
- ***Modelo de actores***: Identifica y describe los roles necesarios 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 la secuencia en que los procesos técnicos, de gestión y de soporte del desarrollo de aplicaciones o sistemas de software deben ejecutarse.

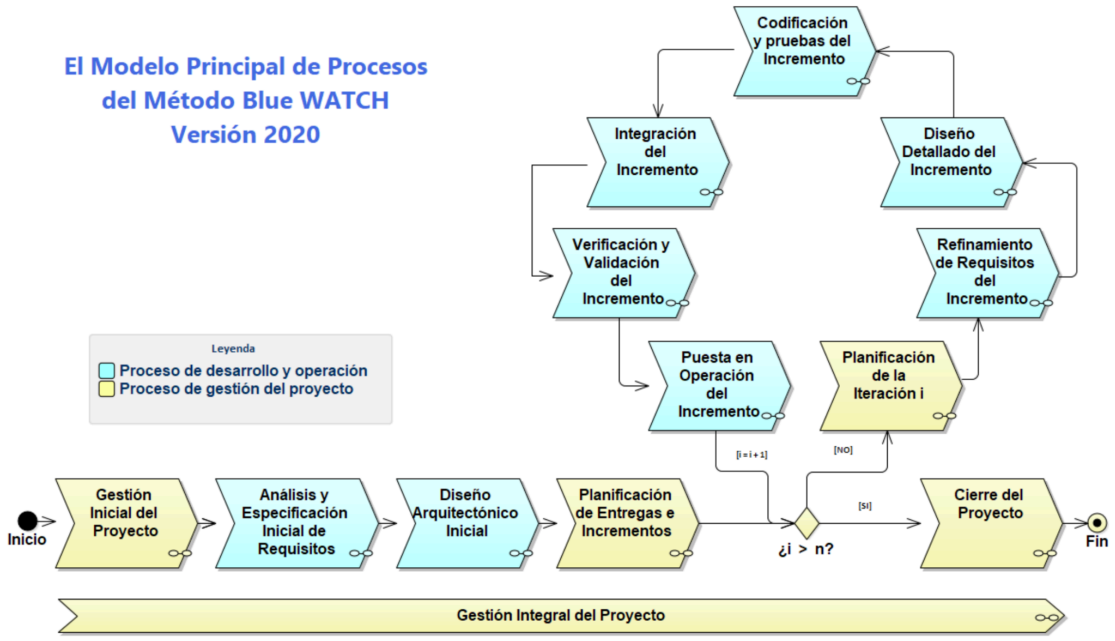


Figura 3.1: Modelo principal de procesos de Blue Watch.

3.5.1. Gestión inicial del proyecto

En esta etapa, se designa al líder del proyecto, que en este caso será el profesor Jonás Montilva, tutor del proyecto que guiará y supervisará las decisiones durante la planificación, gestión, desarrollo e implementación de la solución propuesta. A continuación, se describen algunos aspectos fundamentales para el inicio del proyecto.

3.5.2. Lista de interesados

Los interesados en el proyecto son personas u organizaciones que se beneficiarán del desarrollo del sistema propuesto:

- Médicos en instituciones de salud pública o privada, o independientes en consultorios registrados en el ministerio de salud.
- Pacientes de los médicos mencionados.
- Enfermeros y enfermeras en instituciones de salud pública o privada, o independientes en consultorios registrados en el ministerio de salud.

- Instituciones de salud públicas, privadas o consultorios independientes registrados en el ministerio de salud.
- Ministerio de salud.

3.5.3. Descripción del problema

En el Instituto Autónomo Hospital Universitario de Los Andes y otras instituciones de salud, la gestión de historias clínicas se realiza manualmente, utilizando papel archivado en un sistema físico centralizado. Este método presenta desafíos y riesgos significativos que afectan la eficiencia y seguridad de la atención médica:

- El proceso manual de gestión de historias clínicas es propenso a errores humanos, lo que puede resultar en confusiones en la atención y pérdida de información crítica.
- 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 tratamientos y decisiones médicas.
- La seguridad y la integridad de los datos son preocupantes. El almacenamiento en papel es vulnerable a daños físicos y carece de medidas efectivas de respaldo y recuperación en caso de pérdida de información, poniendo en riesgo la continuidad de la atención y la privacidad de los pacientes.
- Los pacientes no reciben copias de sus historias médicas, lo que dificulta su participación en su propio cuidado de salud y limita su capacidad para compartir información relevante con otros proveedores.

La implementación de un sistema de gestión de historias clínicas basado en tecnología de cadena de bloques se presenta como una solución integral a estos desafíos. Esta tecnología permitirá crear registros de salud electrónicos precisos y actualizados, accesibles desde cualquier punto dentro y fuera de la institución de salud, con medidas sólidas de seguridad y respaldo de datos. Además, facilitará 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. Esta implementación mejorará la eficiencia y calidad de la atención médica en el IAHULA, sentando las bases para una práctica médica más segura, transparente y centrada en el paciente.

3.5.4. Visión del producto

Usuarios potenciales del sistema

- ***Personal médico***: Médicos y enfermeras que necesitan acceder a la información del paciente para proporcionar atención adecuada y oportuna.
- ***Personal administrativo***: Personal encargado de gestionar la admisión de médicos, enfermeros y enfermeras a la institución.
- ***Pacientes***: Individuos que buscan acceder a su información médica y participar activamente en su cuidado de salud.

Necesidades de los usuarios

- ***Personal médico***: Acceso rápido y fácil a la información del paciente, así como manejo de su propia información asociada a la institución.
- ***Personal administrativo***: Eficiencia en la gestión de identificación del personal de salud asociado a la institución y seguimiento de la admisión de pacientes.
- ***Pacientes***: Facilidad en el registro y acceso de datos médicos.

Objetivos del sistema

- ***Descentralizar la Información del paciente***: Crear una plataforma interinstitucional que almacene y gestione de manera segura las historias clínicas de todos los pacientes registrados en el sistema.
- ***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.

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.

3.5.5. Plan inicial del proyecto

Se plantea desarrollar un sistema de gestión de historias clínicas en instituciones de salud que implemente mecanismos de descentralización, privacidad y seguridad, mejorando la calidad de la atención médica y fomentando la interoperabilidad entre instituciones de salud en beneficio de pacientes y profesionales médicos utilizando tecnología de cadena de bloques.

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 e integridad de la información del paciente y demás usuarios.
- Desarrollo de funciones de búsqueda avanzada y filtros para acceder rápidamente a la información del paciente.

Selección del personal de desarrollo

El personal de desarrollo de este proyecto consta de 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.

Plataformas y herramientas de desarrollo

Considerando que el resultado de este proyecto es un producto mínimo viable, las plataformas de hardware a utilizar constan de las tres computadoras

personales de los ingenieros a cargo de su desarrollo, con sistemas operativos Windows, Linux y MacOS.

Los lenguajes de programación escogidos para la construcción de la solución son Golang 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.

En cuanto a la base de datos, se plantea el uso de PostgreSQL para almacenar datos del sistema de manera segura y confiable, acotando que la dirección hacia los documentos de los registros de salud se almacena encriptada en la cadena de bloques implementada con Hyperledger Fabric, el cual es un *framework* de cadena de bloques de código abierto mantenido por la Fundación Linux.

Los documentos asociados a cada uno de los registros de salud serían almacenados en un servicio externo como Google Drive, pero debido a carencia de ingenieros dentro del proyecto, es posible que sean almacenados en el mismo servidor al que se despliegue el sistema propuesto

Capítulo 4

Análisis y definición de requisitos

Este proceso implica identificar, documentar y gestionar las necesidades y expectativas de las partes interesadas respecto a lo que el sistema o producto debe lograr. La importancia de este análisis radica en que proporciona una comprensión clara y detallada de lo que se debe construir, sirviendo como base para la planificación, el diseño y la implementación del proyecto. Un adecuado análisis y definición de requisitos aseguran que todos los involucrados tengan una visión compartida y precisa de los objetivos del proyecto, lo cual minimiza el riesgo de errores y desviaciones en el desarrollo, optimizando el uso de recursos y mejorando la calidad del producto final.

4.1. Análisis del contexto del sistema

Los sistemas de salud buscan la forma más eficiente y efectiva para proveer un servicio de calidad a sus usuarios. A nivel gubernamental, los departamentos pertinentes necesitan regular todas las instituciones de salud, asegurándose de que practiquen de forma segura y se adhieran a los códigos establecidos. Por su parte, las instituciones de salud buscan dar el mejor servicio posible a sus usuarios y administrar su personal médico en relación con los mismos, así como la rotación de los médicos y enfermeros. También es vital la capacidad de transferencia de información de los pacientes entre las instituciones de salud. Los médicos y enfermeros deben encargarse de que todos los usuarios sean atendidos adecuadamente, realizar todos los exámenes necesarios, escribir las prescripciones requeridas y administrar los medicamentos de ser necesario.

4.1.1. Identificación del contexto del sistema

Los sistemas de salud en Venezuela son el contexto sobre el que se hizo el estudio del sistema.

Identificación de los objetivos del contexto del sistema

- Proveer un servicio eficaz y de calidad a los usuarios.

Identificación de los principales conceptos presentes en el contexto

- Pacientes.
- Médicos.
- Instituciones de salud.
- Ente gubernamental.
- Historias clínicas.

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
  - Recibir 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.
- **Personal del 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***
  - Tener fácil acceso a sus historias clínicas y
  - Poder compartirlas con el personal de salud que sea necesario,
  - Poder compartirlas con instituciones de salud.
  - Que sus historias clínicas estén seguros.
  - Tener absoluto control sobre quién tiene acceso a sus historias clínicas.
- ***Médicos***
  - Fácil acceso a las historias clínicas de sus pacientes.
  - Formato digital de las historias clínicas de sus pacientes.
  - Disponibilidad de las historias clínicas de los pacientes.

- ***Enfermeras/enfermeros***:
    - Acceso a la ubicación intrahospitalaria de los pacientes al estar hospitalizados y
    - Acceso a las órdenes de los médicos para poder realizar sus actividades laborales.
- ***Personal administrativo de la institución de salud***:
    - Gestionar el acceso a la plataforma de los médicos y enfermeras asociados a dicha institución.
- ***Personal del ente gubernamental***:
- Gestionar el acceso a la plataforma de las instituciones legalmente constituidas dentro de su red.

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

- Gestión de acceso a las historias clínicas de los pacientes.
- Gestión de las historias clínicas de los pacientes.

Los procesos de identidad, verificación de pertenencia, gestión de información profesional y personal serán abordados en el proyecto complementario a este "Desarrollo de un Sistema de Gestión de Identidad de Pacientes en Instituciones de Salud mediante Tecnología de Cadena de Bloques".

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, estudios, 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 se definen como:

Gestión de historias clínicas

- ***Almacenamiento Seguro***
  - El sistema debe almacenar de forma segura las historias clínicas de los pacientes en la cadena de bloques, garantizando su confidencialidad, integridad e inmutabilidad.
- ***Acceso Controlado***
  - Los pacientes deben tener acceso controlado a sus propias historias clínicas, con la capacidad de visualizarlos, compartirlos con proveedores de atención médica autorizados y otorgar consentimientos para su uso.
- ***Interoperabilidad***
  - El sistema debe permitir la interoperabilidad con otras instituciones de salud, facilitando el intercambio de registros de salud.

Gestión de Consentimientos

- ***Obtención de Consentimiento Informado***
  - El sistema debe facilitar la obtención 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.

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, control de acceso y auditoría de seguridad.
- ***Cumplimiento de Normativas***
  - El sistema debe cumplir con las normativas de privacidad de datos y protección de la salud aplicables en cada jurisdicción.
- ***Gestión de Riesgos***
  - El sistema debe implementar un proceso continuo de gestión de riesgos para identificar, evaluar y mitigar potenciales amenazas a la seguridad de la información.
- ***Escalabilidad***
  - El sistema debe diseñarse con una arquitectura escalable para adaptarse al crecimiento del número de usuarios y la cantidad de datos.

4.2.3. Product Backlog Agrupado por Historias de Usuario Similares

A continuación se van a mencionar todas las historias de usuario necesarias para el funcionamiento del sistema, pero se van a resaltar aquellas que serán abordadas principalmente en este proyecto, mientras que las demás son abordadas en profundidad por el proyecto complementario.

Acceso a historias clínicas

- ***Acceso a historias clínicas de Pacientes***
    - El sistema debe mostrar una lista de todos las historias clínicas del paciente.
    - El sistema debe permitir la visualización de los detalles de cada registro médico.
    - El sistema debe permitir a los pacientes seleccionar y dar acceso a médicos específicos.
    - El sistema debe notificar al médico sobre el acceso otorgado.

- ***Acceso a Historias Clínicas por médicos***
    - El sistema debe permitir el acceso a las historias clínicas de los pacientes.

- ***Visualización de Órdenes Médicas de Pacientes Hospitalizados por Enfermeros***
    - El sistema debe mostrar únicamente las órdenes médicas de los pacientes hospitalizados a los enfermeros.

Adición de Registro de Salud a la Historia Clínica de Un Paciente

- El sistema debe permitir agregar nuevas evoluciones al historial del paciente.
- El sistema debe permitir agregar nuevas órdenes médicas al historial del paciente.
- El sistema debe permitir agregar nuevos análisis al historial del paciente.

Desvinculación de historias clínica de un paciente

- El sistema debe permitirle a un paciente desvincular sus registros de salud de los registros de información del sistema.

Filtrar y buscar información

- ***Filtrar y buscar información de los médicos***
    - El sistema debe permitir la búsqueda de médicos por especialidad.
    - El sistema debe permitir filtrar médicos.
- ***Filtrar y buscar información de los pacientes***
    - El sistema debe permitir buscar pacientes en el directorio del médico.

- El sistema debe permitir filtrar pacientes por criterios como edad, sexo, estatus y fecha de consulta.
- El sistema debe permitir buscar pacientes que tengan registros de salud de una especialidad específica en sus historias clínicas.

Gestión de solicitudes

- ***Gestión de solicitudes para pacientes***
  - El sistema debe permitir listar todas las solicitudes de acceso a las historias clínicas del paciente por parte de médicos.
  - El sistema debe permitir listar a los médicos cuya solicitud de acceso fue aceptada.
  - El sistema debe permitir conceder o denegar la solicitud de acceso a las historias clínicas de los médicos.
  - El sistema debe permitir revocar el acceso a las historias clínicas dado a un médico.

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

- *Gestión de solicitudes para usuarios de gobierno*

Registro de usuarios

- *Registro de pacientes*
- *Registro de médicos*
- *Registro de enfermeros y enfermeras*
- *Registro de usuarios de institución de salud*

Inicio de sesión de usuarios

- *Inicio de sesión de pacientes*
- *Inicio de sesión de médicos*
- *Inicio de sesión de enfermeros y enfermeras*
- *Inicio de sesión de usuarios de institución de salud*
- *Inicio de sesión de usuarios de gobierno*

Acceso a información personal e institucional

- *Acceso a información personal de los médicos*
- *Acceso a información personal de los enfermeros y enfermeras*
- *Acceso a información institucional de las instituciones de salud*

Concesión, denegación y revocación de acceso a la información personal

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

Edición de datos personales y profesionales

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

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

- El sistema debe validar la información ingresada para evitar duplicados.
- La seguridad de los datos de inicio de sesión debe garantizarse mediante encriptación.
- La validación de la información debe ser rigurosa para evitar errores.
- La integridad y consistencia de los datos deben mantenerse después de cada edición o cambio.

Interfaz de Usuario

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

Rendimiento

- 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

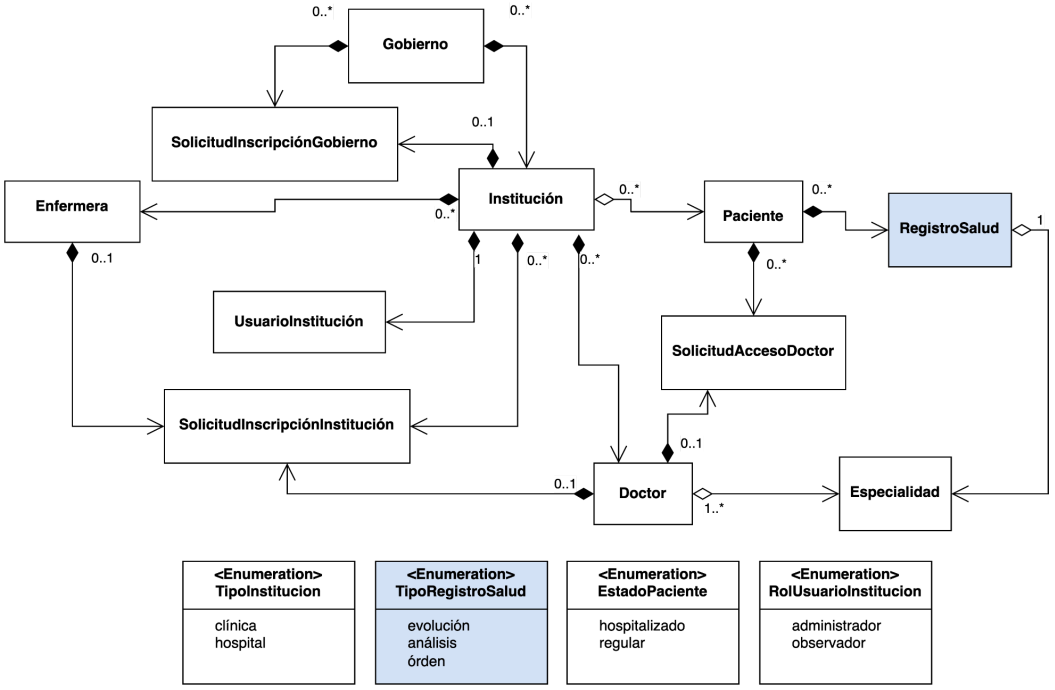


Figura 4.1: Diagrama de clases del sistema.

Definición de clases

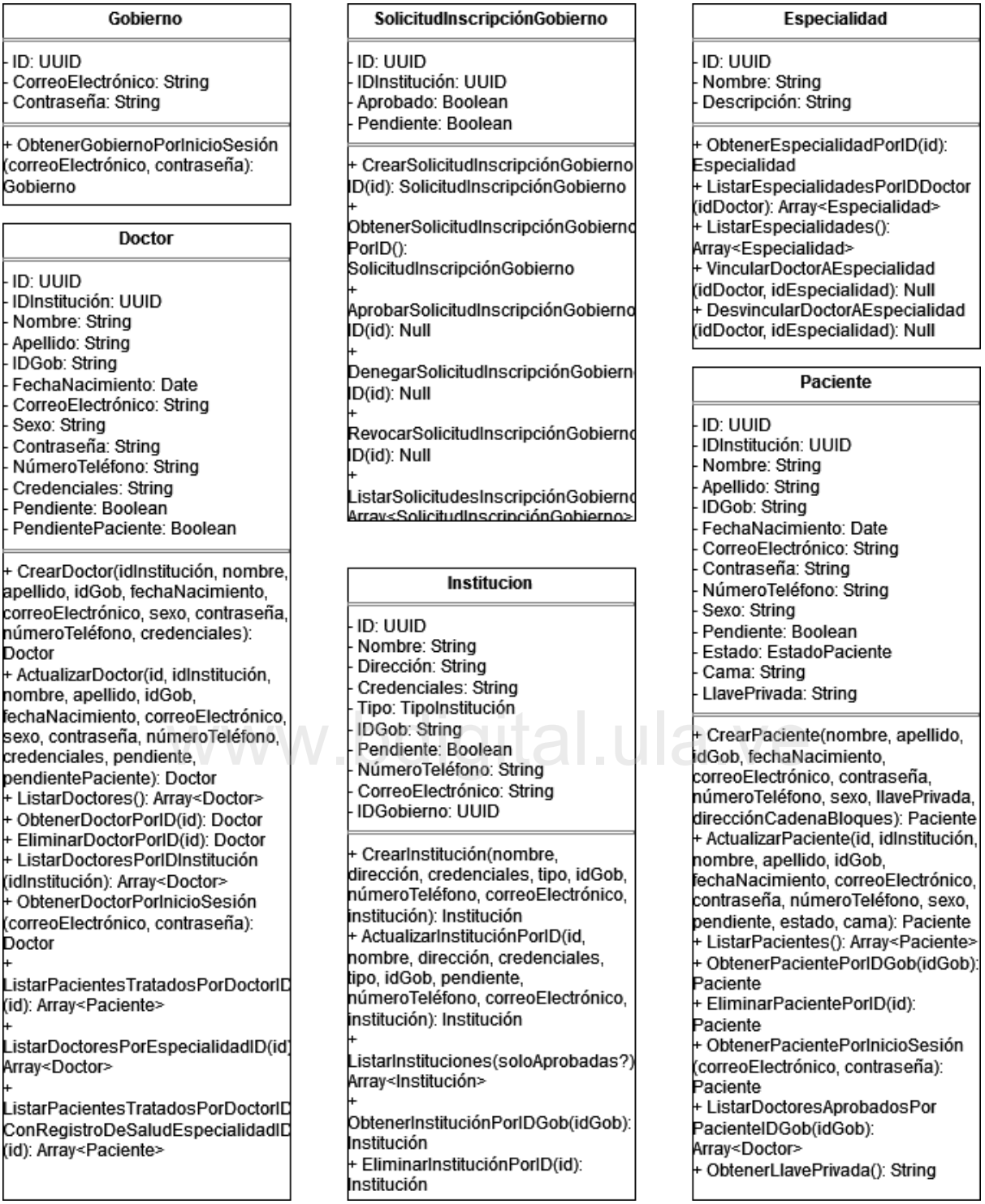


Figura 4.2: Definición de las clases del sistema.

Diagrama de casos de uso

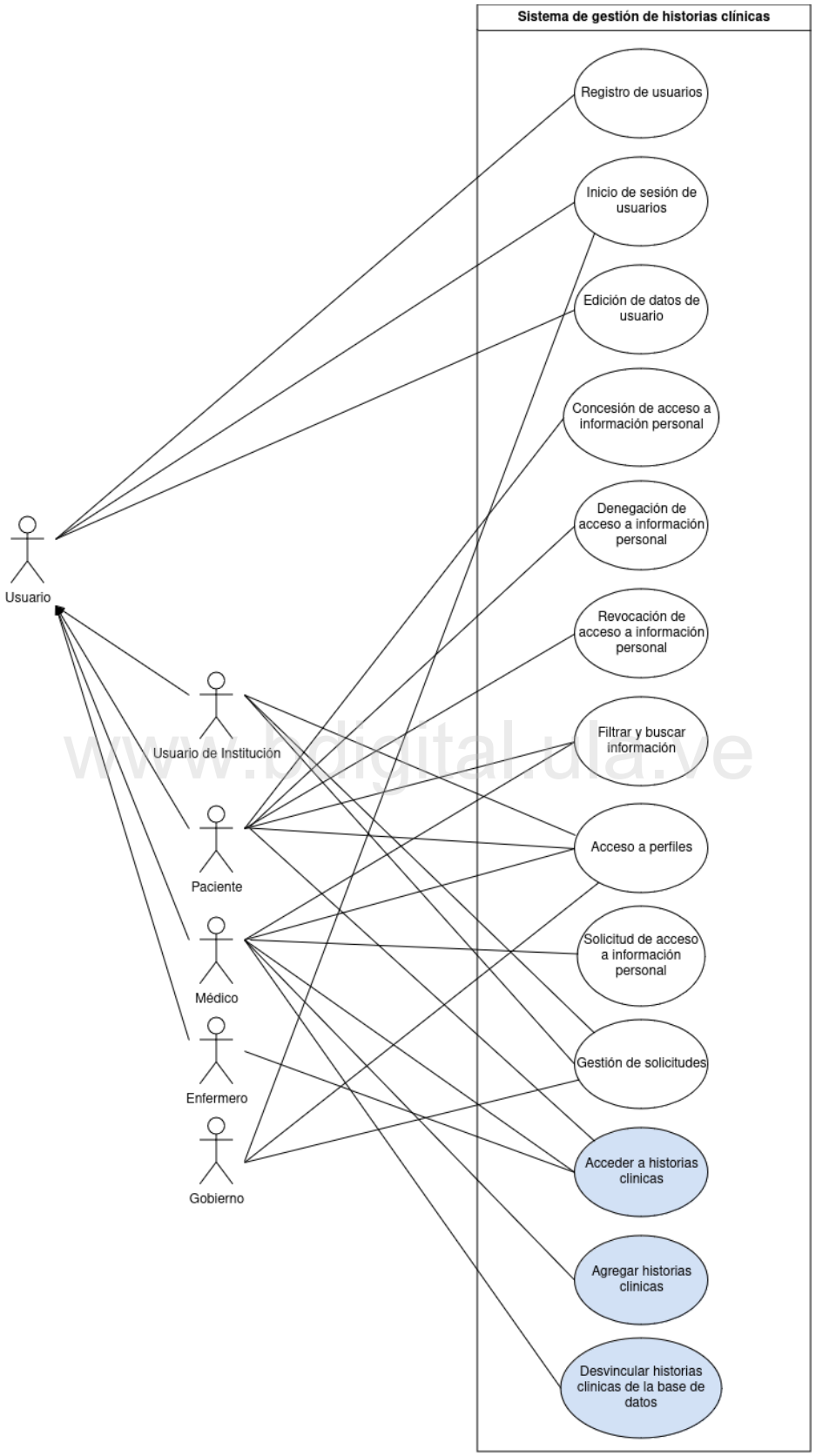


Figura 4.3: Diagrama de casos de uso.

Casos de uso	**Descripción**	**Actor**
Registro de usuarios	Almacena datos del usuario dentro del sistema.	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 con base en criterios especificados	Paciente, Médico
Acceso a perfiles	Lectura de la información de los usuarios almacenados en el sistema	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 cadena de bloques y su información auxiliar en la base de datos auxiliar	Médico
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 relacionados con actores.

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.

Filtrar y buscar 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.

Diagrama de clases

Figura 4.4: Diagrama de clases RCU - Filtrar y buscar información.

Diagrama de secuencia

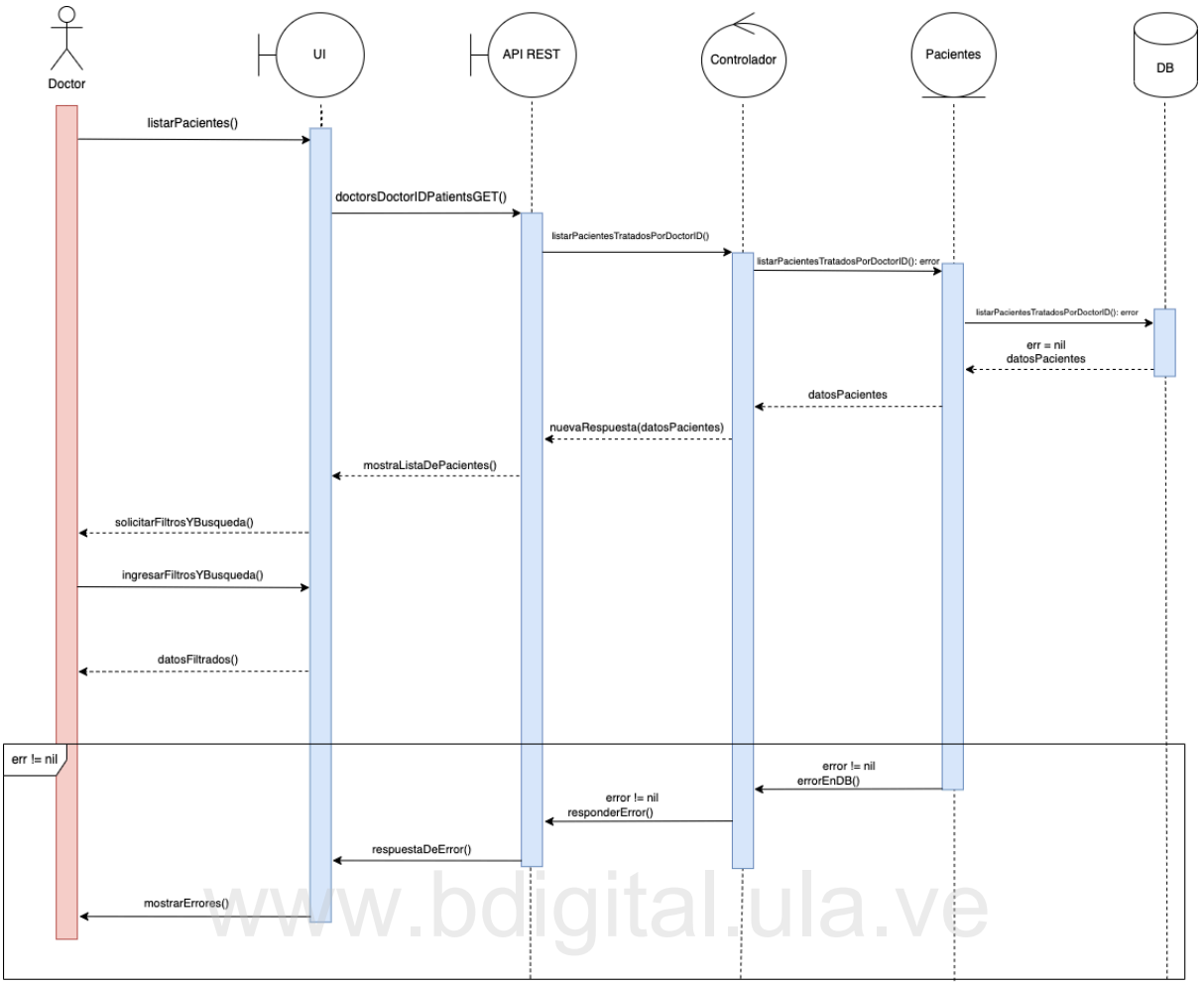


Figura 4.5: Diagrama de secuencia RCU - Filtrar y buscar información.

Denegación de acceso a la información personal

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

Diagrama de clases

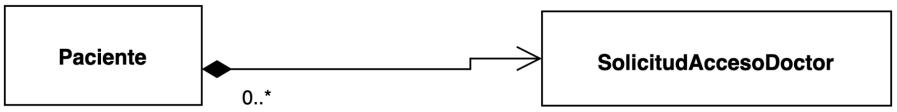


Figura 4.6: Diagrama de clases RCU - Denegación de acceso a la información personal.

Diagrama de secuencia

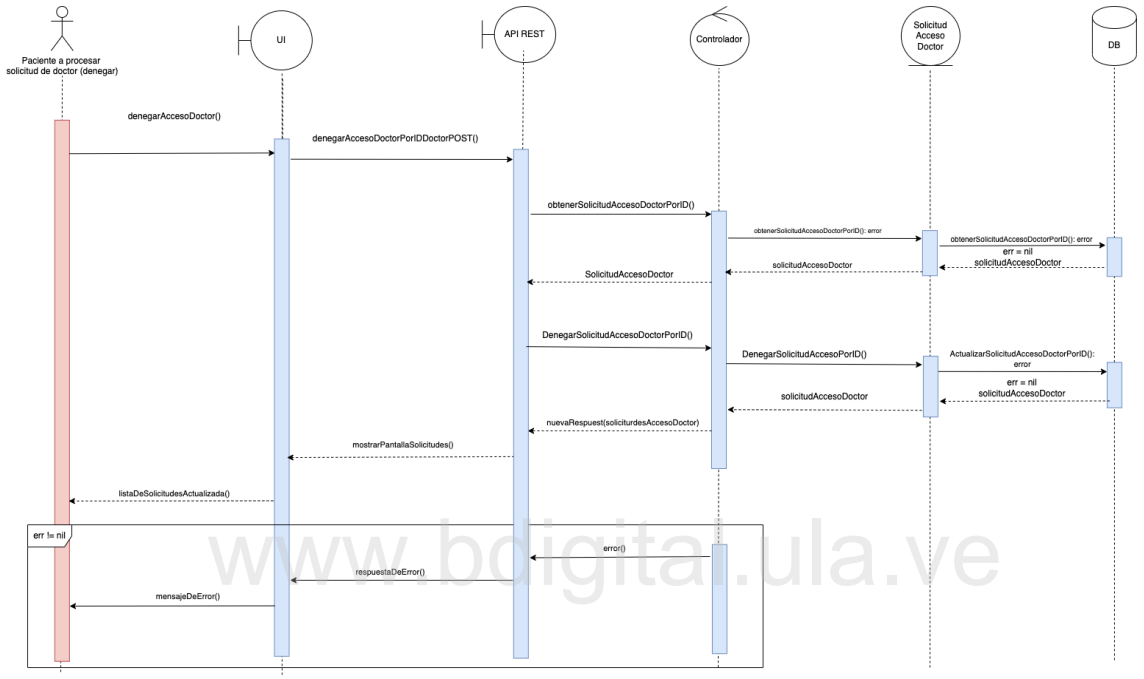


Figura 4.7: Diagrama de secuencia RCU - Denegación de acceso a la información personal.

Concesión de acceso a la información personal

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

Diagrama de clases

Figura 4.8: Diagrama de clases RCU - Concesión de acceso a la información personal.

Diagrama de secuencia

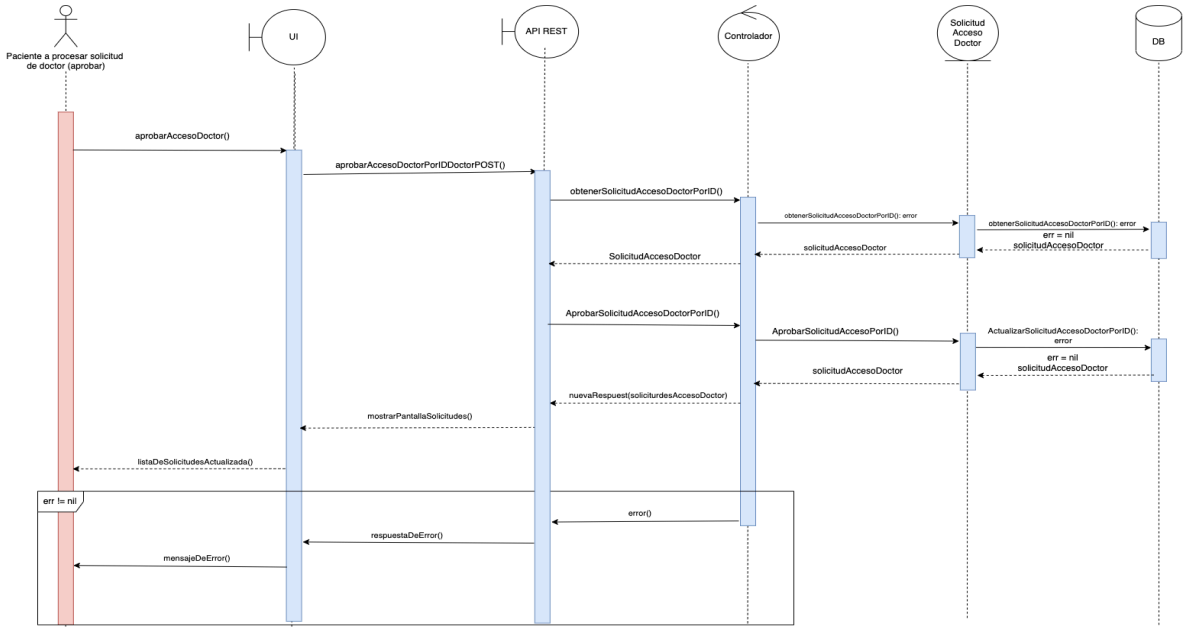


Figura 4.9: Diagrama de secuencia RCU - Concesión de acceso a la información personal.

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.

Diagrama de clases

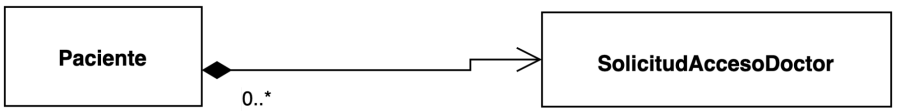


Figura 4.10: Diagrama de clases RCU - Revocación de acceso a la información personal.

Diagrama de secuencia

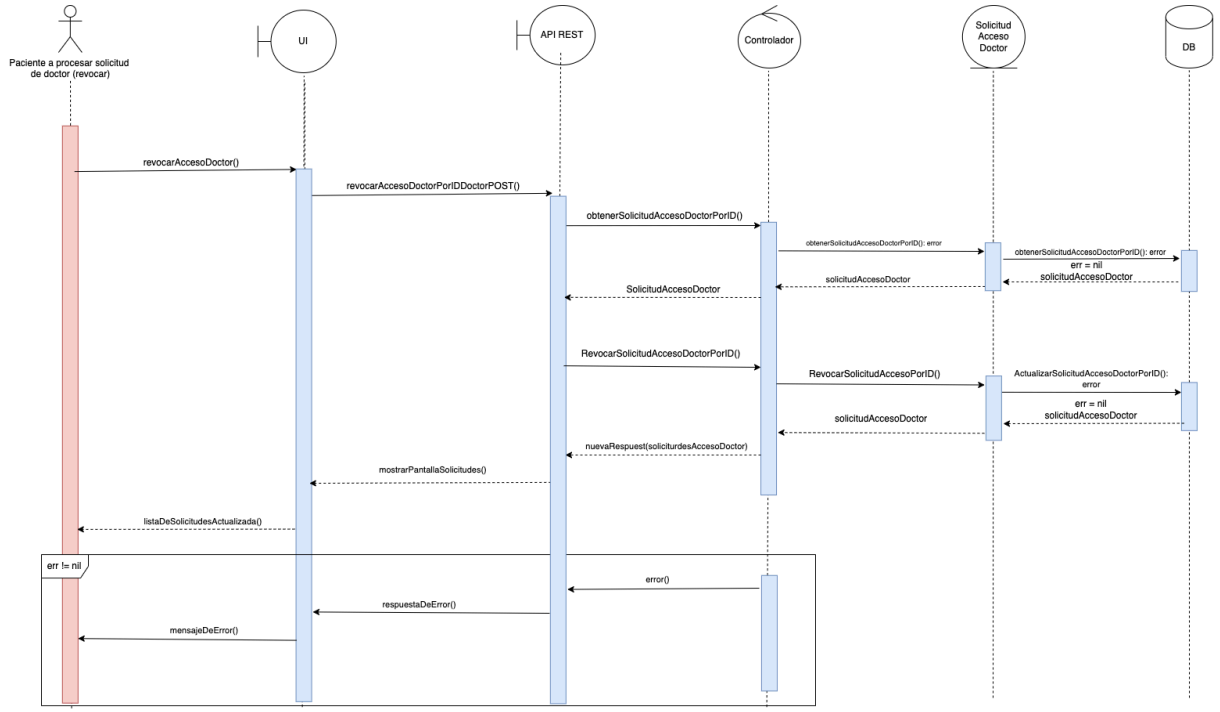


Figura 4.11: Diagrama de secuencia RCU - Revocación de acceso a la información personal.

Agregar a historia clínica

Este proceso permite a los médicos agregar un registro de salud a la historia clínica de un paciente

Diagrama de clases

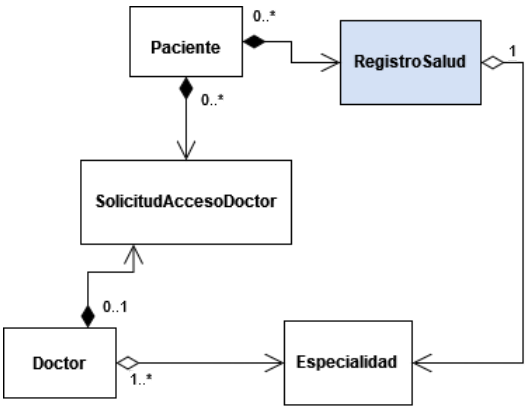


Figura 4.12: Diagrama de clases RCU - Agregar a historia clínica.

Diagrama de secuencia

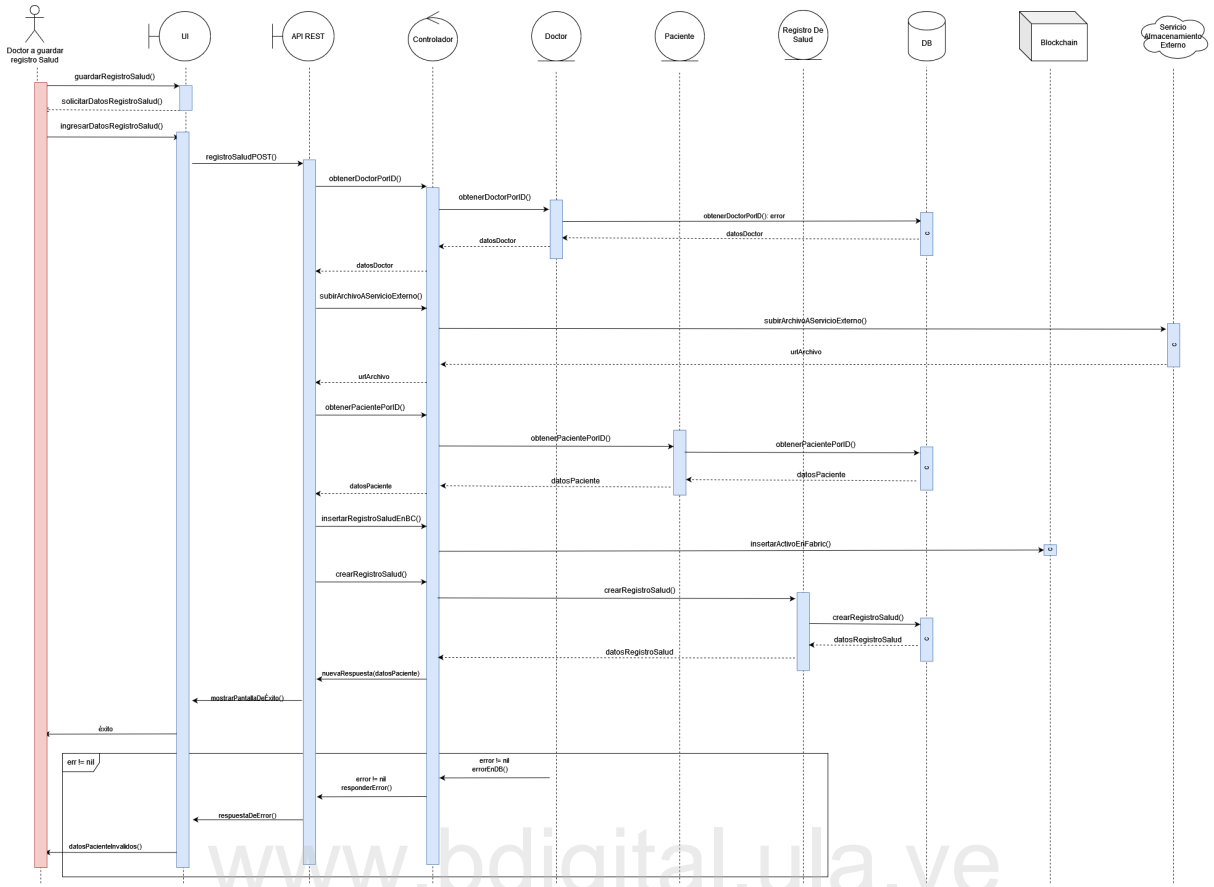


Figura 4.13: Diagrama de secuencia RCU - Agregar a historia clínica.

Acceder a historia clínica

Este proceso permite a los médicos acceder a la historia clínica de un paciente

Diagrama de clases

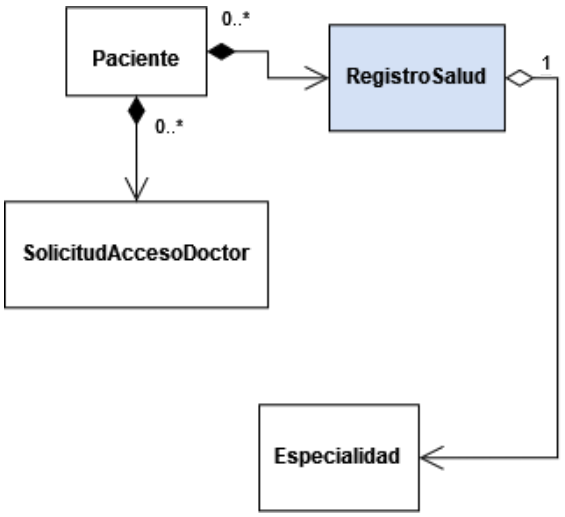


Figura 4.14: Diagrama de clases RCU - Acceder a historia clínica.

Diagrama de secuencia

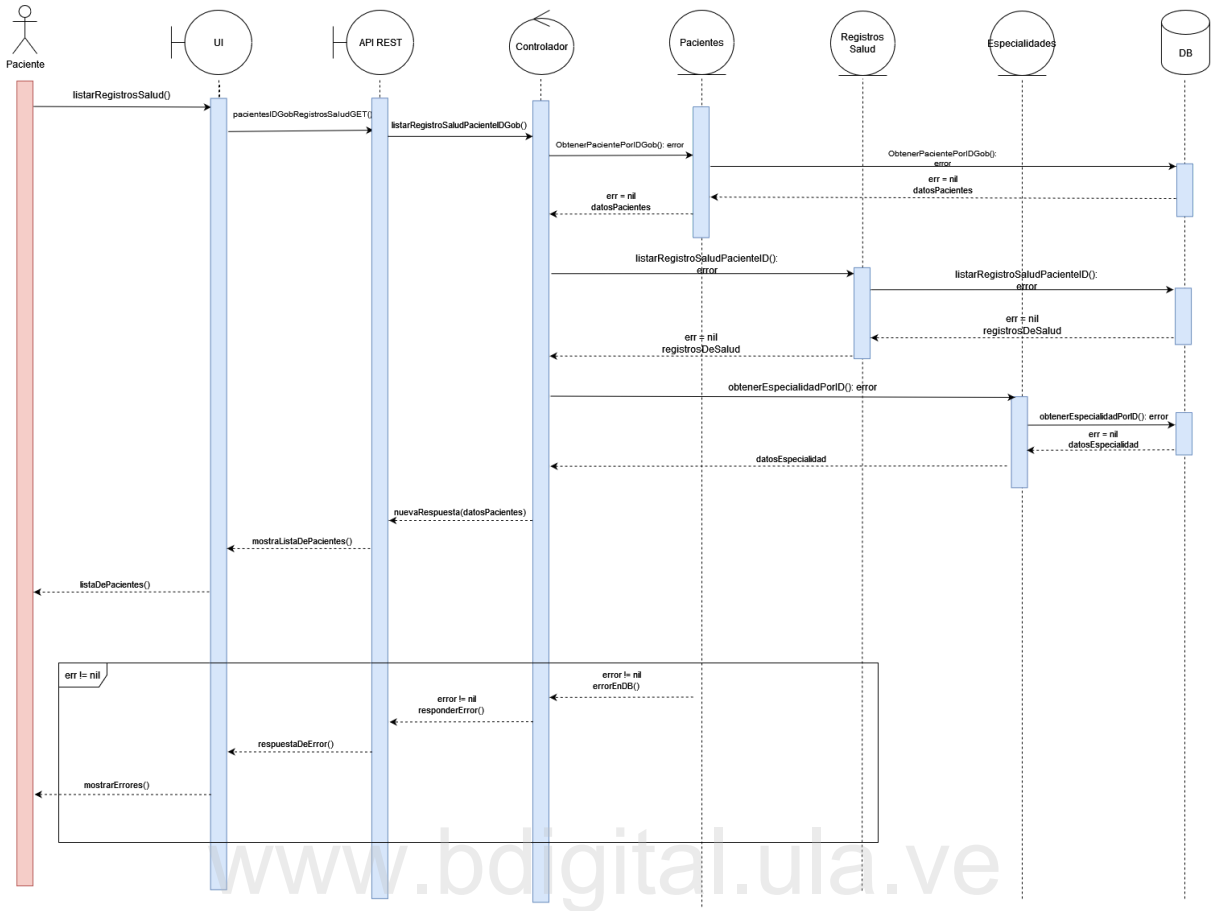


Figura 4.15: Diagrama de secuencia RCU - Acceder a historia clínica.

Desvincular historia clínica

Este proceso permite a los pacientes desvincular sus historias clínicas de la base de datos auxiliar del sistema

Diagrama de clases

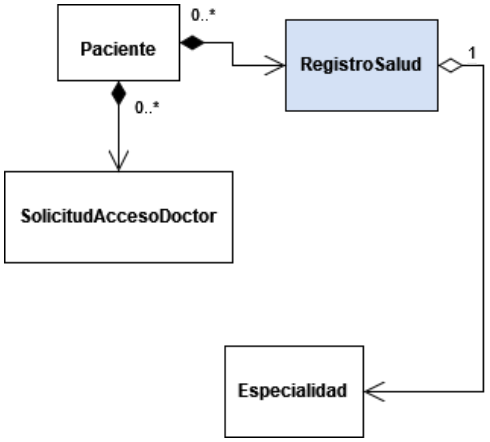


Figura 4.16: Diagrama de clases RCU - Desvincular la historia clínica.

Diagrama de secuencia

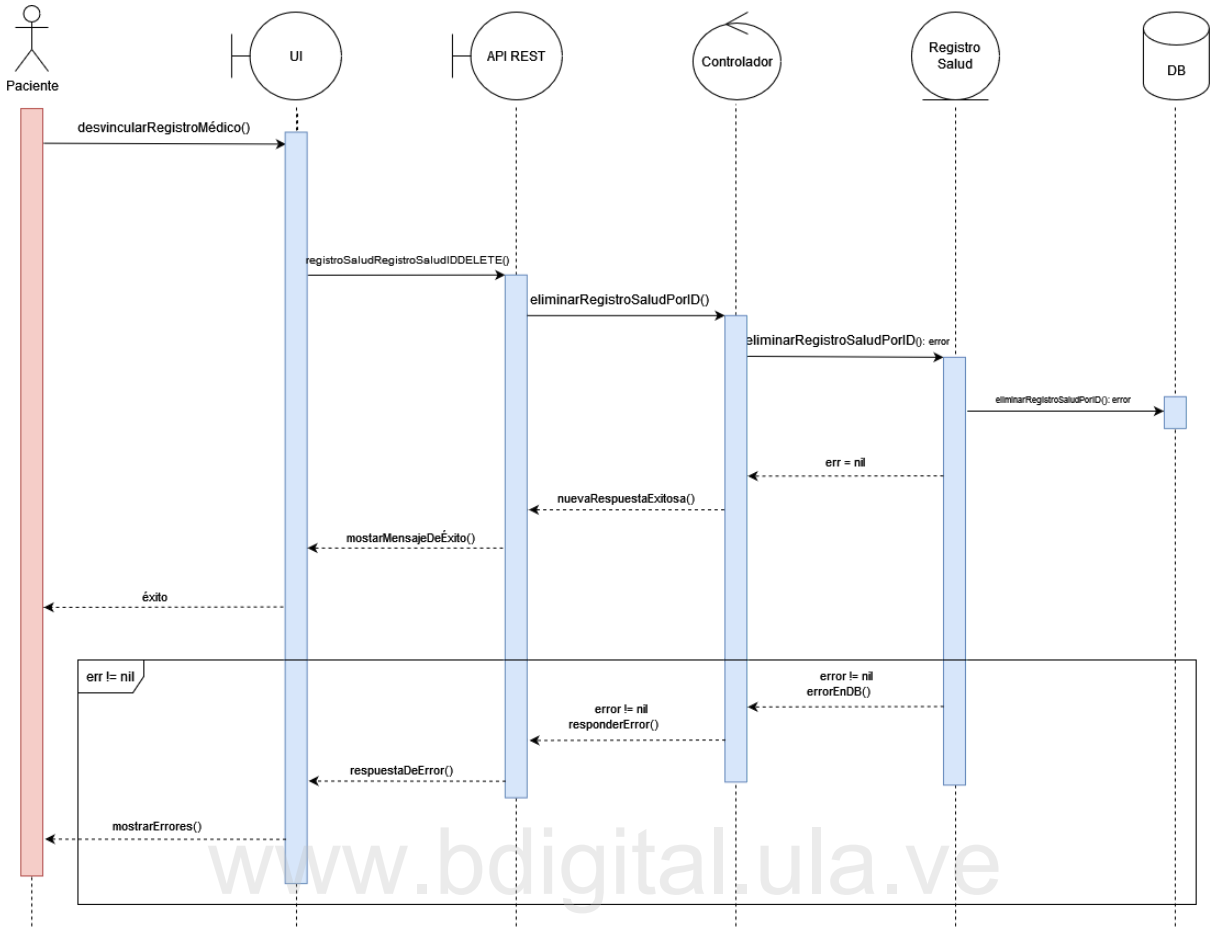


Figura 4.17: Diagrama de secuencia RCU - Desvincular la historia clínica.

Capítulo 5

Diseño de la Aplicación

Al desarrollar el sistema de gestión de historias clínicas se busca una solución que sea segura, eficiente y escalable, abordando problemas de seguridad, privacidad, eficiencia y alcance en las instituciones de salud. Para producir un prototipo que realmente abarque todos estos retos, este proyecto se desarrolló de forma complementaria con el proyecto "Desarrollo de un Sistema de Gestión de Identidad de Pacientes en Instituciones de Salud mediante Tecnología de Cadena de Bloques". Se tiene como objetivo principal de este capítulo definir la arquitectura de la aplicación y especificar sus componentes desde un nivel de abstracción alto.

La gestión de historias clínicas son fundamentales en cualquier institución de salud, los médicos hacen uso de las mismas para poder tomar decisiones médicos informadas con no solo el contexto actual sino histórico de un paciente. La implementación de tecnología de cadena de bloques tiene el potencial de mejorar considerablemente la seguridad, transparencia y eficiencia en la gestión de estos datos. Por tanto, es esencial diseñar una aplicación robusta y eficiente para alcanzar estos objetivos.

En primer lugar, se ofrecerá una visión general de la arquitectura propuesta, describiendo cómo interactúan entre sí los distintos componentes del sistema y con la cadena de bloques. Posteriormente, se detallarán los principales componentes del sistema, incluyendo la interfaz de usuario, la lógica de negocios y la capa de datos. Cada componente será analizado desde una perspectiva de alto nivel, proporcionando una comprensión clara de su función e interacciones dentro del sistema global.

A lo largo del capítulo, se subrayan los principios de diseño que guiarán la construcción del sistema, asegurando que sea escalable, seguro y fácil de mantener. También se explicarán las decisiones técnicas clave y las justificaciones detrás de ellas, basadas en las necesidades específicas del entorno de salud y las ventajas inherentes de la tecnología de cadena de bloques.

El diseño permitirá una implementación efectiva del sistema, estableciendo una base sólida para la gestión de historias clínicas de pacientes entre diferentes instituciones.

5.1. Refinamiento de requisitos arquitectónicos

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

Los requisitos que afectan la arquitectura de la solución propuesta para los usuarios potenciales del sistema son los siguientes:

Seguridad
Se garantiza la seguridad de los datos del producto mínimo viable mediante el cifrado de datos sensibles en la base de datos y dentro del blockchain, y el

control de acceso granular, restringiendo las posibles acciones a realizar con los roles disponibles para los usuarios.

Autenticación Robusta
El acceso a los datos se autentica de forma robusta utilizando un token para las solicitudes a la API REST. Este token solo puede generarse con la clave proporcionada por el paciente al registrarse, la cual está encriptada en la base de datos.

Control de Acceso Granular
El acceso a los datos de los pacientes se gestiona mediante tablas que registran los permisos otorgados por los pacientes a los médicos. La aplicación incluye una jerarquía de permisos que asegura que solo médicos e instituciones autorizados participen. Los usuarios gubernamentales se configuran manualmente para garantizar unicidad. Las instituciones deben ser verificadas y aprobadas por un usuario gubernamental, y los médicos deben ser aprobados por un administrador institucional.

Gestión de Revocación de Consentimiento
La interfaz gráfica de la aplicación permite a los pacientes revocar su consentimiento para el uso de sus datos a través de métodos provistos por la API REST.

Arquitectura Escalable
La arquitectura distribuida de la cadena de bloques facilita la escalabilidad según las necesidades del sistema. La separación de la capa de negocios y datos permite la orquestación con herramientas como Kubernetes.

5.1.2 Especificación de Requisitos Arquitectónicos No Funcionales

Seguridad

- ***Confidencialidad de los datos***: Los datos deben protegerse del acceso no autorizado.
- ***Integridad de los datos***: El sistema debe prevenir modificaciones no autorizadas.
- ***Disponibilidad de los datos***: Los datos deben estar disponibles para usuarios autorizados cuando los necesiten.

- ***Autenticación***: Verificar la identidad de los usuarios antes de otorgar acceso.
- ***Autorización***: Controlar el acceso a recursos y funciones según los roles y permisos.

Escalabilidad

- ***Escalabilidad de usuarios y datos***: Manejar un número creciente de usuarios y datos sin afectar el rendimiento.

Confiabilidad

- ***Tolerancia a fallos***: Continuar funcionando en caso de fallas de software.
- ***Capacidad de recuperación***: Restaurar el servicio rápidamente tras fallas, con mínima pérdida de datos.
- ***Previsibilidad***: Comportarse de manera predecible y consistente bajo condiciones normales y de carga.

Usabilidad

- ***Facilidad de uso***: Ser fácil de usar para los usuarios.
- ***Facilidad de aprendizaje***: Facilitar a los usuarios aprender rápidamente sin necesidad de mucha capacitación.

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

Confidencialidad de los datos

Almacenar de forma segura los datos de los pacientes en la cadena de bloques y la base de datos auxiliar, utilizando mecanismos de cifrado y control de acceso estrictos.

Integridad de los datos

La inmutabilidad de la cadena de bloques asegura que el historial de registros de salud de los pacientes no se puede alterar, permitiendo la trazabilidad y transparencia en la atención médica.

Disponibilidad de los datos

Garantizar la disponibilidad del sistema mediante nodos redundantes en la red blockchain y mecanismos de recuperación ante desastres.

Autenticación
Implementar mecanismos robustos para verificar la identidad de los usuarios, asegurando que solo personal autorizado acceda a datos sensibles.

Autorización
Permitir un control de acceso granular basado en roles y responsabilidades de cada usuario en el sistema de salud.

Escalabilidad de usuarios y datos
Diseñar la aplicación con una arquitectura modular y orientada a microservicios, facilitando la escalabilidad horizontal al añadir más nodos a la red blockchain.

Tolerancia a fallos
Utilizar mecanismos de redundancia y replicación de datos para asegurar el funcionamiento continuo del sistema.

Capacidad de recuperación
Implementar mecanismos robustos de recuperación para restaurar rápidamente el sistema en caso de fallas, incluyendo copias de seguridad regulares y planes de recuperación ante desastres.

Previsibilidad
Realizar pruebas exhaustivas y monitorear el rendimiento del sistema proactivamente para asegurar un comportamiento consistente.

Facilidad de uso y de aprendizaje
Desarrollar una interfaz de usuario intuitiva y accesible desde dispositivos móviles y de escritorio, con menús claros e iconos intuitivos.

5.2. Diseño de la Arquitectura

5.2.1. Análisis de la Lista de Requisitos Arquitectónicos

La arquitectura del sistema debe diseñarse para cumplir con los requisitos de seguridad, escalabilidad, confiabilidad y usabilidad. La tecnología blockchain ofrece características ideales para la gestión segura de datos de pacientes,

almacenándolos en un registro distribuido e inmutable, accesible solo mediante criptografía. Cada bloque contiene un *hash* del bloque anterior, creando una cadena inviolable que garantiza la integridad de los datos. La descentralización de la blockchain asegura la disponibilidad de los datos incluso en caso de fallos en nodos individuales.

Para una gestión eficiente y escalable de las historias clínicas de los pacientes, se propone una arquitectura modular compuesta por:

Cadena de bloques
Almacena la dirección encriptada de los archivos de registros de salud de un paciente.

Base de Datos Auxiliar
Almacena metadatos e índices para optimizar el rendimiento y accesibilidad.

API REST
Sirve como punto de entrada único para todas las solicitudes de la aplicación, proporcionando autenticación y autorización para interactuar con los componentes del sistema.

Controlador Principal
Gestiona la lógica de negocio, incluyendo autenticación, autorización, registro de pacientes y administración de accesos.

Servicio Externo de Almacenamiento de Archivos
Se encarga de almacenar los archivos asociados a los registros de salud de un paciente, el cual se encarga de escoger dicho servicio e identificar las posibles fortalezas y vulnerabilidades de utilizar el mismo, así como el control total de sus archivos.

La API REST y el controlador principal, junto con la base de datos auxiliar y la blockchain, deben estar en servicios independientes y escalables, almacenados en contenedores "dockerizados" para portabilidad y consistencia en diferentes entornos.

La arquitectura aprovecha la escalabilidad horizontal de la tecnología blockchain y los microservicios para manejar un mayor volumen de usuarios y

transacciones. La redundancia y replicación de datos garantizan la continuidad del servicio incluso en caso de fallos.

Para una experiencia de usuario fluida, se propone un cliente basado en React y Next.js, tecnologías front-end que permiten interfaces dinámicas y receptivas, junto con Electron y Capacitor para empaquetar aplicaciones web en aplicaciones móviles nativas.

5.2.2. Capas de la Aplicación

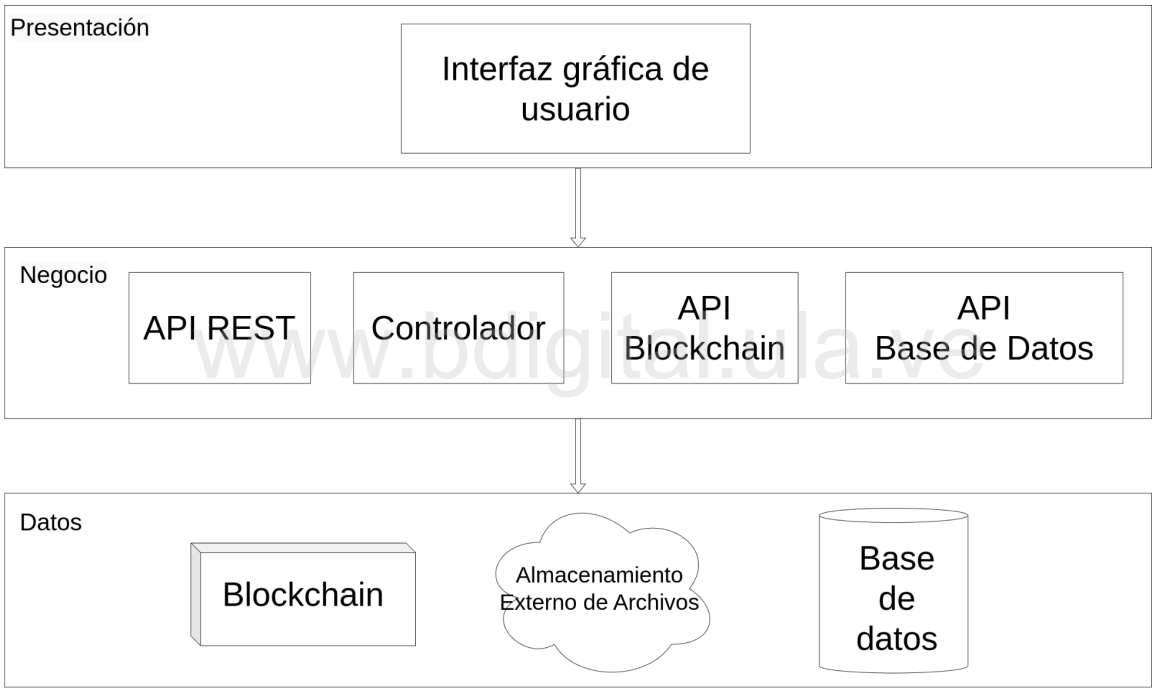


Figura 5.1: Diagrama arquitectónico de la aplicación.

La aplicación se estructura en tres capas:

Capa de Presentación

- ***Responsabilidades***: Interactuar con el usuario, presentar información y capturar datos.
- ***Componentes***: Interfaz de usuario web, móvil y de escritorio con React, Next.js, Electron y Capacitor.

Capa de Negocio

- ***Responsabilidades***: Implementar la lógica del sistema, gestionar historias clínicas, autenticación, autorización y acceso a datos.
- ***Componentes***: API REST, controlador principal, API de blockchain, API de base de datos y API de servicio externo para almacenamiento de archivos.

Capa de Datos

- ***Responsabilidades***: Almacenar y gestionar la información sobre las historias clínicas de los pacientes, así como la información general de todos los usuarios, asegurar la integridad y disponibilidad de los datos.
- ***Componentes***: Cadena de bloques configurada a partir de Hyperledger Fabric, base de datos auxiliar PostgreSQL y el servicio externo de almacenamiento de archivos escogido por el usuario.

La interacción entre estas capas se realiza de la siguiente manera: El usuario interactúa con la capa de presentación, que envía las solicitudes a la API REST. La API REST valida la solicitud y la envía al controlador correspondiente en la capa de negocio. El controlador procesa la solicitud y accede a los datos en la capa de datos. Los datos recuperados se envían de vuelta a la API REST y luego a la capa de presentación para mostrarse al usuario.

5.3. Diseño de la Interfaz Gráfica de Usuario

Después de un análisis exhaustivo de los requisitos relacionados con la interfaz gráfica de usuario (IGU), se identificaron las funciones de los usuarios a través de casos de uso e historias de usuario. Esto permitió diseñar la estructura jerárquica general de la IGU y la estructura de las páginas web.

La estructura jerárquica de la IGU se define como la organización lógica de los elementos de la interfaz, estableciendo relaciones de importancia y prioridad. Esta organización guía al usuario de forma intuitiva a través de la aplicación, facilitando la realización de tareas.

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, utilizando la siguiente representación:

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

Médicos

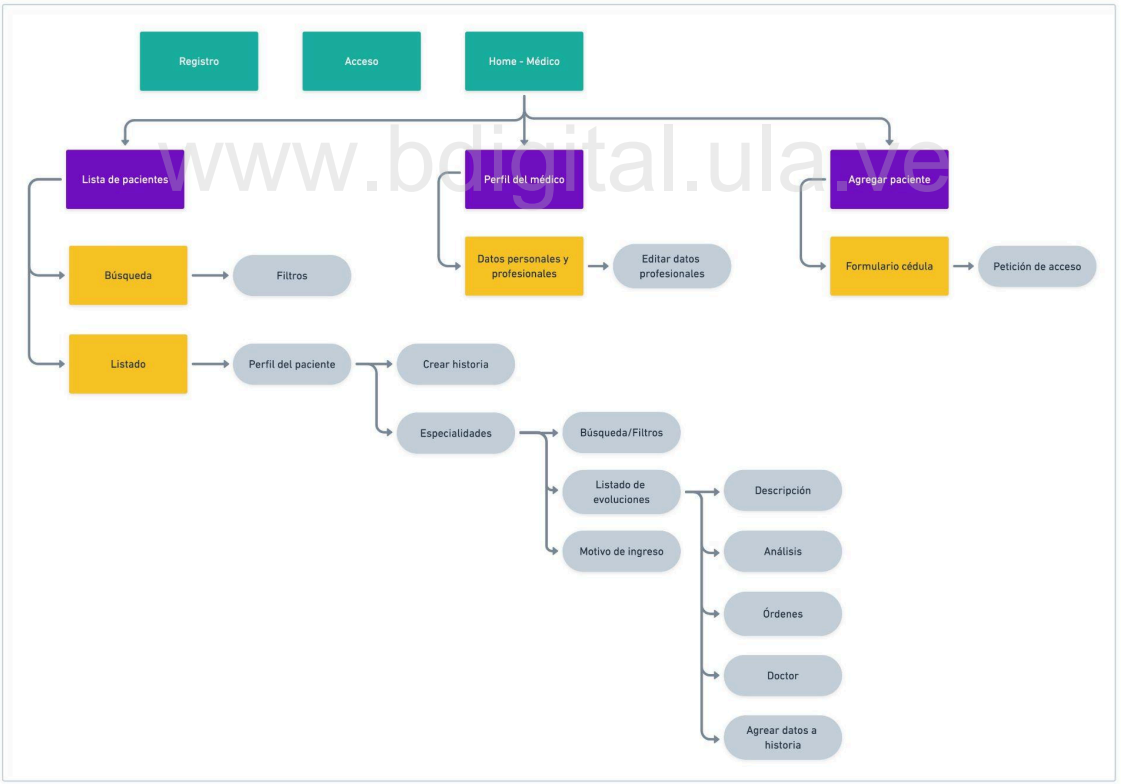


Figura 5.3: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para médicos.

Enfermeros y enfermeras

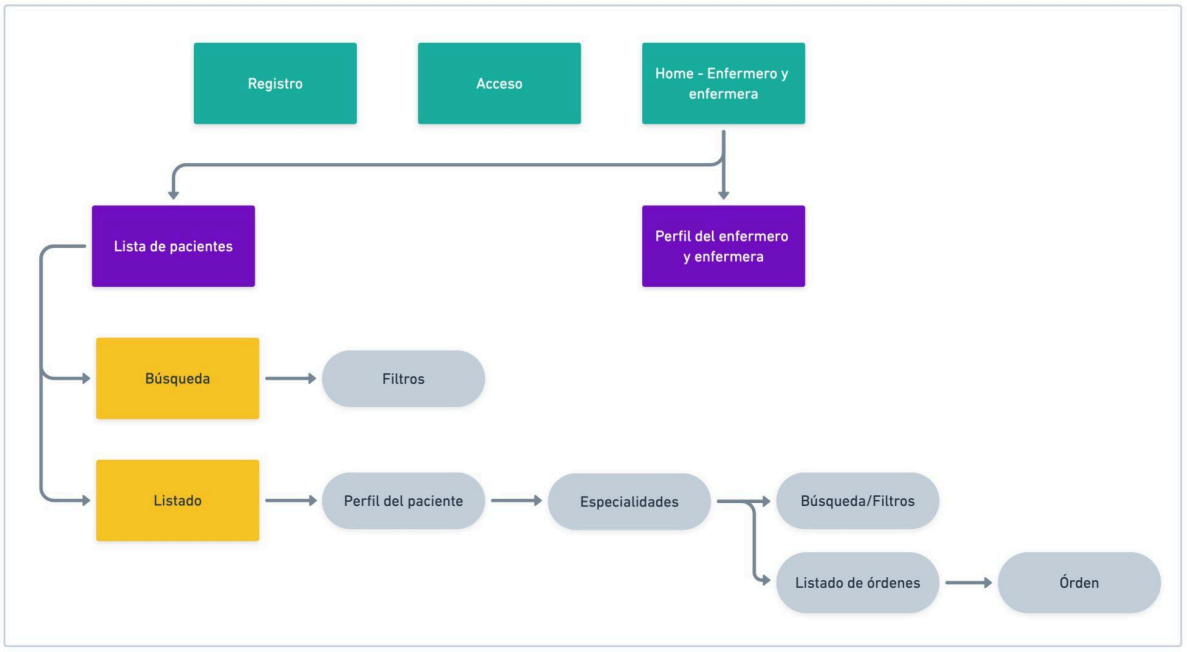


Figura 5.4: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para enfermeros y enfermeras.

Pacientes

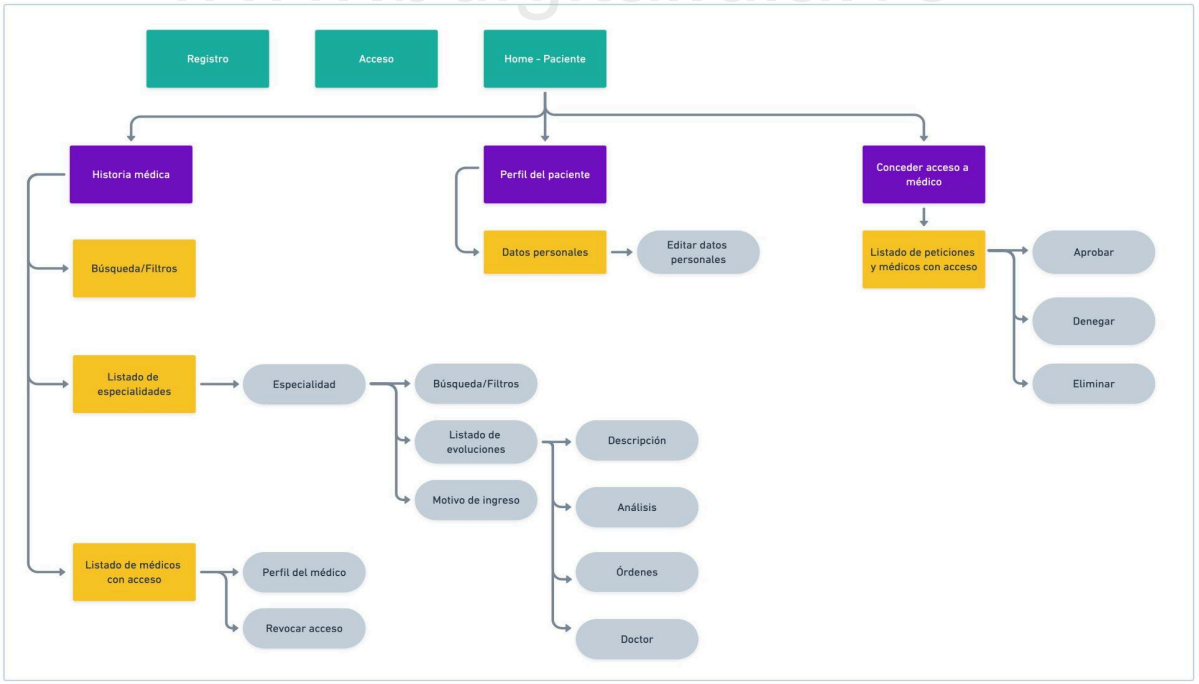


Figura 5.5: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para pacientes.

Instituciones

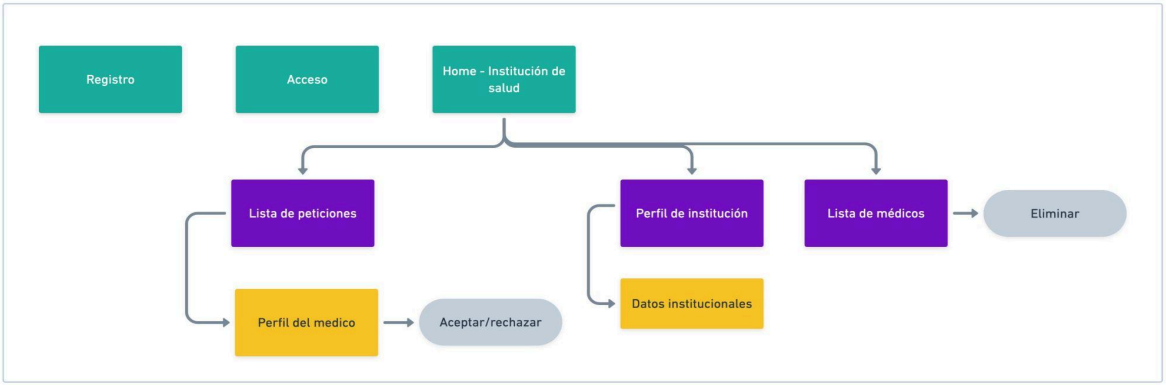


Figura 5.6: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para usuarios de instituciones.

Ente gubernamental

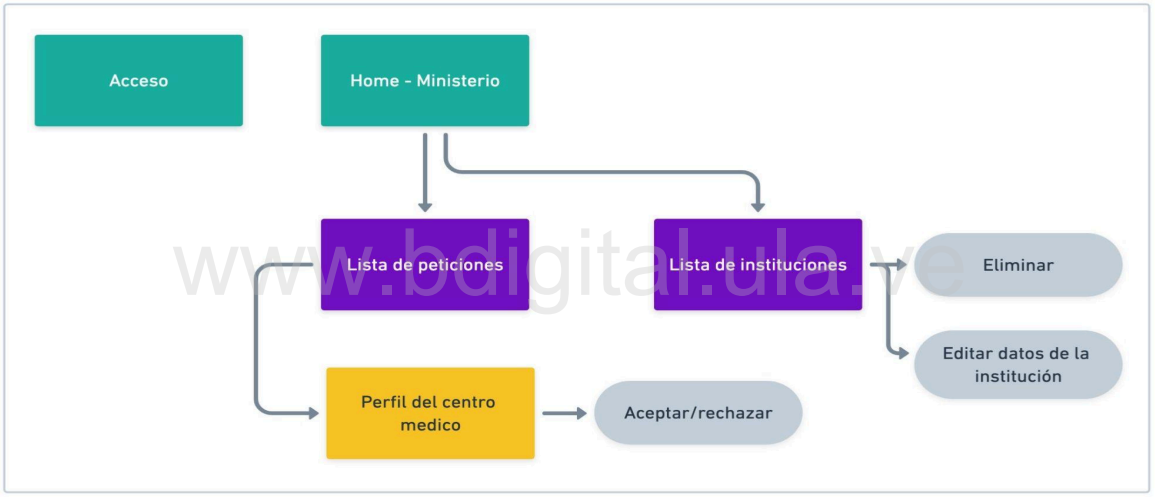


Figura 5.7: Estructura de la arquitectura de la información de la interfaz gráfica de usuario para usuarios del ente gubernamental.

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

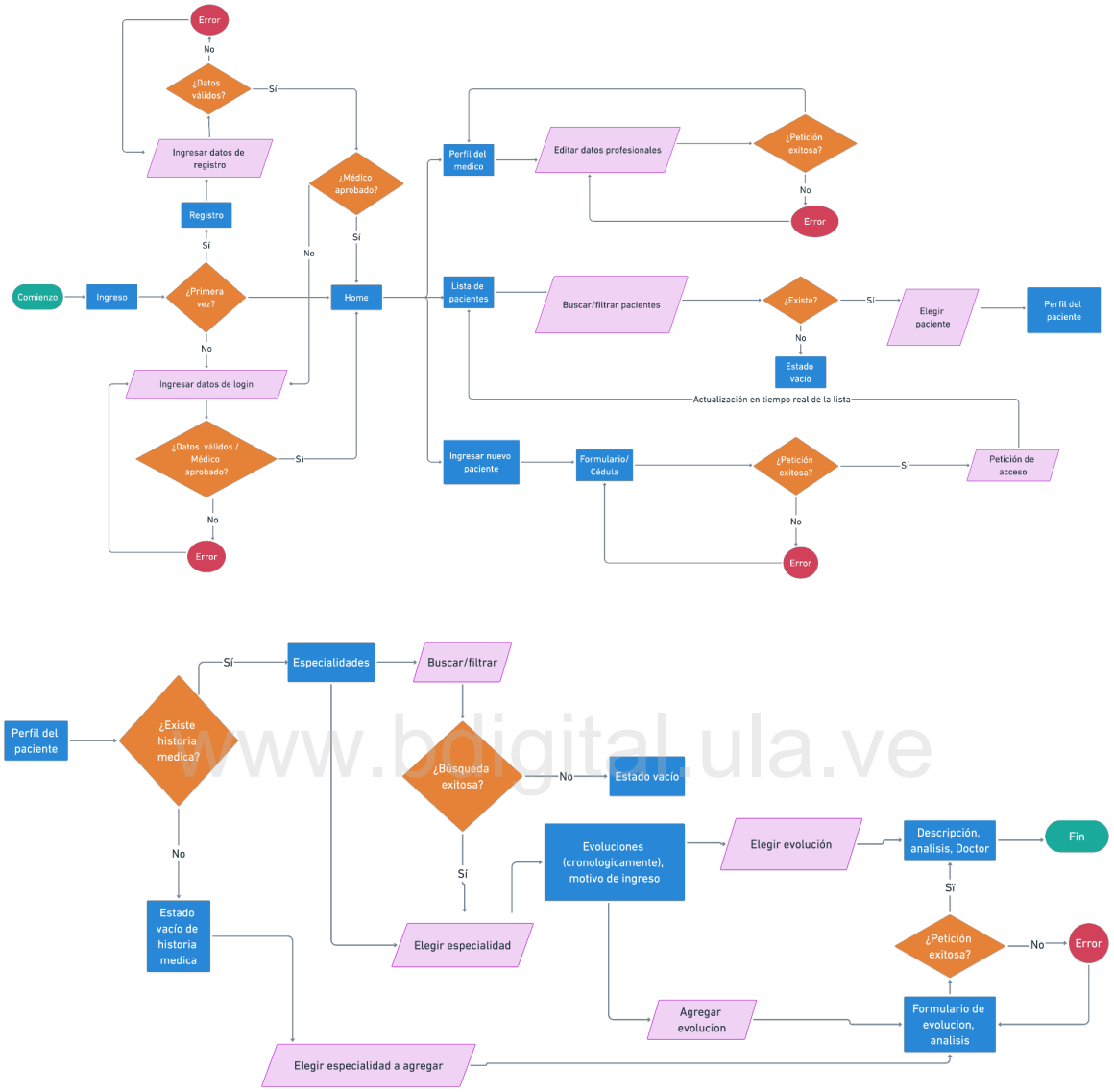


Figura 5.9: Estructura jerárquica de la interfaz gráfica de usuario para médicos.

Enfermeros y enfermeras

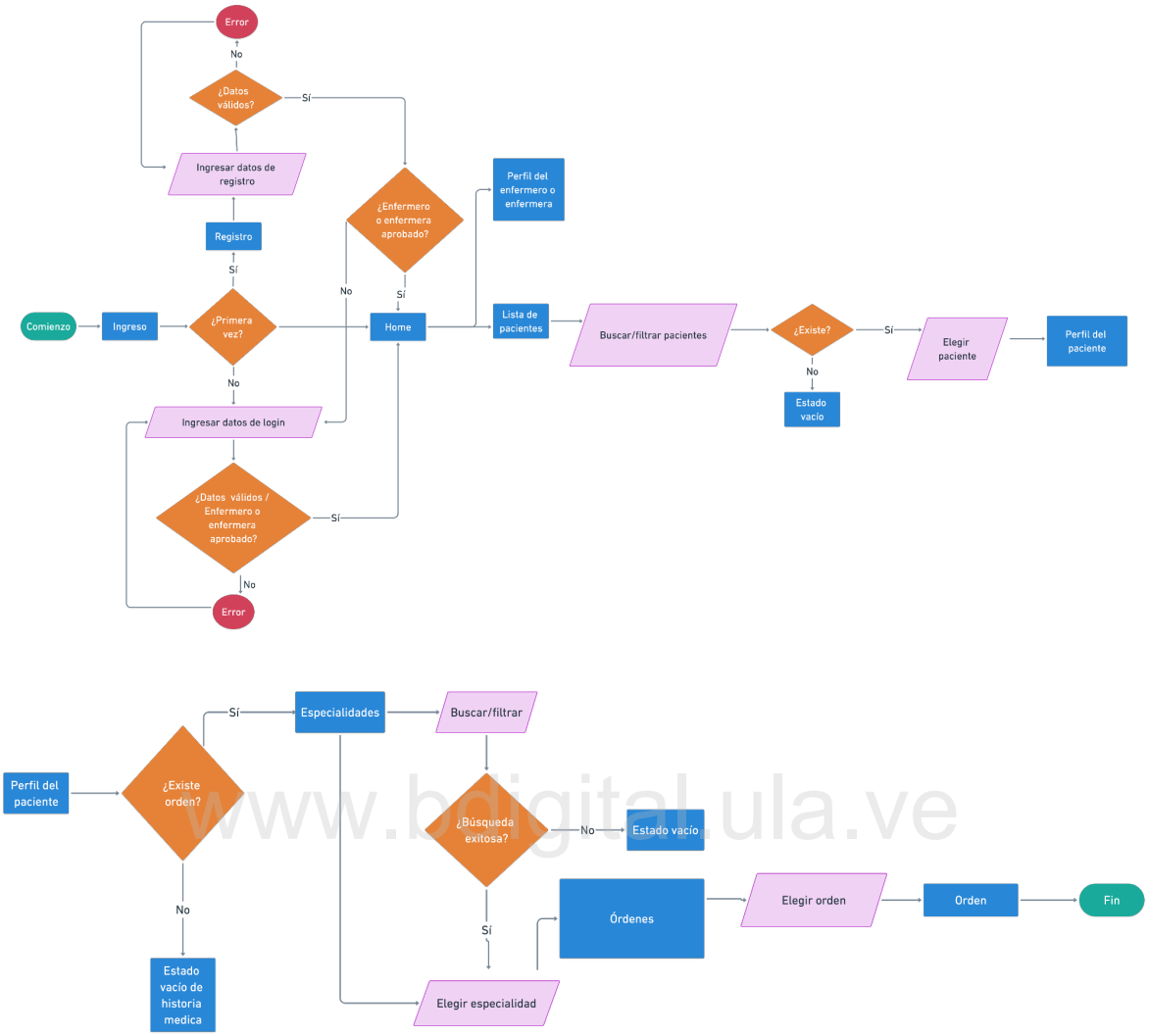


Figura 5.10: Estructura jerárquica de la interfaz gráfica de usuario para enfermeros y enfermeras.

Pacientes

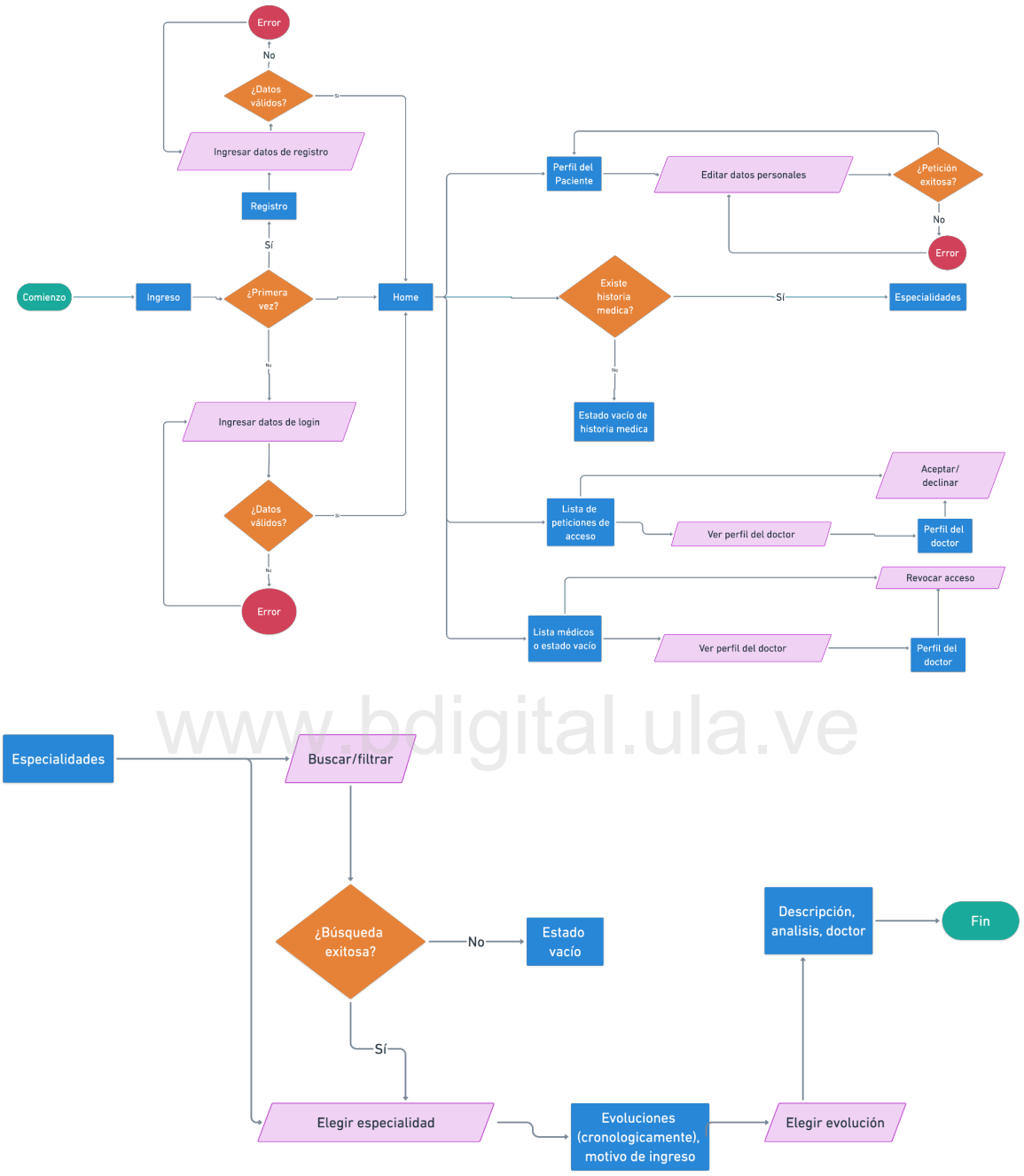


Figura 5.11: Estructura jerárquica de la interfaz gráfica de usuario para pacientes.

Institución

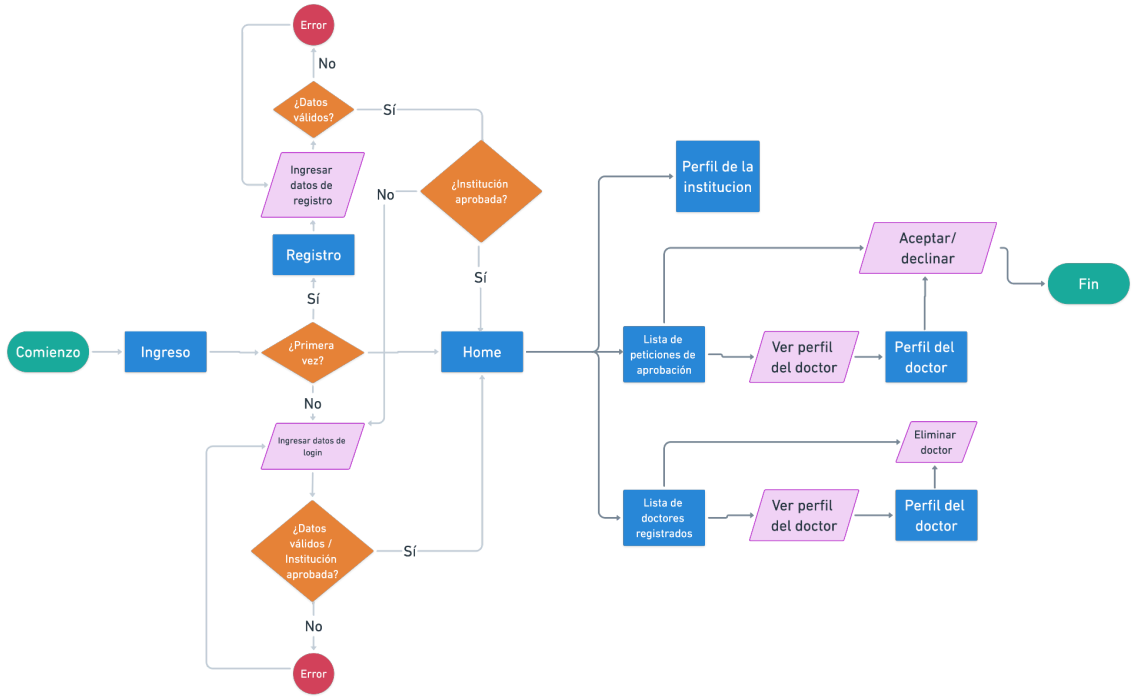


Figura 5.12: Estructura jerárquica de la interfaz gráfica de usuario para instituciones.

Ente gubernamental

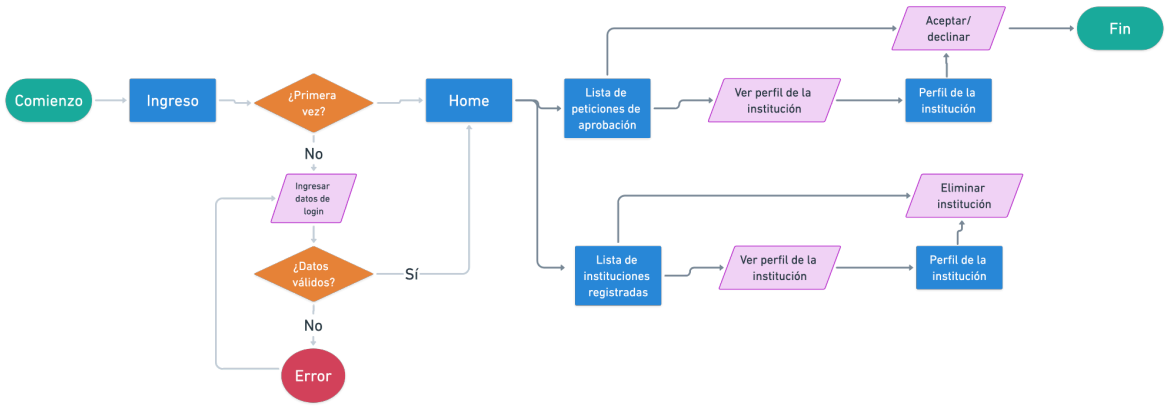


Figura 5.13: Estructura jerárquica de la interfaz gráfica de usuario para el ente gubernamental.

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.

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.

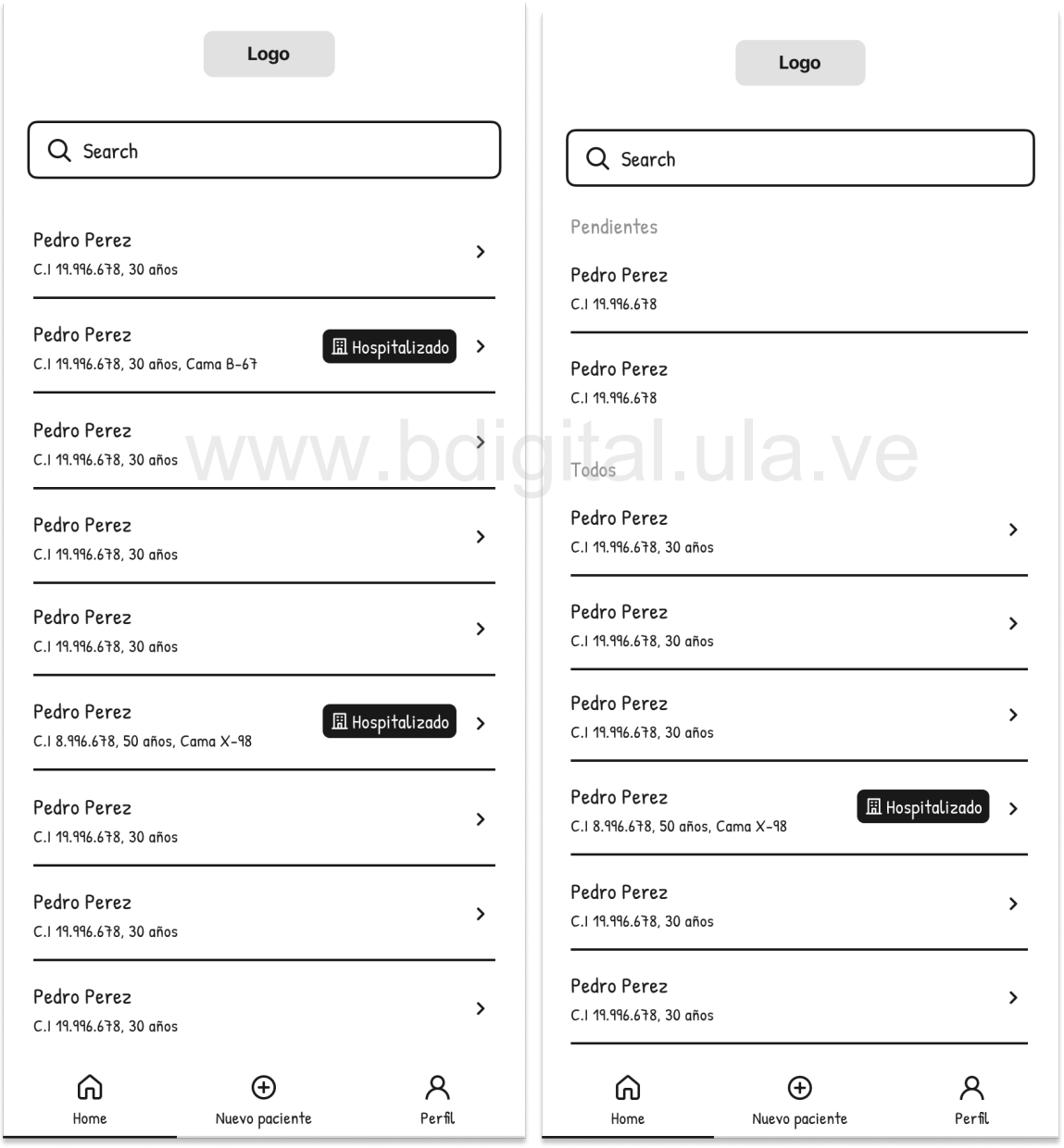


Figura 5.14: Menú de navegación.

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.

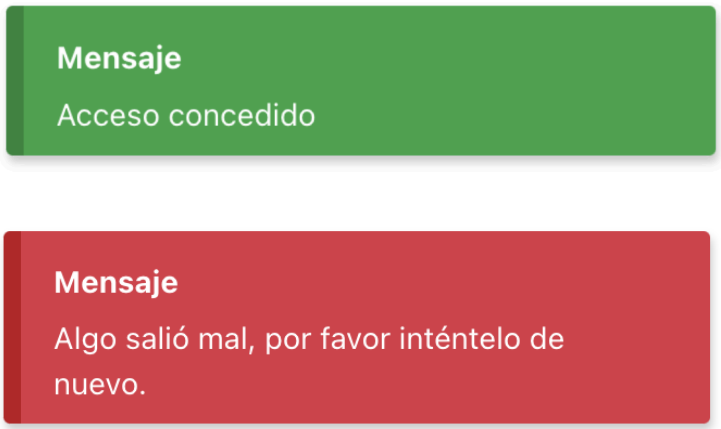


Figura 5.15: Mensajes de éxito y error.

Estados vacíos

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

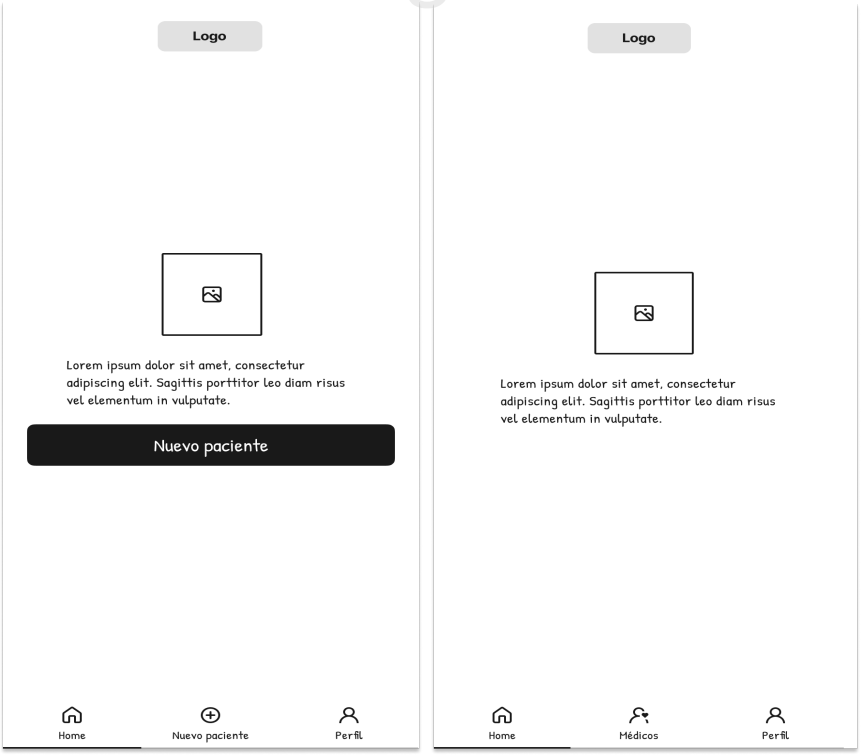


Figura 5.16: Estados vacíos.

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 5.17: Filtros y búsqueda.

Historias clínicas

Las historias clínicas se pueden agregar por tipos, listar y agrupar.

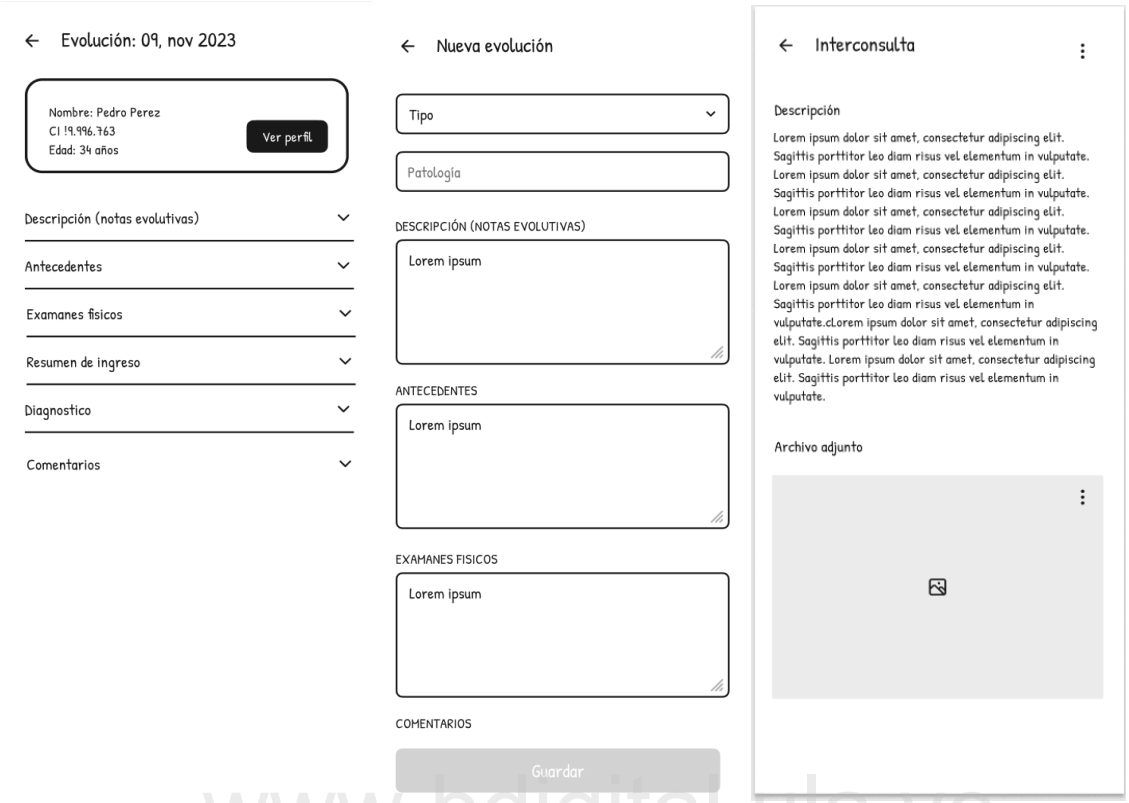


Figura 5.18: Historias Clínicas.

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.

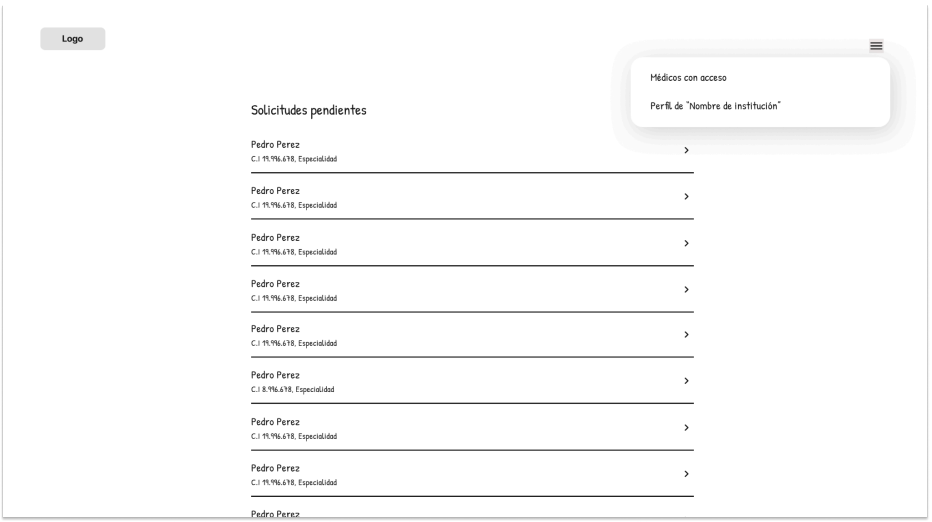


Figura 5.19: Vista de escritorio.

5.4. Diseño de la Base de Datos

El diseño de la base de datos garantiza la integridad, seguridad y trazabilidad de los registros de salud e historias clínicas de los pacientes. Se realizó un análisis detallado de los requisitos, identificando datos esenciales y sus interrelaciones. Se elaboró un esquema conceptual y se definieron lineamientos para la administración de la base de datos, estableciendo reglas de acceso, seguridad y respaldo. El diseño final se consolidó en un modelo de datos para la implementación de la base de datos auxiliar. Este contiene todas las tablas necesarias para el funcionamiento del sistema, incluyendo las definidas por el proyecto complementario.

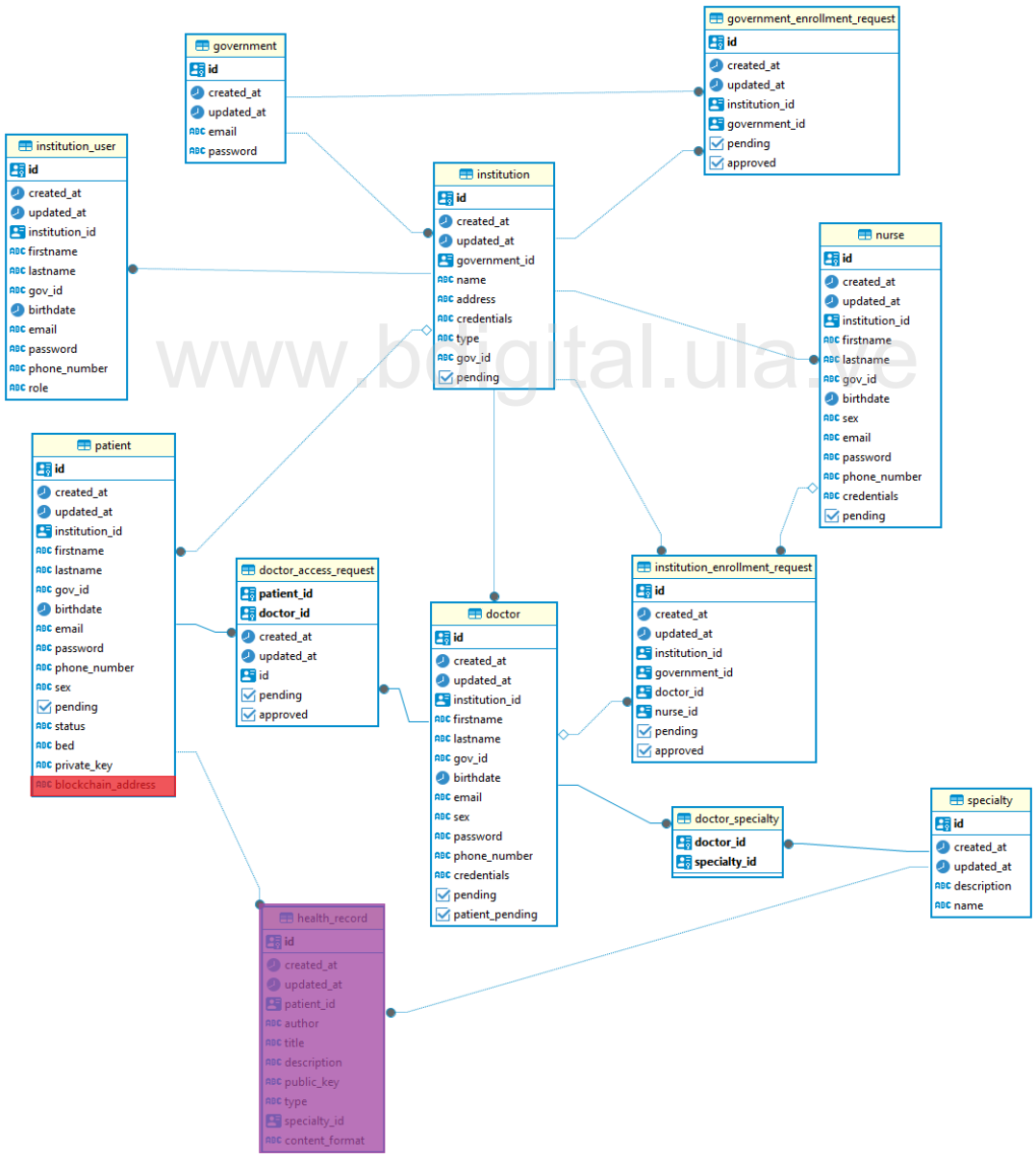


Figura 5.20: Diagrama de bases de datos.

Capítulo 6

Implementación y evaluación

Definir la cantidad de iteraciones necesarias para generar incrementos funcionales y presentables a los usuarios, así como establecer la duración de cada iteración, es una necesidad fundamental para el desarrollo de cualquier proyecto de software de mediana a grande escala. Mediante un enfoque incremental y de ciclos iterativos, se busca desarrollar la aplicación de manera eficiente, asegurando que cada iteración aporte mejoras significativas y responda adecuadamente a las necesidades de los usuarios. En este capítulo se detalla el proceso de planificación, desarrollo y evaluación continua que permitió la evolución progresiva del sistema, garantizando su fiabilidad y eficacia en la gestión segura y eficiente de historias clínicas.

6.1. Planificación de entregas

Al momento de desarrollar un proyecto con un tiempo limitado disponible para su completación, es necesario tener objetivos claros a corto plazo que permitan iterar sobre el mismo e ir puliendo hasta obtener el resultado definido en la fase de diseño, en el caso del presente proyecto se siguieron los siguientes pasos:

Establecimiento de criterios para descomponer el sistema en incrementos entregables

El criterio utilizado para desglosar el sistema fue el de descomposición por aspectos. Este enfoque consiste en extraer funciones transversales o comunes a varios 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 durante el desarrollo.

División de la arquitectura del sistema en incrementos entregables

La arquitectura del sistema se dividió en incrementos funcionales que proporcionan valor al cliente. Estos incrementos, utilizables de manera parcial y temporal por los usuarios, permiten una implementación gradual y una validación continua de las funcionalidades desarrolladas. En cada incremento se listan los requisitos funcionales y no funcionales derivados de la lista del producto, asegurando que cumplan con el principio de modularidad y facilitando su desarrollo, prueba e integración.

Establecimiento de la duración fija de las iteraciones o *sprints*

Se determinó que la duración de cada iteración o sprint sería de dos semanas. Este período se eligió 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, facilitando la adaptación a cambios y mejoras continuas en el desarrollo del sistema.

Estimación del esfuerzo de cada incremento entregable

El equipo de desarrollo estimó el esfuerzo en horas por miembro del equipo requerido para producir cada incremento entregable. Esta estimación se basó en la cantidad y complejidad de los requisitos asociados a cada incremento, así como en 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 incluye los componentes de las diferentes capas de la aplicación relacionados con 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 proporciona a los usuarios el primer acercamiento al uso de la aplicación, permitiéndoles crear sus perfiles y acceder a la aplicación, teniendo acceso a sus datos almacenados.

Entrega 2

El segundo incremento integra 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 añade 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 relacionada con 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, así como las especialidades en las que los pacientes tienen historias clínicas asociadas, incluyendo sus pruebas unitarias, de integración y de sistema en conjunto con la documentación relacionada. Al integrarse con los

incrementos anteriores, las funcionalidades de este incremento permiten a los pacientes, médicos y enfermeros navegar hacia las historias clínicas y los datos a los que tienen acceso.

Entrega 4
El cuarto incremento comprende 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 relacionados con las evoluciones, órdenes y análisis de sus historias clínicas, limitados por las capacidades de su rol. Al igual que en los incrementos anteriores, estos 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 con el dominio de la aplicación.

6.2. Desarrollo y operación de incrementos

Definido el plan de entregas e incrementos, se procedió al desarrollo y operación de estos. Las siguientes etapas se repitieron iterativamente para cada incremento, realizando una demostración del software al final de cada uno 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

Para incorporar requisitos nuevos o pendientes de la iteración anterior se revisó la lista del producto (*product backlog*), al comienzo de cada iteración. Esto implicó analizar las necesidades del cliente, priorizar las funcionalidades y definir los criterios de aceptación para cada requisito.

Se seleccionaron las historias de usuario relacionadas con las funcionalidades del incremento actual de la lista del producto para ser implementadas en la iteración. Estas historias se plasmaron en la lista de iteraciones o *sprint backlog*, la cual fue elaborada o actualizada en conjunto con el equipo de desarrollo.

Para cada historia de la lista de iteraciones, 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 del incremento

Se revisó y actualizó el diagrama de casos de uso para detallar los requisitos funcionales a implementar en el incremento después de la planificación actual. 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.

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

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, fueron revisados y actualizados al final del proceso.

6.2.3. Diseño detallado del incremento

Durante esta fase de las iteraciones, se revisó y ajustó la arquitectura de la aplicación para incluir los cambios correspondientes al incremento actual. Esto implicó examinar los componentes de la Vista Arquitectónica relacionados con el incremento, definir las interfaces entre componentes y especificar las tecnologías para su implementación.

Además, se optimizó el grafo de navegación de la interfaz gráfica de usuario (IGU) para mejorar la navegación de los usuarios a través de las funcionalidades del sistema. Se aseguraron flujos de navegación intuitivos y consistentes, y se

refinaron los mock-ups de cada página de la IGU del incremento. Esto incluyó mejorar la distribución de elementos gráficos, la organización de la información y la aplicación de la paleta de colores definida.

Posteriormente, se diseñaron los elementos gráficos de la IGU asociados al incremento, como íconos, botones, menús y otros elementos visuales. Se tuvo en cuenta la estética general del software, la ergonomía y la accesibilidad, y se actualizó el Diseño Detallado de la IGU con las especificaciones técnicas necesarias.

Se desarrolló el esquema conceptual y parcial del incremento, representando la estructura lógica de los datos utilizados. Dicho esquema presenta las entidades, atributos y relaciones entre ellas, y se integró al esquema conceptual general de la base de datos. Esto implicó mapear entidades y relaciones del esquema conceptual general con las del esquema parcial, actualizar el esquema lógico de la base de datos y, finalmente, actualizar el esquema físico que representa cómo los datos se almacenarán en el sistema de almacenamiento físico y la cadena de bloques.

Con la estructura visual y conceptual de la aplicación definida, se diseñaron los programas del incremento, seleccionando una función específica de la lista de iteraciones para un análisis y diseño detallado. Para cada función, se elaboró una Realización de Caso de Uso, siguiendo un enfoque independiente de la plataforma. Este análisis incluyó los siguientes pasos:

- ***Identificación de clases de negocio***: Se identificaron las clases de negocio participantes en la función seleccionada, representando los objetos del dominio del problema.
- ***Definición de atributos, relaciones y responsabilidades***: Se definieron atributos, relaciones y responsabilidades para cada clase de negocio.
- ***Elaboración del diagrama de clases participantes***: Se creó un diagrama de clases para visualizar las clases de negocio, sus atributos, relaciones y responsabilidades. Se identificaron los servicios que cada clase debe ofrecer y las operaciones que pueden realizar sobre sus atributos y relaciones.

Se procedió a actualizar el comportamiento de las clases participantes, incorporando las operaciones identificadas. Finalmente, se elaboró un diagrama de secuencias para describir el comportamiento de la función seleccionada, mostrando la interacción entre las clases participantes y las operaciones invocadas en cada paso del caso de uso.

El siguiente paso fue refinar la realización del caso de uso teniendo en cuenta la plataforma de software elegida y los requisitos arquitectónicos del proyecto, incluyendo:

- ***Análisis de requisitos arquitectónicos***: Se analizaron los requisitos arquitectónicos asociados al caso de uso seleccionado.
- ***Identificación de mecanismos de implementación***: Se identificaron mecanismos de implementación para traducir los requisitos arquitectónicos en código fuente.
- ***Identificación de patrones de diseño***: Se identificaron patrones de diseño aplicables al programa.

Se identificaron las clases de interfaz para manejar la interacción usuario-programa, las clases de control para manejar el flujo de control del programa, y las clases que gestionan aspectos de persistencia, seguridad y manipulación de errores. Posteriormente, se refinó el diagrama de secuencias incorporando las nuevas clases identificadas, mostrando una visión completa de la interacción entre clases y operaciones invocadas en cada paso del caso de uso.

6.2.4. Codificación, Pruebas e Integración de los Incrementos

En cada iteración, se seleccionaron las RCU a implementar de acuerdo con el plan de desarrollo y las prioridades del proyecto, revisando y actualizando los diagramas de secuencias y de clases. Se diseñaron pruebas unitarias y de integración para cada RCU, verificando el funcionamiento correcto de cada módulo y su interacción.

Se codificaron los programas del incremento utilizando técnicas de programación convencionales, empezando por el esqueleto del programa y definiendo el orden de codificación, integración y prueba de los métodos de las clases del programa. Se revisó y actualizó el diseño de pruebas de métodos para

todas las clases del programa, cubriendo todos los escenarios posibles y garantizando la calidad del código.

La calidad del código se verificó utilizando herramientas de análisis estático y revisiones de código, corrigiendo errores, problemas de estilo y posibles vulnerabilidades. Se refactorizó el código fuente de los módulos que lo requerían, mejorando su legibilidad, mantenibilidad y extensibilidad. Se integraron los programas del incremento y se realizaron pruebas de integración, corrigiendo errores hasta obtener resultados satisfactorios.

Para verificar el correcto funcionamiento del incremento en conjunto con el resto del sistema se llevaron a cabo pruebas a nivel de sistema, cubriendo todos los escenarios posibles y asegurando la calidad del sistema en su totalidad.

6.2.5. Verificación y Validación de los Incrementos

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

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

Se llevó a cabo una reunión de retrospectiva al final de cada iteración para revisar el proceso de desarrollo del incremento, identificando lecciones aprendidas, buenas prácticas y áreas de mejora para las próximas iteraciones.

6.2.6. Cierre de proyecto

Siguiendo los principios de Blue Watch, se evalúa la ejecución del proyecto en esta etapa. Aunque el entregable es un prototipo, se ajustó la evaluación a las características específicas de esta fase. Se realizó un análisis exhaustivo de los resultados obtenidos durante las pruebas para determinar el cumplimiento de los objetivos planteados y emitir conclusiones y recomendaciones sólidas.

Capítulo 7

Conclusiones y recomendaciones

En Venezuela es muy común ver historias clínicas almacenadas en carpetas y portafolios físicos. Cada vez que es necesaria hacer la búsqueda de la información de los pacientes en las instituciones de salud que implementan estos sistemas se ralentiza el proceso de atención médica para todos los usuarios, además de que en caso de que sea necesario que el paciente sea atendido en otra institución de salud no hay una forma eficiente, y a veces en lo absoluto, por parte del paciente de llevar esos documentos originales a dicha institución.

Se buscó abordar estas problemáticas diseñando y desarrollando un producto mínimo viable de un sistema de gestión de historias clínicas basado en la tecnología de cadena de bloques que permite el almacenamiento cronológico,

inmutable y electrónico de dichos registros de salud, lo cual hace el proceso mucho más rápido y eficiente.

Durante la fase de diseño se tomaron en cuenta varias tecnologías de blockchain y la implementada, llamada Hyperledger Fabric, demostró ser adecuada y extremadamente efectiva para abordar las dificultades expresadas anteriormente.

Tras el desarrollo del sistema se hicieron pruebas automáticas y manuales tanto de integración como unitarias para verificar que los objetivos de seguridad y efectividad se cumplían, las cuales fueron exitosas en su totalidad, demostrando que un despliegue posible de este sistema en el sistema de salud de la región en la que se realizó este estudio probaría beneficio no solo para el personal que trabaje allí sino también para todos sus usuarios.

Se realizaron pruebas de interoperabilidad, que simulaban el uso del sistema para el acceso simultáneo entre distintas instituciones a una única historia clínica de un paciente sobre la cual dicho paciente tenía control de acceso total sobre la misma.

Los objetivos planteados al principio de este proyecto fueron cumplidos en su totalidad, y para trabajos futuros se podría ahondar en otros aspectos como:

- Utilizar una red de cadena de bloques pública como la red de Ethereum, buscando enfatizar en la democratización y distribución del sistema de almacenamiento de historias de usuario.
- Diseñar e implementar un sistema de almacenamiento externo de archivos para los registros de salud que mantengan los mismos principios de seguridad, privacidad y eficiencia.
- Desplegar la aplicación en un sistema de salud funcional para evaluar su efectividad en el campo.
- Agregar más capaz de seguridad a la comunicación entre los clientes y el servidor de la lógica de negocio utilizando algún protocolo seguro de capa de red (SSL, TSL, DTSL, etc)

En conclusión, podemos decir que un sistema de gestión de historias clínicas implementado en un sistema de salud en la región sería una gran mejoría a los sistemas que ya están en pie en todos los aspectos, y podrían dar paso no solo a un mejor servicio, sino a innovación y avances tecnológicos dentro de los procesos de salud existentes.

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 las historias clínicas de pacientes?
- ¿Alguna vez has tenido problemas sobre las historias clínicas con algún paciente?
- ¿Qué te parece que está bien en la forma actual en que manejas las historias clínicas de pacientes?
- ¿Qué te parece que está mal en la forma actual en que manejas las historias clínicas de pacientes?

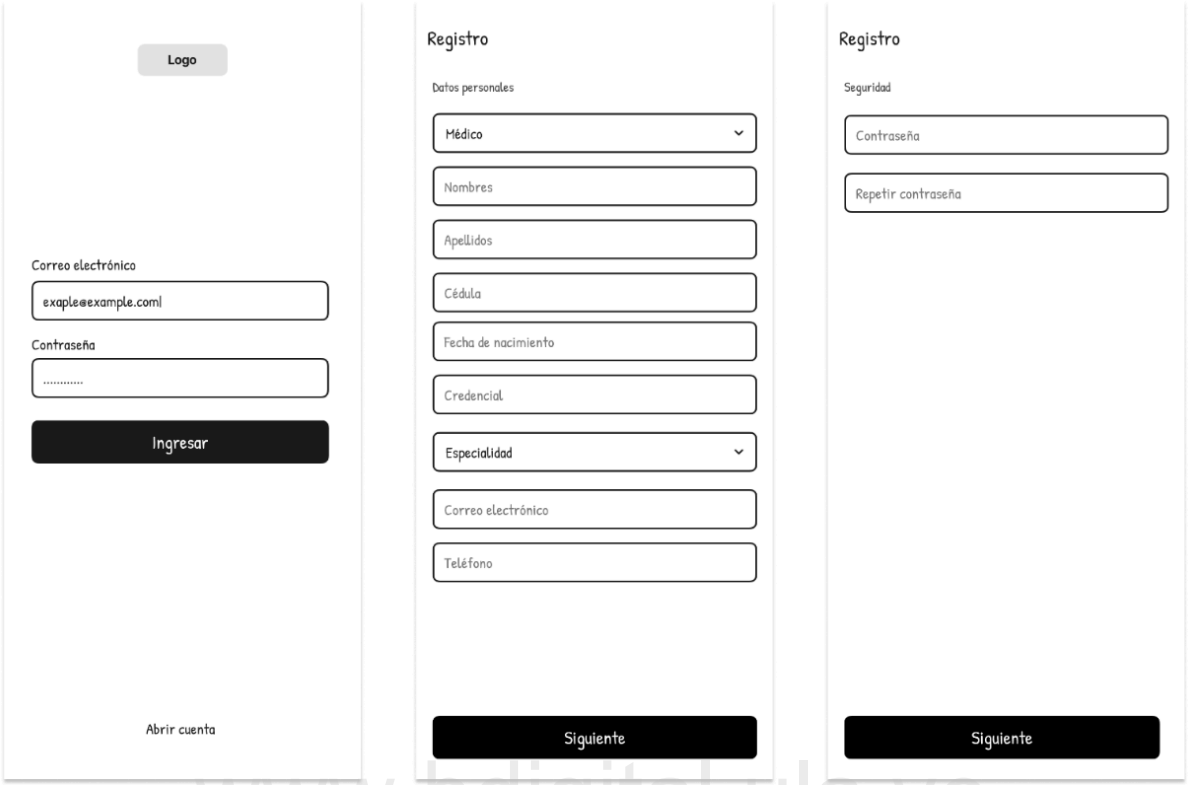
- ¿Qué le mejorarías a la forma actual en que manejas las historias clínicas de pacientes?
- ¿Te sería factible una aplicación de escritorio, tablet o móvil para manejar las historias clínicas de tus pacientes?

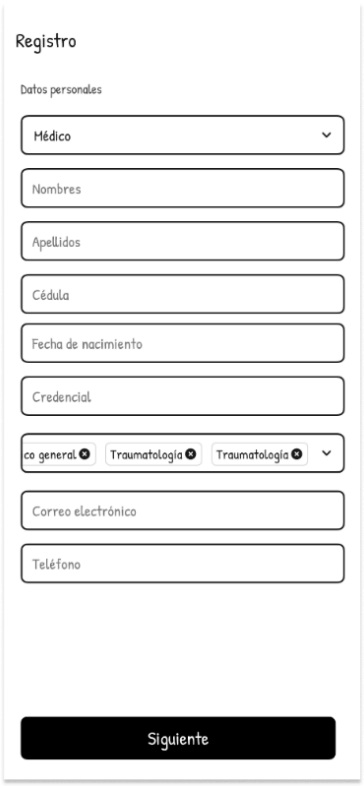
Entrevistas a los pacientes

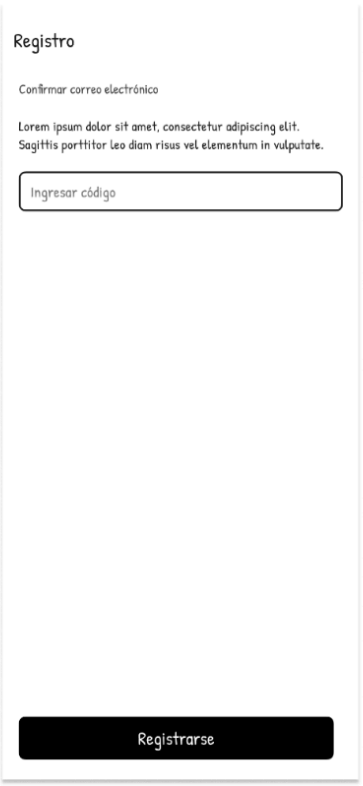
Las preguntas principales en las entrevistas a los pacientes regulares del IAHULA que conformaron la muestra de este proyecto fueron las siguientes:

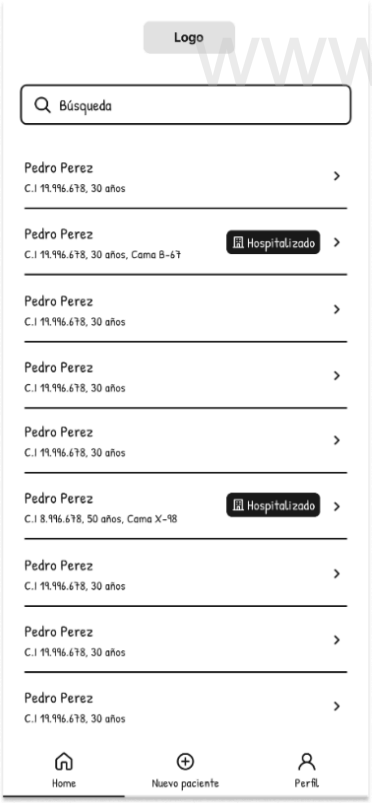
- ¿De qué forma te identificas cuando vas al médico? Es decir, ¿Qué información personal te solicita el médico para sus registros?
- ¿Alguna vez has tenido problemas con el acceso a tu historia clínica en alguna institución de salud?
- ¿Qué mejorarías de la forma en que te identificas en las instituciones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

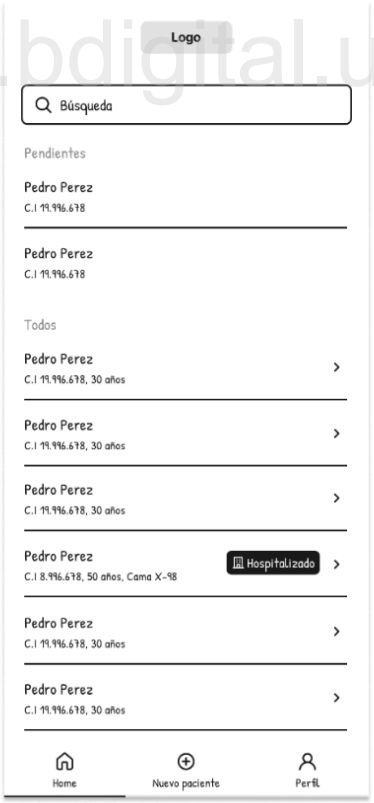
Interfaz gráfica

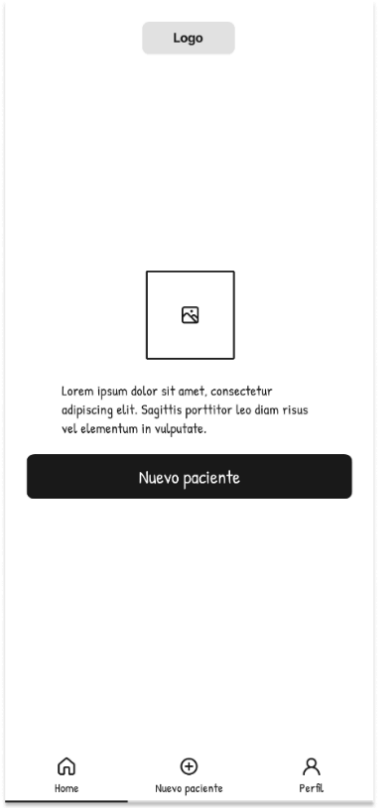


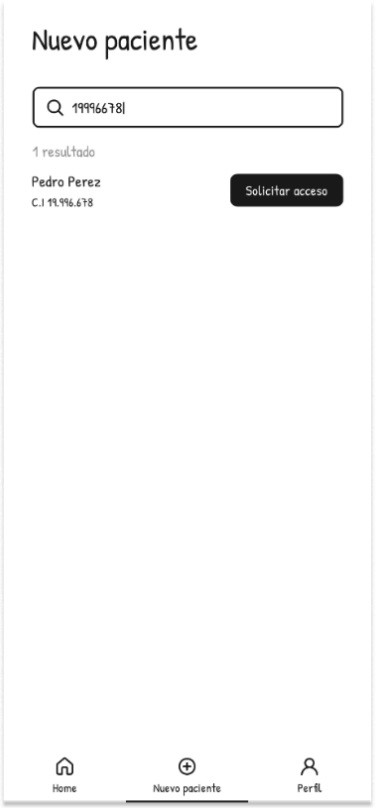


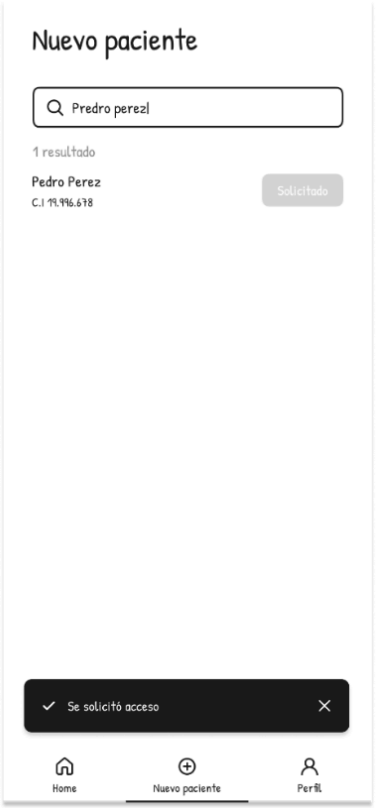


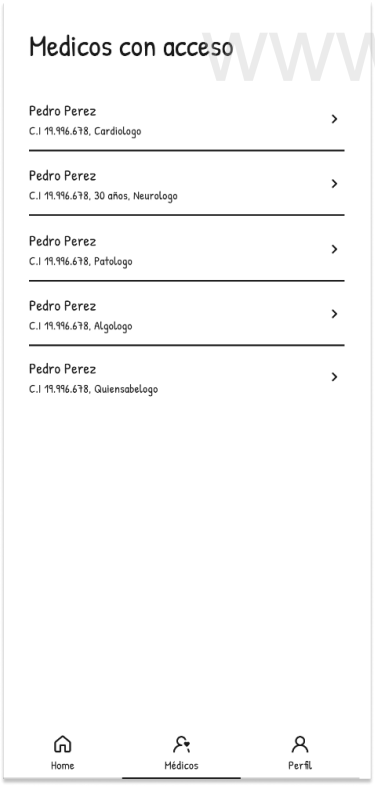


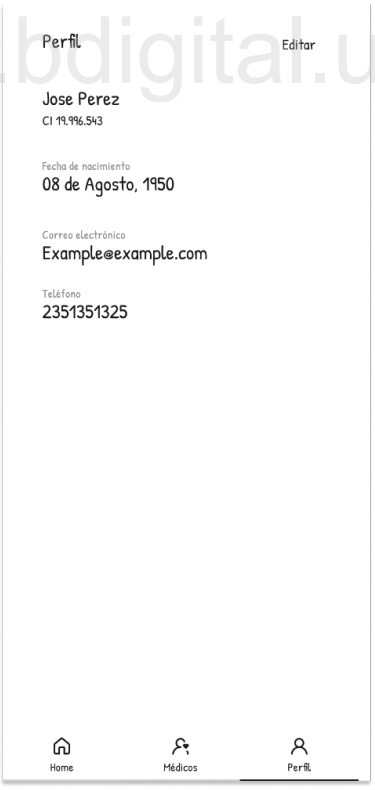


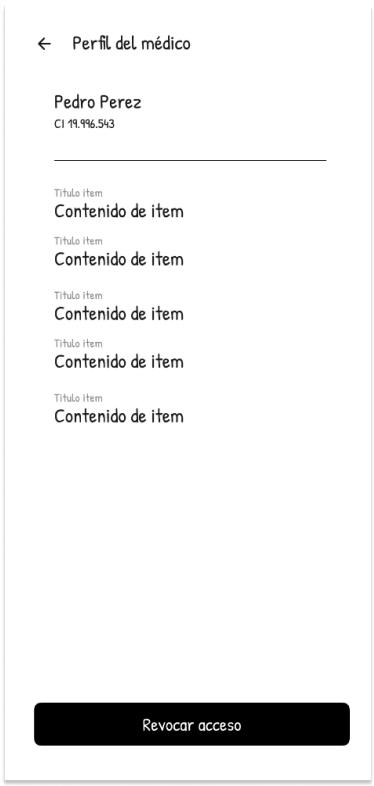


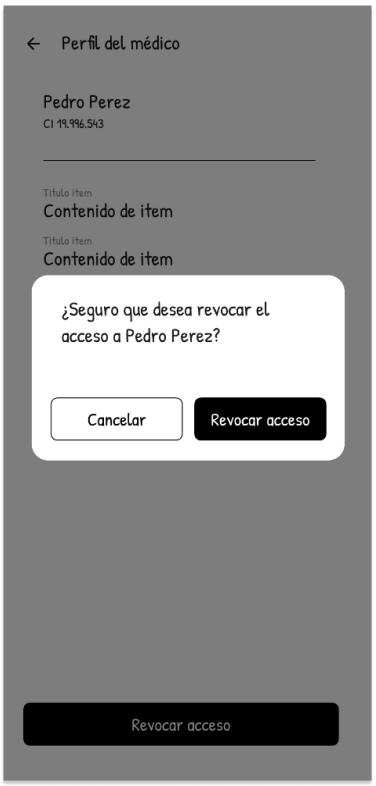


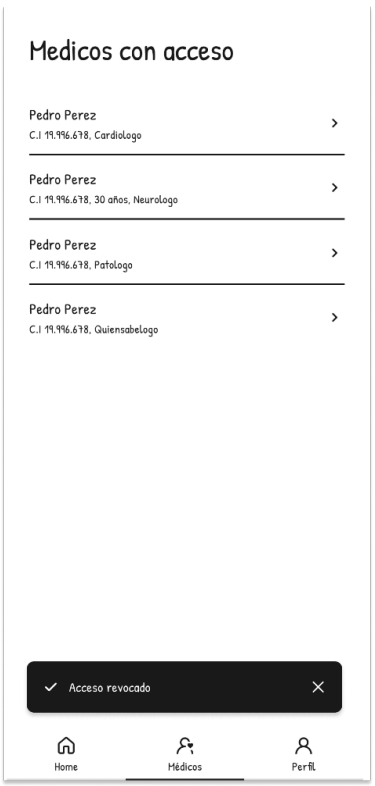


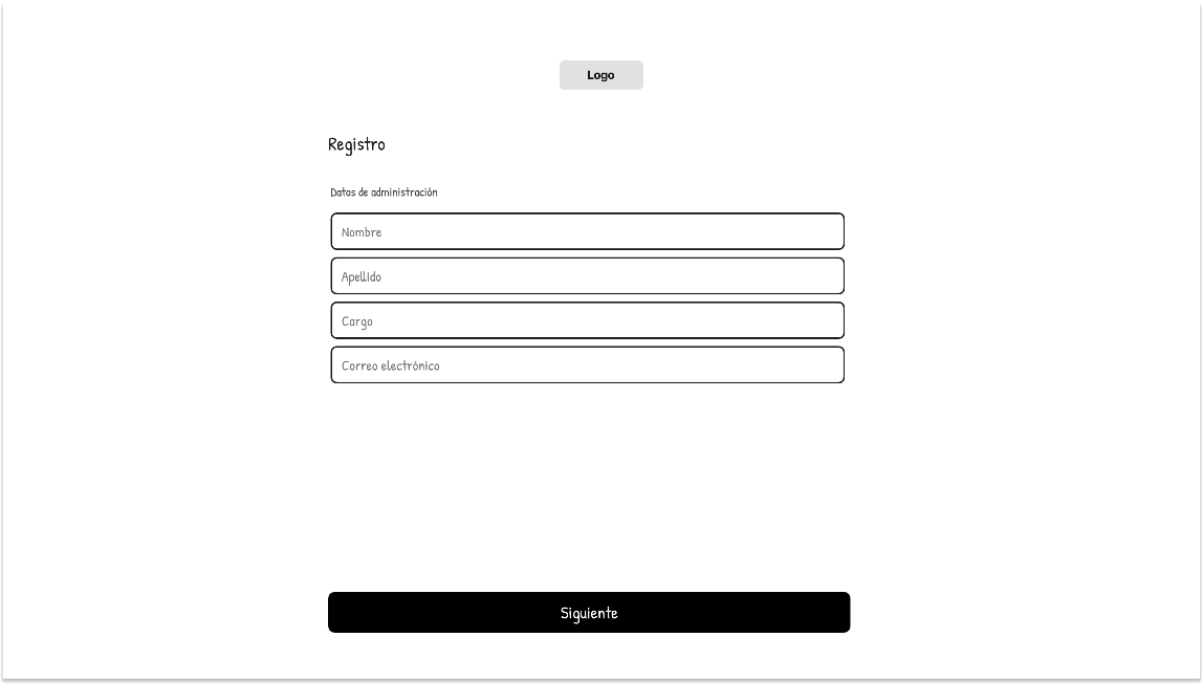


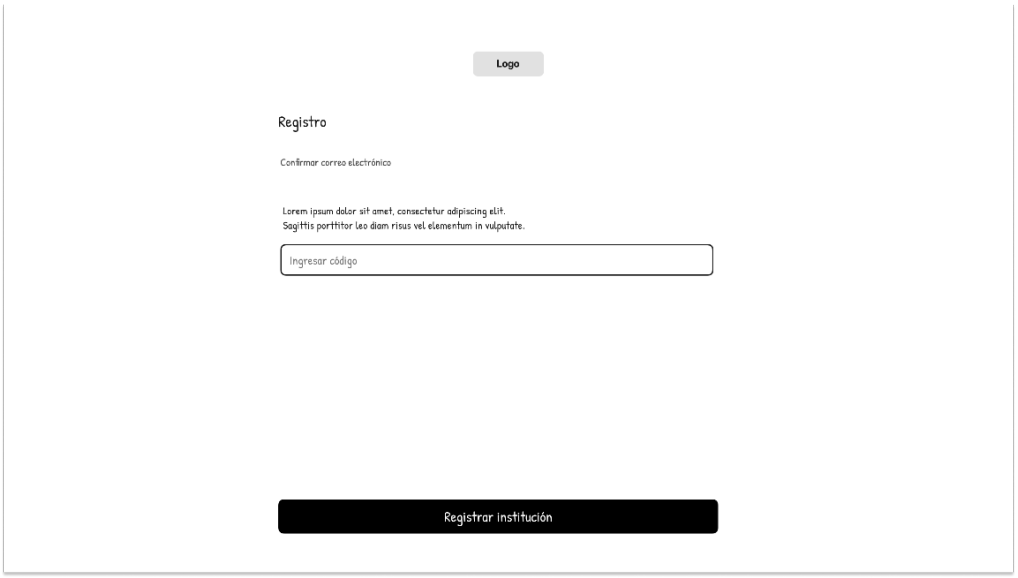


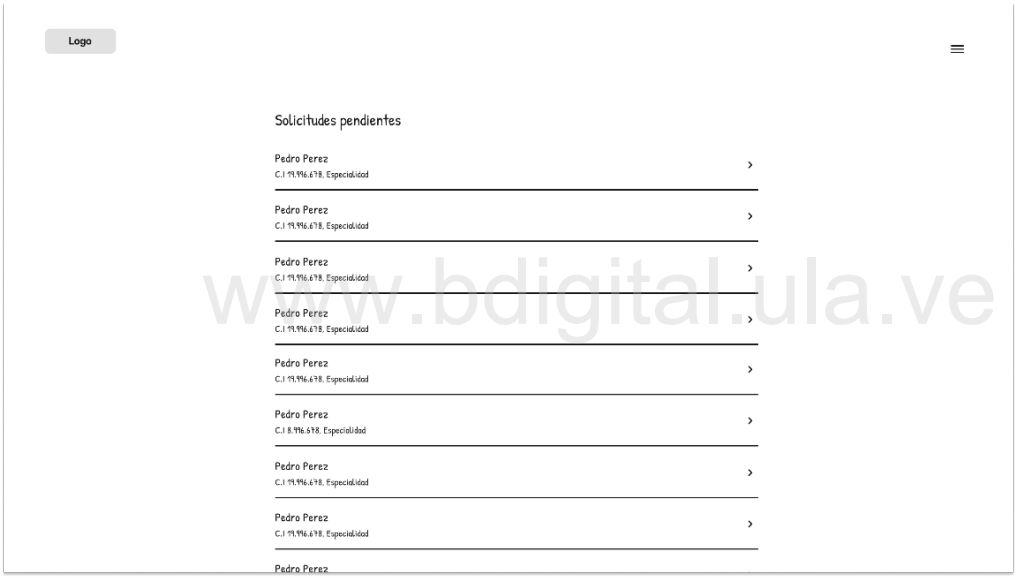


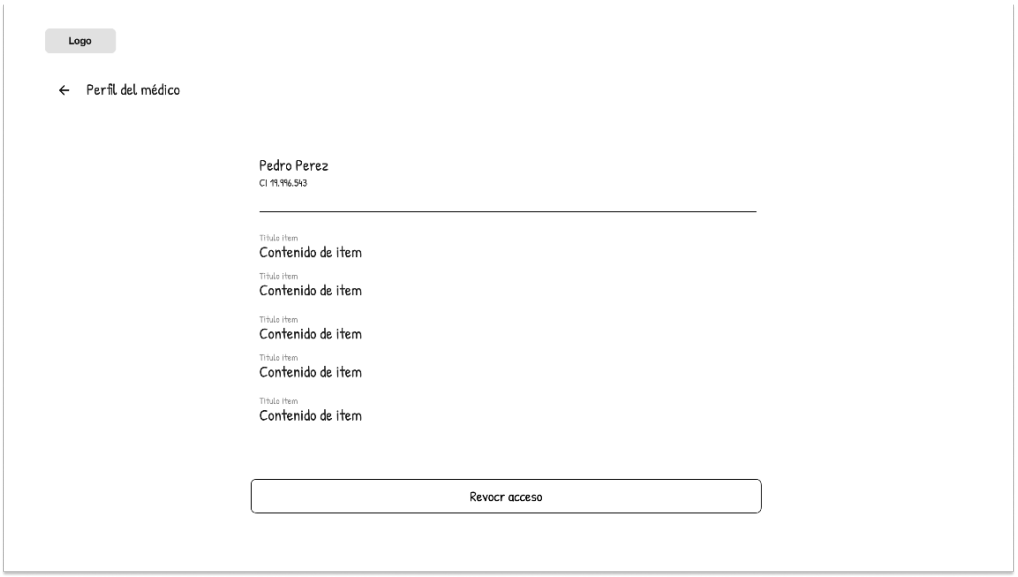


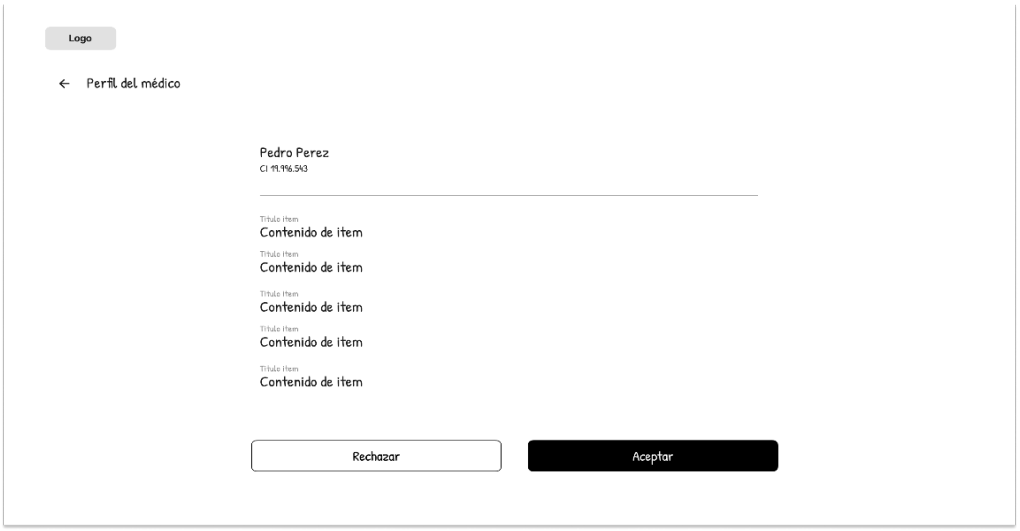


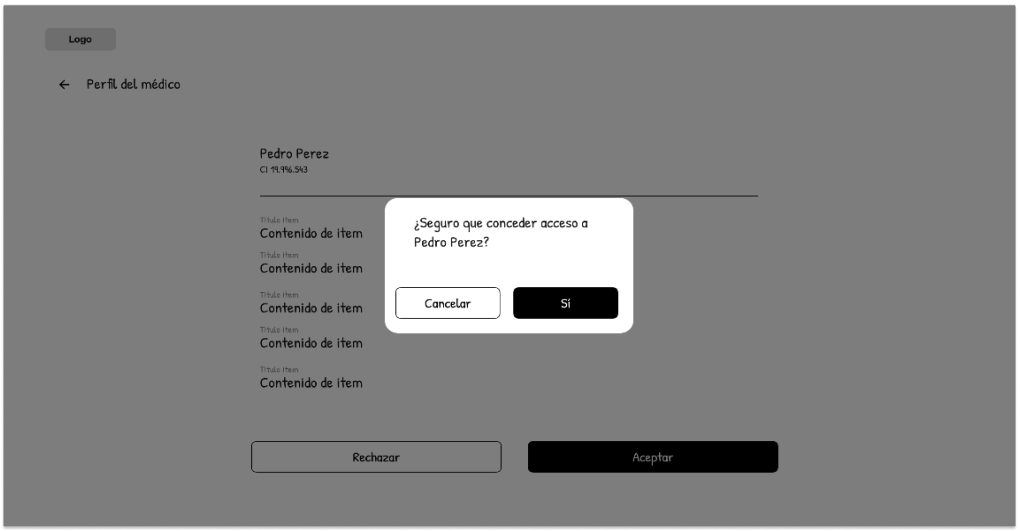












Bibliografía

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

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). https://doi.org/10.1007/s10916-018-1121-4

Dennis, A., Barbara Haley Wixom, David Paul Tegarden, & Seeman, E. (2015). System analysis & design : an object-oriented approach with UML (5th ed.).

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

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. 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

Hasselgren, A., Kralevska, K., Gligoroski, D., Pedersen, S. A., & Faxvaag, A. (2020). Blockchain in healthcare and health sciences—A scoping review. International Journal of Medical Informatics, 134(1), 104040. https://doi.org/10.1016/j.ijmedinf.2019.104040

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

Kothari, C. R. (2004). Research Methodology : Methods & Techniques (2nd ed.). New Age International (P) Ltd., Publishers, Cop. https://ds.amu.edu.et/xmlui/bitstream/handle/123456789/9330/Research%20Methodology.pdf?sequence=1&isAllowed=y

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/

Martin, R. C. (2017). Clean Architecture. Prentice Hall.

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

Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system Bitcoin: A Peer-to-Peer Electronic Cash System. Bitcoin. org. Disponible en https://bitcoin.org/en/bitcoin-paper

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

Schwaber, K. (2015). Agile project management with scrum. Microsoft.

Sharma, P., Jindal, R., & Borah, M. D. (2020). Blockchain Technology for Cloud Storage. ACM Computing Surveys, 53(4), 1–32. https://doi.org/10.1145/3403954

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

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

Vladimir Khorikov. (2020). Unit testing : principles, practices, and patterns. Manning.

Zhou, L., Wang, L., & Sun, Y. (2018). MIStore: a Blockchain-Based Medical Insurance Storage System. Journal of Medical Systems, 42(8). https://doi.org/10.1007/s10916-018-0996-4