

ELABORACIÓN DE UN REPOSITORIO WEB DE TÉCNICAS QUIRÚRGICAS

Santiago Ramírez Valencia.

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO.
FACULTAD DE INGENIERÍA.
TECNOLOGÍA EN DESARROLLO DE SOFTWARE
MEDELLÍN
2023**

**ELABORACIÓN DE UN REPOSITORIO WEB DE TÉCNICAS
QUIRÚRGICAS**

ii

Santiago Ramírez Valencia.

Trabajo de grado para optar al título:
Tecnólogo en desarrollo de software.

Asesores:

Rubén Darío Fonnegra Tarazona
Jaime Ernesto Soto

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO.
FACULTAD DE INGENIERÍA.
TECNOLOGÍA EN DESARROLLO DE SOFTWARE
MEDELLÍN
2023

1.	Planteamiento del Problema	1
1.1.	Descripción	1
1.2.	Formulación	3
2.	Justificación	4
3.	Objetivos	5
3.1.	Objetivo general.....	5
3.2.	Objetivos específicos.....	5
4.	Marco Teórico.....	6
4.1.	Procedimientos quirúrgicos	6
4.2.	Plataforma web	6
4.3.	Bases de datos no relacionales (NoSQL).....	7
4.4.	El Teorema de CAP	10
4.5.	MERN Satack web development.....	13
4.5.1.	MERN Frontend.....	14
4.5.2.	MERN Backend.....	15
5.	Metodología	17
5.1.	Tipo de proyecto	17
5.2.	Método	17
5.3.	Instrumentos de recolección de información	19
5.3.1.	Fuentes primarias.....	19
5.3.2.	Fuentes secundarias.....	19
6.	Resultados	20
6.1.	Estructuración de la información	20
6.2.	Base de datos no relacionales	21
6.3.	Creación base de datos en mongoDB	23
6.4.	Interfaz de usuario.....	29
6.4.1.	Maqueta de la interfaz de usuario.....	29
6.4.2.	Creación interfaz de usuario.....	30
6.5.	Integración	34
6.6.	Validación.....	36
6.7.	Control de versiones	40
7.	Recomendaciones	41
8.	Conclusiones.....	43
	Anexo A.....	47
	Anexo B.....	53

Lista de tablas

Tabla 1. Descripción de vistas y detalles de la maqueta de diseño web 30

Figura 1. Clasificación de las bases de datos según el teorema CAP.....	12
Figura 2. MERN Stack.....	14
Figura 3. Elementos clave.....	20
Figura 4. Esquema base de datos.....	23
Figura 5. Esquema colección cirugías.....	24
Figura 6. Esquema colección usuarios.....	25
Figura 7. Creación de bases de datos.....	26
Figura 8. Creación de colecciones.....	27
Figura 9. Estructura de los documentos de la colección de cirugías.....	27
Figura 10. Estructura de los documentos de la colección usuario.....	28
Figura 11. Mapa de navegación del aplicativo.....	32
Figura 12. Vista administrador elaborada con React y Tailwind CSS.....	33
Figura 13. Diagrama de arquitectura de implementación.....	35
Figura 14. Validación de autenticación de usuario.....	37
Figura 15. Validación del formulario de cirugía.....	38
Figura 16. Validación del formulario de usuario.....	39
Figura 17. Vista del usuario.....	47
Figura 18. Vista del login.....	47
Figura 19. Vista del Administrador en el apartado Admin.....	48
Figura 20. Vista del Administrador en el apartado Usuarios.....	48
Figura 21. Vista del Administrador en el formulario Cirugías.....	49
Figura 22. Vista del Administrador en el formulario Usuarios.....	50
Figura 23. Vista del Administrador al consultar una cirugía.....	51
Figura 24. Vista del Administrador al consultar un usuario.....	52
Figura 25. Vista del usuario con React y Tailwind.....	53
Figura 26. Vista del login con React y Tailwind.....	53
Figura 27. Vista del Administrador en el apartado Admin con React y Tailwind.....	54
Figura 28. Vista del Administrador en el apartado Usuarios con React y Tailwind.....	54
Figura 29. Vista del Administrador en el formulario Cirugías con React y Tailwind.....	55
Figura 30. Vista del Administrador en el formulario Usuarios con React y Tailwind.....	56
Figura 31. Vista del Administrador al consultar una cirugía con React y Tailwind.....	57
Figura 32. Vista del Administrador al consultar un usuario con React y Tailwind.....	58

Stack MERN: Se refiere al conjunto de tecnologías utilizadas en el desarrollo de aplicaciones web modernas y escalables, incluyendo MongoDB, Express, React y Node.js.

Repositorio web: Se refiere a una plataforma en línea donde se almacenan y comparten recursos y archivos, permitiendo el acceso y colaboración por parte de un equipo de trabajo.

Planeación de técnicas quirúrgicas: Hace referencia al proceso de determinación de las técnicas y procedimientos quirúrgicos más adecuados para el tratamiento de algún paciente en particular, de acuerdo a la opinión de los médicos especialistas.

Aplicación: Empleo de una cosa o puesta en práctica de los procedimientos adecuados para conseguir un fin.

Desarrollo de Software: Es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (software).

Front-End: Parte visual que normalmente ven los usuarios, lo que se carga cuando una persona ingresa a una página web, logos, textos, enlaces entre otros y que permiten la interacción con el sitio.

Back-End: Forma de llamar a toda la arquitectura que procesa los datos que vienen desde el Front-End, es decir los que interactúan con los usuarios.

GitHub: Plataforma para el desarrollo colaborativo el cual permite tener control de versionamiento ya que todo el código se almacena en un repositorio en la nube.

Website: Lugar donde se almacenan archivos electrónicos de manera que puedan ser accedidos desde cualquier lugar del mundo a través de Internet.

The project involves the development of a web platform for managing and accessing information on surgical procedures. The platform will allow users to search for and access specific surgical techniques, as well as detailed information on each procedure. The objective is to provide a useful and efficient tool for healthcare professionals and students in the field of surgery.

The following activities were carried out during the project:

Requirements analysis: The needs and requirements of the platform were identified, including the information structure, required functionalities, and system characteristics.

Information gathering: Primary and secondary sources of information, such as books and web portfolios, related to surgical procedures and specific techniques were sought and collected.

Architecture design: The architecture of the platform was designed, considering aspects such as the database, user interface, and technologies to be used.

Technology selection: Appropriate technologies were selected for platform development, including the use of a NoSQL database like MongoDB and the React JavaScript library for the frontend.

Bibliography structuring in the database: The collected information was organized and structured in the database, following an established process to facilitate future updates.

Interface development: The frontend of the application was developed using React, creating a dashboard with dynamic filters to facilitate procedure search.

Database integration with the interface: The connection between the MongoDB database and the user interface was established, ensuring that the information is properly presented on the platform.

Platform validation: Validation tests were conducted to ensure the proper functioning of the application.

The project's outcome highlights the following:

Information structuring: Key fields for collecting information on surgical procedures were defined, establishing a guide for the database.

NoSQL database: MongoDB was chosen as the database due to its scalability, flexibility in data schema, support for unstructured data, and high performance in reads and writes.

Creation of the MongoDB database: The "Surgical_repository" database was created with two main collections: "surgeries_collection" for surgical procedures and "user_collection" for users and roles.

User interface: The layout and creation of the user interface were done using React and Tailwind CSS, following a defined structure in the mockup.

Integration: The different parts of the application were integrated using the MERN (MongoDB, Express, React, Node.js) stack, along with tools like Mongoose, Cloudinary, Axios, and Formik to facilitate the development and operation of the platform.

El proyecto consiste en el desarrollo de una plataforma web para la gestión y acceso a información sobre procedimientos quirúrgicos. La plataforma permitirá a los usuarios buscar y acceder a técnicas quirúrgicas específicas, así como a información detallada sobre cada procedimiento. El objetivo es proporcionar una herramienta útil y eficiente para profesionales de la salud y estudiantes en el campo de la cirugía.

En el transcurso del proyecto se llevaron a cabo las siguientes actividades:

Análisis de requisitos: Se identificaron las necesidades y requisitos de la plataforma, incluyendo la estructura de la información, funcionalidades requeridas y características del sistema.

Recopilación de información: Se buscaron y recopilaron fuentes primarias y secundarias de información, como libros y portafolios web, relacionadas con procedimientos quirúrgicos y técnicas específicas.

Diseño de la arquitectura: Se realizó un diseño de la arquitectura de la plataforma, considerando aspectos como la base de datos, la interfaz de usuario y las tecnologías a utilizar.

Selección de tecnologías: Se seleccionaron las tecnologías adecuadas para el desarrollo de la plataforma, incluyendo el uso de una base de datos NoSQL como MongoDB y la biblioteca de JavaScript React para el frontend.

Estructuración de la bibliografía en la base de datos: Se organizó y estructuró la información recopilada en la base de datos, siguiendo un proceso establecido para facilitar futuras actualizaciones.

Desarrollo de la interfaz: Se desarrolló el frontend de la aplicación utilizando React, creando un dashboard con filtros dinámicos para facilitar la búsqueda de procedimientos.

Vinculación de la base de datos con la interfaz: Se estableció la conexión entre la base de datos MongoDB y la interfaz de usuario, asegurando que la información se presente correctamente en la plataforma.

Validación de la plataforma: Se realizaron pruebas de validación para garantizar el correcto funcionamiento de la aplicación.

En el resultado del proyecto se destaca lo siguiente:

Estructuración de la información: Se definieron los campos clave para la recopilación de información sobre procedimientos quirúrgicos, estableciendo una guía para la base de datos.

Base de datos NoSQL: Se optó por utilizar MongoDB como base de datos debido a su xi
escalabilidad, flexibilidad en el esquema de datos, soporte para datos no estructurados y alto
rendimiento en lecturas y escrituras.

Creación de la base de datos en MongoDB: Se creó la base de datos
"Surgical_repository" con dos colecciones principales: "surgeries_collection" para los
procedimientos quirúrgicos y "user_collection" para los usuarios y roles.

Interfaz de usuario: Se realizó la maquetación y creación de la interfaz de usuario
utilizando React y Tailwind CSS, siguiendo una estructura definida en la maqueta.

Integración: Se integraron las diferentes partes de la aplicación utilizando el stack MERN
(MongoDB, Express, React, Node.js), junto con herramientas como Mongoose, Cloudinary,
Axios y Formik para facilitar el desarrollo y funcionamiento de la plataforma.

La planificación y ejecución de cirugías es un proceso altamente complejo que requiere de un cuidadoso seguimiento de cada paso y procedimiento realizado durante el acto quirúrgico. Actualmente, los estudiantes y profesionales del área de la salud se enfrentan a desafíos significativos al intentar recopilar la información necesaria para sus turnos quirúrgicos. El acceso a los detalles precisos y actualizados sobre las cirugías se vuelve una tarea ardua debido a la distribución dispersa de la información en diferentes fuentes.

En este contexto, se plantea el presente trabajo de grado, que tiene como objetivo desarrollar una plataforma web utilizando el Stack MERN (MongoDB, Express, React y Node.js) para abordar esta problemática. La plataforma proporcionará una solución integral que permitirá subir y organizar toda la información relacionada con una cirugía, junto con imágenes y otros datos relevantes del procedimiento quirúrgico.

El proyecto se centrará en la creación de una interfaz intuitiva y amigable, con características clave que faciliten la gestión eficiente de la información. El administrador de la plataforma podrá crear nuevas cirugías, editar la información existente y eliminar los registros obsoletos. Además, se implementará un sistema de inicio de sesión con jerarquías de permisos, lo que garantizará un acceso controlado a los diferentes usuarios.

El principal problema que se aborda con esta herramienta es la falta de centralización de la información sobre las cirugías. En el sistema actual, los cuadros quirúrgicos se envían a los

estudiantes el día anterior, dejándoles un tiempo limitado para buscar y recopilar toda la información necesaria. Esto dificulta la preparación adecuada y puede afectar negativamente el desempeño y la comprensión de los procedimientos quirúrgicos. xiii

La plataforma propuesta permitirá superar estas limitaciones, ya que brindará a los usuarios la capacidad de acceder a un repositorio centralizado de información sobre las cirugías. Esto significa que los estudiantes y profesionales médicos podrán encontrar fácilmente los detalles precisos y actualizados de cada procedimiento quirúrgico, lo que mejorará su preparación, conocimiento y rendimiento durante los turnos quirúrgicos.

A lo largo de este trabajo de grado, se abordarán los desafíos técnicos asociados con el desarrollo del proyecto, así como las consideraciones de seguridad y escalabilidad necesarias para garantizar el buen funcionamiento de la plataforma.

En resumen, este proyecto tiene como objetivo principal desarrollar una plataforma web utilizando el Stack MERN, que centralizará la información relacionada con las cirugías, proporcionando la capacidad de cargar, editar y acceder de manera eficiente a los detalles de los procedimientos quirúrgicos. Al hacerlo, se busca resolver el problema actual de la dispersión de información y la falta de tiempo para la preparación adecuada de los estudiantes y profesionales médicos. A través de este enfoque innovador, se espera mejorar la calidad de la atención médica y el rendimiento en los entornos quirúrgicos.

1. Planteamiento del Problema

1.1.Descripción

En la actualidad existe una problemática con los estudiantes de diferentes carreras del área de la salud (Medicina, Instrumentación Quirúrgica y Enfermería) ya que inician en los primeros semestres una asignatura denominada práctica académica, en la cual se debe presentar a un hospital universitario y realizar una jornada laboral completa, en la cual poco a poco y según su desempeño, van adquiriendo responsabilidades y aportando en cada uno de los procedimientos que se van realizando durante el día. Es en dicha práctica donde se presenta la problemática planteada, ya que en muchas ocasiones no se tiene conocimiento del procedimiento que se va a realizar ya que no se tiene experiencia y no se ha cursado la materia referente a esa especialidad; y el estudiante debe estudiar dicha técnica en tiempos inminentes. Esta información es usualmente difícil de conseguir y no es de libre acceso, asunto que además se complica con la alta variabilidad geográfica contenida en esta información. Por ejemplo, la técnica de laparoscopia es ejecutada de manera diferente en EEUU, Japón y Europa (1.1. 1.2, 1.3), y puede variar inclusive entre escuelas de medicina del mismo país. Otra forma de buscar la información es en los libros que se recomienda en la universidad y por último son las notas de clase, pero como fue mencionado anteriormente, es posible que el estudiante aún no haya cursado y obtenido los conocimientos necesarios y no tenga suficiente material o tiempo para preparar de manera correcta su turno.

Actualmente existe una aplicación muy completa en Google play store y APP Store (Dispositivos Android y Apple) de forma gratuita (2), llamada touch surgery, desarrollada por Digital Surgery (Metronic) y lanzado el 2 de diciembre del 2013, aplicación en inglés que brinda un repositorio aproximado de 150 simulaciones de procedimientos quirúrgicos; para acceder a su plataforma web debe ser miembro de una institución. Aun así, hay especializaciones dentro de la aplicación que no cuentan con mucha información, por ejemplo, técnicas quirúrgicas. En diferentes especialidades se encuentran muy pocas técnicas, por ejemplo, en Urología la app solo cuenta con cuatro técnicas quirúrgicas, en una especialidad donde se pueden encontrar más de 70 técnicas quirúrgicas. (3)

También podemos encontrar aplicaciones como Atlas de anatomía humana que brinda información detallada del cuerpo humano, pero tienen un costo en su versión completa entre ellas: Atlas de anatomía humana 2023 (4.1), de igual forma aplicaciones web como: Lecturio (4.2) y Osmosis (4.3). Sin embargo, estas plataformas requieren normalmente de un pago o suscripción por lo que el acceso a la información está limitado y no es de dominio público, lo cual no resuelve el inconveniente planteado para los estudiantes.

Teniendo en cuenta lo anterior, se presenta una necesidad de desarrollar una plataforma web en español que ayude a docentes, estudiantes, practicantes y profesionales de la salud, a encontrar técnicas y procedimientos quirúrgicos con un alto grado de

confiabilidad y en un corto tiempo, para poder preparar de manera más óptima los turnos quirúrgicos.

1.2. Formulación

¿Es posible realizar un repositorio web con información de procedimientos técnicos e instrumental quirúrgico que realice filtros inteligentes para búsquedas de diferentes procedimientos quirúrgicos enfocados al Área de la Salud?

2. Justificación

El presente proyecto plantea desarrollar un repositorio web en español de técnicas quirúrgicas para estudiantes, docentes, practicantes y profesionales del área de la salud, que debe realizar búsquedas constantes en diferentes sitios (páginas web, libros, Apps móviles, y notas personales) para encontrar técnicas y procedimientos quirúrgicos que les ayude a la posterior preparación de una intervención quirúrgica. Esto permitirá reducir los tiempos de búsqueda del personal, incrementando la eficiencia y estandarización de la calidad de sus servicios. Es importante reducir dichos tiempos ya que el cronograma de cirugía es muy ajustado y deja poco margen de preparación.

3. Objetivos

3.1.Objetivo general

Diseñar y crear un repositorio web enfocado al área de la salud para brindar ayuda en la planeación de técnicas quirúrgicas a usar en los diferentes pacientes según el criterio de los médicos.

3.2.Objetivos específicos.

- Recopilar la documentación de los procedimientos quirúrgicos para la plataforma.
- Estructurar la base de datos usando la información recopilada en el objetivo específico anterior.
- Desarrollar la interfaz de usuario amigable teniendo en cuenta los tipos de información y consultas a realizar.
- Integrar la base de datos con la interfaz de usuario desarrollada anteriormente.
- Realizar pruebas de calidad y funcionalidad de la plataforma.

4. Marco Teórico

4.1.Procedimientos quirúrgicos

Se refiere a la operación instrumental, total o parcial, de lesiones causadas por enfermedades o accidentes, con fines diagnósticos, de tratamiento o de rehabilitación de secuelas.

Según la OMS, la cirugía mayor es “todo procedimiento realizado en quirófano que permite la incisión, escisión, manipulación o sutura de un tejido; generalmente requiere anestesia regional o general o sedación profunda para controlar el dolor” (OMS). Es decir, un procedimiento quirúrgico no se lleva a cabo únicamente en un quirófano, se realizan también en salas de procedimientos (de baja o mediana complejidad) de los servicios de urgencias (UCI), consultorios especializados de odontología, unidades de gastroenterología, servicios de radiología intervencionista, entre otros. (5) (Ministerio de Salud, 1994, Artículo 23)

4.2.Plataforma web

Un sitio web es la página principal o índice que alberga a un grupo de páginas web referentes a una temática o investigación en particular, relacionadas entre sí por hiperenlaces y son accesibles mediante una misma dirección. (Pacherres y Liliana, 2018).

Estas páginas pueden contener videos, textos, gráficos, contenido de sonido o audio y otros hipervínculos. En la actualidad hay una evolución de esta web en la que desarrolla aplicaciones que generan colaboración que reemplacen las de escritorio. Es una etapa que ha definido nuevos proyectos en Internet y está preocupándose por brindar mejores soluciones para el usuario final. La web ha tenido una evolución que es innegable ante la sociedad, las empresa y organizaciones la ven más necesaria en sus actividades diarias, esta plataforma hace uso de ancho de banda en el ambiente de las telecomunicaciones y se define el ancho de banda como la capacidad que tiene un medio de transmisión para enviar y recibir datos y es uno de los elementos más importantes para el acceso adecuado a internet y a la web, existe una relación directamente proporcional entre el ancho de banda, la velocidad y la cantidad de datos que pueden ser transmitidos a través de un sistema de telecomunicaciones. En resumen la web es un conjunto de ordenadores que conectados entre sí albergan una información que puede ser compartida por su ancho de banda, dando así la oportunidad de recibir una información visual o numérica según como lo requiera el usuario de la red.

4.3.Bases de datos no relacionales (NoSQL)

La respuesta a la necesidad de gestionar volúmenes masivos de información surge de la base de datos NoSQL, término acuñado a finales de los 90 y que engloba todas las tecnologías de almacenamiento estructurado que no cumplen el esquema relacional. (Yanes, 2012).

La cantidad de información manejada por comunidades, redes sociales, buscadores, y muchos otros proyectos en el ámbito de la Web 2.0 es abrumadora, lo que ha hecho que surjan nuevas arquitecturas de almacenamiento de información, que deben ser de alto rendimiento, escalables y distribuidas. (Yanes, 2012).

Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala. (Yanes, 2012).

Las bases de datos NoSQL utilizan una variedad de modelos de datos para acceder y administrar datos. Estos tipos de bases de datos están optimizados específicamente para aplicaciones que requieren grandes volúmenes de datos, baja latencia y modelos de datos flexibles, lo que se logra mediante la flexibilización de algunas de las restricciones de coherencia de datos en otras bases de datos. (Yanes, 2012).

Aunque esta tecnología surgió de unas necesidades muy concretas, su difusión y algunos proyectos para encapsular sus funcionalidades y hacerlas más amigables a desarrolladores acostumbrados a SQL está provocando que también se usen en proyectos

de pequeño tamaño, con lo que todo indica que a medio plazo convivirán con las bases de datos tradicionales independientemente del volumen de datos a gestionar.

Dentro de las plataformas NoSQL encontramos varios grupos (Yanes, 2012).:

- Basadas en clave/valor. Se almacenan valores asociados a una clave. Son sencillas y las de mayor rendimiento.
- Basadas en documentos. Son una particularización de las clave/valor, en las que el valor puede ser un documento. Permiten consultas complejas.
- Basadas en columna. Los valores se almacenan en columnas en lugar de filas. Son útiles cuando se gestionan datos agregados.
- Basadas en grafos. Las relaciones se tratan como un dato más.
- Basadas en objetos. Los datos son objetos y las relaciones punteros entre ellos. Permiten operaciones muy complejas pero suelen tener bajo rendimiento.
- Otras. Cubren necesidades muy específicas y tienen escasa implantación: basadas en tupla, multivaluadas, jerárquicas, etc.(Yanes, 2012).

4.4.El Teorema de CAP

Teorema CAP (Consistency Availability Partition tolerance): en el 2000, Eric Brewer propuso la idea de que en un entorno distribuido un sistema no puede mantener continuamente consistencia perfecta, disponibilidad y tolerancia partición simultáneamente (Castro Romero, 2012).

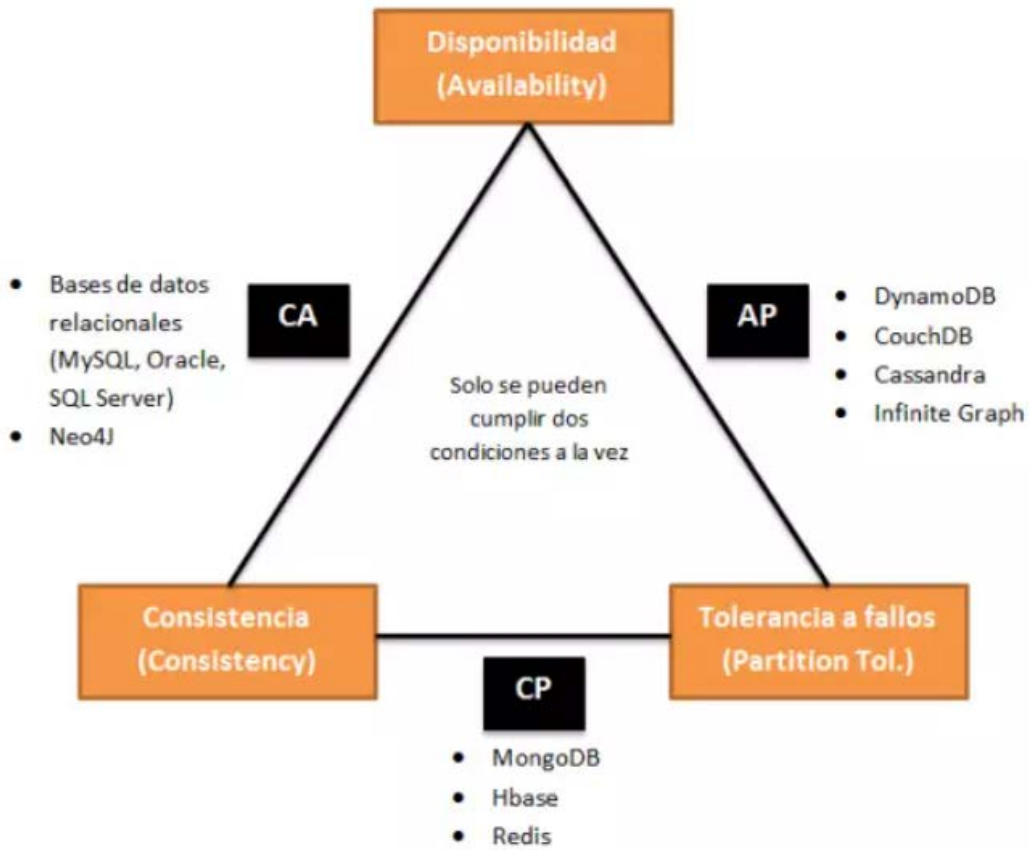
- Consistencia: los usuarios tienen la posibilidad de acceder simultáneamente a un mismo registro.
- Disponibilidad: la garantía de que cada solicitud recibe una respuesta.
- Tolerancia de reparto: el sistema sigue funcionando a pesar de la pérdida arbitraria de información.

Este punto es importante, pues permite entender que para ganar velocidad se debe sacrificar por lo menos una de las características mencionadas; pero el enfoque es que el sistema puede estar haciendo cambios entre las combinaciones de estos elementos, teniendo en cuenta las siguientes posibilidades:

- CP: el sistema ejecutará las operaciones de forma consistente, aunque se pierda la comunicación entre nodos (partición del sistema), pero no se asegura que el sistema responda (disponibilidad).
- AP: el sistema siempre responderá a las peticiones, aunque se pierda la comunicación entre nodos (partición del sistema). Los datos procesados pueden no ser consistentes.
- CA: el sistema siempre responderá a las peticiones y los datos procesados serán consistentes. En este caso no se permite una pérdida de comunicación entre nodos (partición del sistema). (Castro Romero, 2012)

Como se observa en la Figura 1 el Teorema de CAP ayuda a los desarrolladores a entender las limitaciones inherentes a los sistemas distribuidos y a tomar decisiones informadas al diseñar y construir un sistema distribuido que cumpla con los requisitos de negocio.

Figura 1. Clasificación de las bases de datos según el teorema CAP.



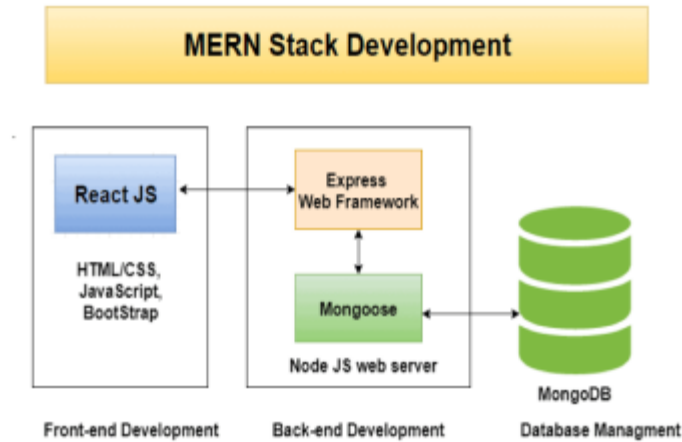
Nota. Genbeta. (2017). *clasificación de las bases de datos según el teorema CAP* [Image]. Recuperado de <https://www.genbeta.com/desarrollo/nosql-clasificacion-de-las-bases-de-datos-segun-el-teorema-cap>

4.5.MERN Satack web development

MERN es un diccionario usado para describir un conjunto de lenguajes basados en JavaScript, tecnologías que son usadas en las aplicaciones web. Esto fue diseñado con la idea de hacer procesos de desarrollo que juegan un rol importante en el proceso interno de una aplicación web. Actualmente los desarrolladores alrededor del mundo están trabajando para hacer una interfaz de usuario improvisada, que ayuda a una construcción de aplicaciones con pocos requerimientos y sin fechas límite. (Mehra, 2021)

La Figura 2 muestra el stack MERN, que es un conjunto de tecnologías de programación para la creación de aplicaciones web modernas y escalables. MERN representa: MongoDB, Express, React y Node.js.

Figura 2. MERN Stack



Nota. blog. (2020). *Illustrative diagram of the MERN stack* [Image]. Recuperado de <https://www.bocasay.com/how-does-the-mern-stack-work/>

4.5.1. MERN Frontend.

React js:

Es una librería de Javascript diseñada para construir una web con recursos. React tiene una memoria virtual llamada DOM, que está basada en la nube por lo que da unos resultados basados en un proceso de memoria interna que procesa la información en la nube sin necesidad de gastar recursos de la computadora. React está construida alrededor de los objetos y no de las plantillas como otros marcos de software. La sección puede ser creada con una función de la clase React object, el primer lugar donde se accede a la librería. (Mehra, 2021)

React.js performance

Cuando se mira la acción de React u otras estructuras, es importante recordar que no está todo el fin del Script de Javascript, la propuesta principal de la estructura de React es mejorar por medio de un desarrollo UI, que almacena altos niveles de eficiencia en algún entorno de requerimientos que puede ser reducido, en un nivel básico de alta acción, pero el resultado puede ser pobre y fácilmente formateado, por que todas las ubicaciones de JavaScript eventualmente son usados y basado en un API, que da un estructura y acción diferente en el código. (Mehra, 2021)

4.5.2. MERN Backend.

Node Js

Es una plataforma de software que ayuda a crear una red única de aplicaciones y eventos. Entre estos se incluye una construcción con librerías HTTP, que resguarda los desarrollos y creaciones en el mismo servidor web, creando así una sorprendente aplicación web, que controla otras aplicaciones web. (Mehra, 2021)

Express Js

Es un lenguaje basado en el software abierto pero que se escribe como Javascript, este usa componentes para Node Js, es fácil y rápido está diseñado usando el módulo de

middleware de Node.js llamado connect que implementa el módulo http. Permite definir rutas de la aplicación basadas en métodos HTTP y URL, incluye varios módulos de middleware que se pueden usar para realizar rutinas adicionales a pedido y respuesta, y fácil conexión con bases de datos como MySQL, MongoDB, Redis. (Sanchit, 2018)

Se puede decir que MERN contiene marcos extremadamente confiables para un front-end ultrarrápido en desarrollo, decidir cuál es mejor para alguien se reduce a un énfasis en la arquitectura, el código mantenibilidad, etc. (Sanchit, 2018)

5. Metodología

5.1. Tipo de proyecto

El proyecto es de tipo exploratorio con enfoque cualitativo, ya que se busca indagar en problemas poco estudiados, con una perspectiva innovadora y que sirva de base para futuros proyectos con un enfoque similar.

5.2. Método

1 Recopilación de la bibliografía: Se tendrá como referencia los siguientes libros de confianza:

- Pediatric Urology. Surgical Complications and Management.
- Cirugía General Caycedo 4ª Edición.
- Principles of Gynecologic Oncology Surgery.
- Netter Anatomy Flash Cards 5th Edition.
- Terminología Anatómica. Terminología Anatómica Internacional

También se realizarán búsquedas en las siguientes bibliotecas virtuales como:

- e-libro.
- EBSCO.
- Openlibra.
- BioMed Central.

2. Estructuración de la bibliografía en base de datos: Se recopilarán procedimientos y técnicas quirúrgicas, según tipo y especialidad para comenzar a poblar la base de datos con la información organizada y estructurada, dejando así un proceso establecido, rápido y eficaz a la hora de ingresar más contenido en un futuro.

3. Desarrollo de interfaz: En esta fase se desarrollará el front del aplicativo con un dashboard que realice filtros dinámicos para facilitar la búsqueda de diferentes procedimientos.

4. Vinculación de la BD + interfaz: se realizará la conexión del modelo de base de datos con la vista del Usuario, garantizando de esta manera que la información almacenada en la base de datos se presente de manera correcta y efectiva en la interfaz.

5. Validación de la Plataforma: se realizarán validaciones de casos de prueba previamente establecidos y pruebas de performance para garantizar el buen funcionamiento de la aplicación.

5.3.Instrumentos de recolección de información

5.3.1. Fuentes primarias.

Nuestras fuentes primarias son los siguientes libros:

- Pediatric Urology. Surgical Complications and Management.
- Cirugía General Caycedo 4ª Edición.
- Principles of Gynecologic Oncology Surgery.
- Netter Anatomy Flash Cards 5th Edition.
- Terminología Anatómica. Terminología Anatómica Internacional

5.3.2. Fuentes secundarias.

Nuestras fuentes secundarias son los siguientes portafolios web:

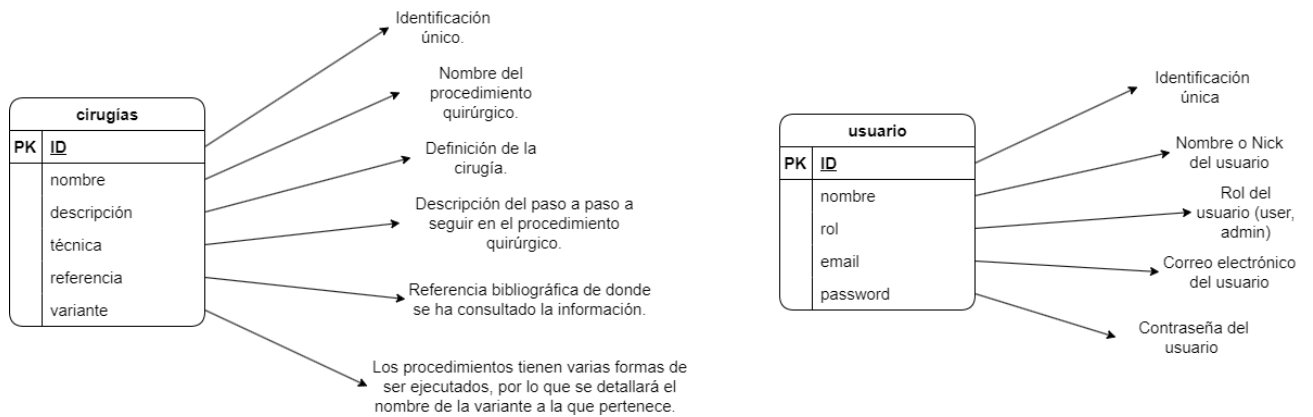
- e-libro.
- EBSCO.
- Openlibra.
- BioMed Central.

6. Resultados

6.1. Estructuración de la información

Al recopilar información sobre los procedimientos quirúrgicos, en la Figura 3 se detallan los siguientes campos claves:

Figura 3. Elementos clave



Nota. Diagrama que detalla los elementos clave que se encuentran en el proyecto.

Figura elaborada por el autor.

6.2.Base de datos no relacionales

Después de estructurar la información, existen varias razones por las cuales una base de datos NoSQL como MongoDB podría ser una buena opción para el caso que se presenta:

- **Escalabilidad horizontal:** Si se espera que el repositorio de información crezca de manera significativa con el tiempo, es posible que se requiera agregar más servidores para manejar la carga de trabajo. MongoDB es altamente escalable y puede distribuir la carga de trabajo entre múltiples servidores, lo que la hace una buena opción para casos donde se espera que la base de datos crezca significativamente con el tiempo.
- **Flexibilidad en el esquema de datos:** Ya que la aplicación en un futuro puede servir para diferentes áreas de la salud, es posible que la estructura de la información cambie con el tiempo, lo que significa que puede ser difícil predecir de antemano cómo deberá estructurarse la base de datos. MongoDB es una base de datos NoSQL orientada a documentos, lo que significa que puede almacenar datos con diferentes estructuras en el mismo conjunto de documentos. Esto brinda una mayor flexibilidad a la hora de manejar cambios en la estructura de la información.

- **Soporte para datos no estructurados:** Con el tiempo se espera que la base de datos contenga ilustraciones o videos interactivos, MongoDB podría ser una buena opción ya que es capaz de almacenar y manejar datos no estructurados, como archivos multimedia. Esto hace que sea más fácil almacenar y recuperar este tipo de datos en la base de datos.
- **Alto rendimiento en lecturas y escrituras:** Si se espera que la base de datos tenga un alto volumen de lecturas y escrituras, MongoDB puede ser una buena opción debido a su arquitectura distribuida y su capacidad para manejar transacciones. Esto significa que puede manejar una alta carga de trabajo sin afectar el rendimiento.

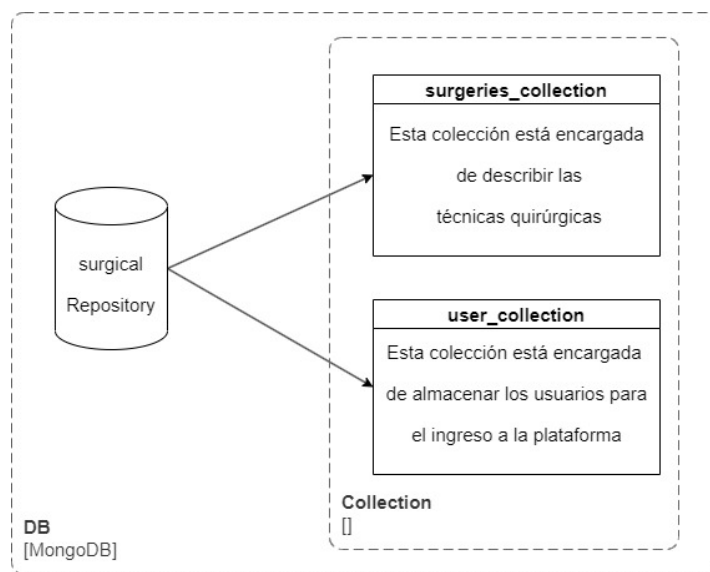
En conclusión, MongoDB es una buena opción para el caso presentado debido a su escalabilidad horizontal, flexibilidad en el esquema de datos, soporte para datos no estructurados y alto rendimiento en lecturas y escrituras

6.3. Creación base de datos en mongoDB

Se procede a crear la base de datos con el lenguaje y las diferentes herramientas de MongoDB.

Esquema General de base de datos: la base de datos contará primeramente con 2 colecciones, cirugías y usuarios.

Figura 4. Esquema base de datos

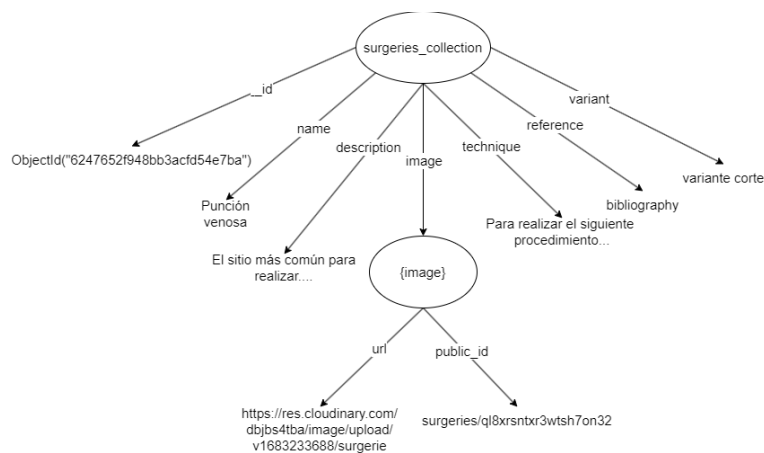


Nota. Esquema base de datos donde se detallan las colecciones user y surgeries.

Figura elaborada por el autor.

Esquema surgeries_collection: En la colección cirugías podemos ver que la mayoría de valores serán string, en la parte de imágenes vemos un nodo indicándonos que va a ser un objeto el cual almacenará la url y el id público que nos permitirá identificarla, de esta forma podremos crear el paso a paso en las técnicas de una manera más ordenada como se puede ver en la Figura 5.

Figura 5. Esquema colección cirugías

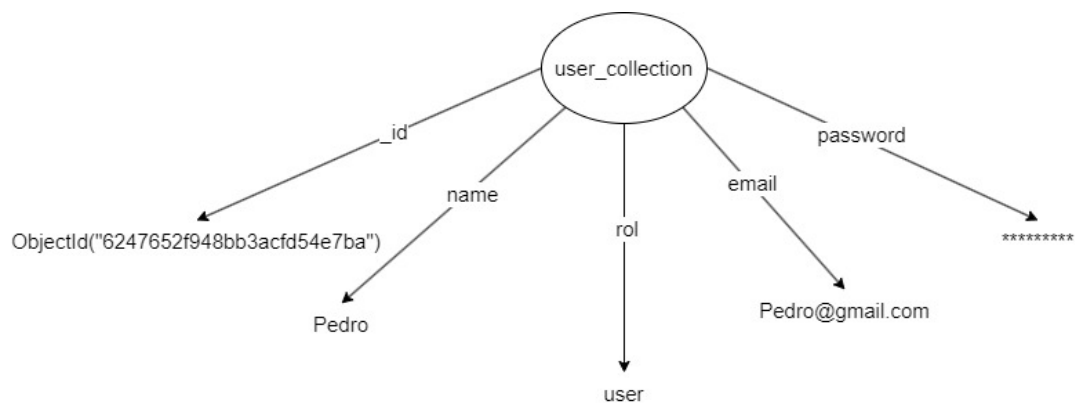


Nota. Esquema de normalización de la colección cirugías. Figura elaborada por el autor.

Como gestor de imágenes se utiliza cloudinary, ya que nos ofrece buena integración con lenguajes de programación, escalabilidad y optimización de imágenes. Además, ofrece también un paquete que cuenta con 24 Gigas de almacenamiento gratuito, lo cual es suficiente para el planteamiento inicial del proyecto. Es importante resaltar que la creación de las credenciales de acceso se realizó usando la cuenta de dominio institucional, por el planteamiento de uso académico inicial de la plataforma.

Esquema user_collection: Como se observa en la Figura 6 en la colección de usuarios, se ponen las credenciales, usuarios y roles que permitirán ingresar a la aplicación y configurar sus permisos.

Figura 6. Esquema colección usuarios

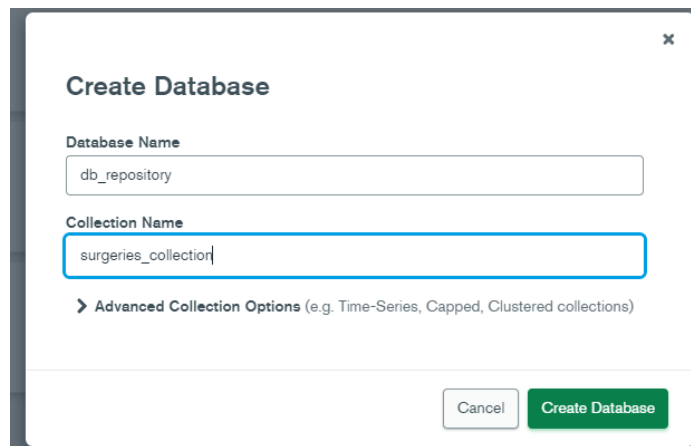


Nota. Esquema de normalización de la colección usuarios. Figura elaborada por el autor.

Teniendo en cuenta el esquema se procede hacer el repositorio principal y sus diferentes colecciones, como indica la Figura 4 este diseño de base de datos da una idea de como hacer los diferentes componentes de esta misma. MongoDB utiliza 3 interfaces importantes para el desarrollo de la base de datos, Compass, MongoDB Shell y Studio 3T, con esta IDE se crea la base de datos de una manera confiable y segura, evitando inconvenientes en el momento de hacer las consultas de los diferentes procesos quirúrgicos como se observa en la Figura 7. Adicionalmente MongoDB maneja Atlas, un componente que por medio de clusters conectados a la base de datos en creación generan

una alta disponibilidad de datos, evitando bloqueos con un solo proveedor y eliminando los problemas de latencia en las peticiones de los usuarios finales. A continuación, se muestra la creación de la base de datos “Surgical_repository” y sus diferentes colecciones: “surgeries_collection” y “user_collection”.

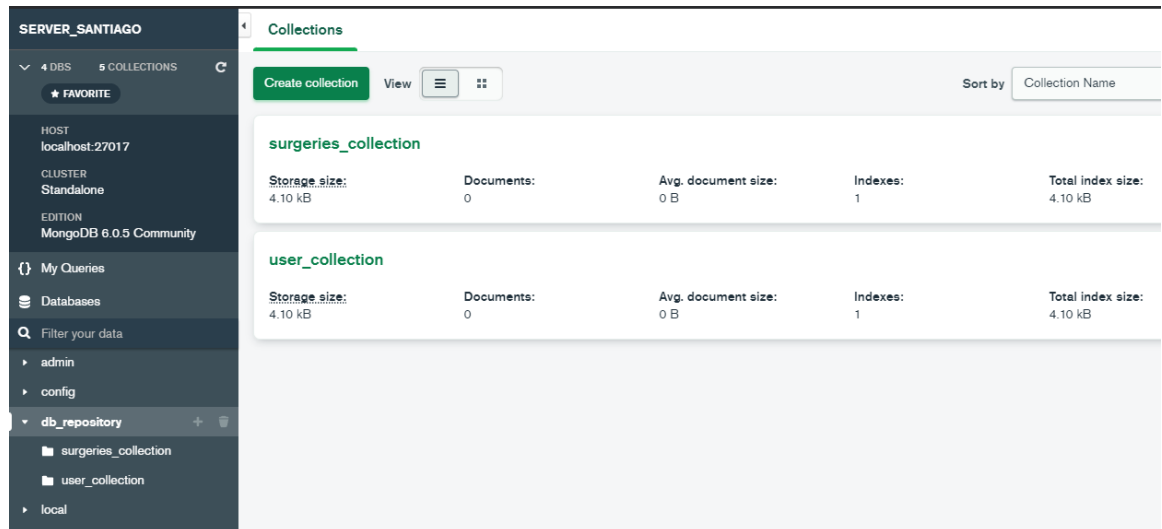
Figura 7. Creación de bases de datos



Nota. Creación de base de datos desde mongo compass. Figura elaborada por el autor.

El esquema de la Figura 4, da una idea de cómo sería el funcionamiento y distribución de la base de datos no relacional, esto facilita la visualización de los documentos y permite tener una guía muy específica para realizar la base de datos. Posteriormente en la IDE Compass se visualiza las colecciones de la base de datos que forma una estructura especial para este proyecto, con el cual se seguirá avanzando a las otras herramientas de desarrollo con la metodología MERN.

Figura 8. Creación de colecciones



Nota. Creación de colecciones cirugías y usuarios. Figura elaborada por el autor.

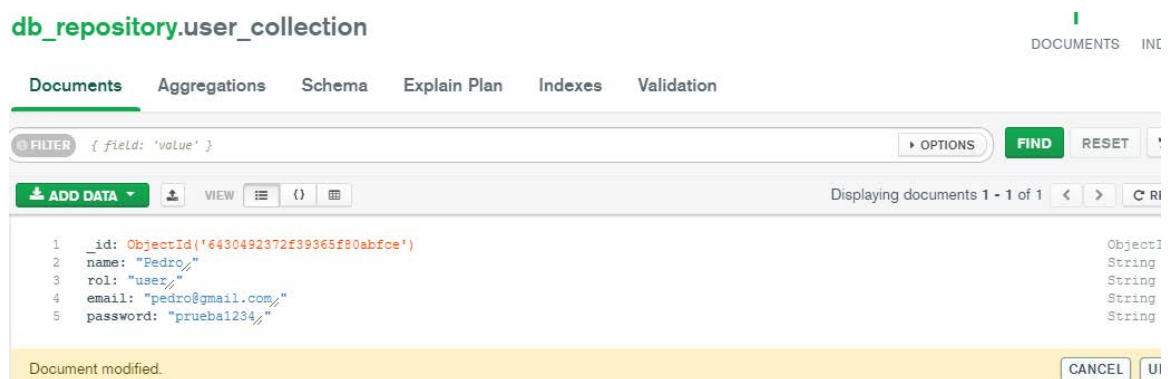
Figura 9. Estructura de los documentos de la colección de cirugías.



Nota. Se hacen los documentos dentro de la colección “surgeries_collection” desde mongo compas. Figura elaborada por el autor.

Teniendo creadas las colecciones como se ve en la Figura 8, y los documentos de la colección cirugías como se observa en la Figura 9, se procede a realizar los documentos de “user_collection”. En este se ponen las credenciales y usuarios que acceden a los documentos de técnicas quirúrgicas, como se indica en la Figura 10.

Figura 10. Estructura de los documentos de la colección usuario



Nota. Se hacen los documentos dentro de la colección “user_collection” desde mongo compas. Figura elaborada por el autor.

6.4. Interfaz de usuario

6.4.1. Maqueta de la interfaz de usuario.

La maquetación de un proyecto es de vital importancia, ya que nos ayudará a establecer la estructura, estilo y la funcionalidad de la aplicación, de acuerdo a ello se realizará la maquetación de la aplicación con el software figma que nos brinda las herramientas necesarias para la construcción de la estructura de la aplicación, en la Tabla 1 observamos las vistas del aplicativo y la descripción de cada una de ellas.

Tabla 1. Descripción de vistas y detalles de la maqueta de diseño web

Vista	Descripción
Vista del usuario.	La vista del usuario contará con una sidebar, una barra de búsqueda y el listado de cirugías. (véase Figura 17 en anexos).
Vista del login	La vista del login contará con sus respectivos campos para la validación del ingreso y un botón para volver. (véase Figura 18 en anexos).
Vista del administrador en el apartado admin.	La vista del Administrador del apartado Admin contará con sus respectivos campos para la realización del CRUD de las cirugías. (véase Figura 19 en anexos).
Vista del administrador en el apartado Usuarios.	La vista del Administrador del apartado Usuarios contará con sus respectivos campos para la realización del CRUD de los usuarios. (véase Figura 20 en anexos).
Vista del administrador en el formulario cirugías.	La vista del Administrador del formulario cirugías, contará con sus respectivos campos para la realización de la inserción de nuevas cirugías y la actualización de las mismas. (véase Figura 21 en anexos).
Vista del Administrador en el formulario Usuarios.	La vista del Administrador del formulario usuarios, contará con sus respectivos campos para la realización de la inserción de nuevos usuarios y la actualización de los mismos. (véase Figura 22 en anexos).
Vista del Administrador al consultar una cirugía.	La consulta de una cirugía en específico, contará con la la visualización del nombre, descripción, técnica, referencia y la imagen. (véase Figura 23 en anexos).
Vista del Administrador al consultar un usuario.	La consulta de un usuario en específico, contará con la visualización del nombre, roll, email y el password. (véase Figura 24 en anexos)

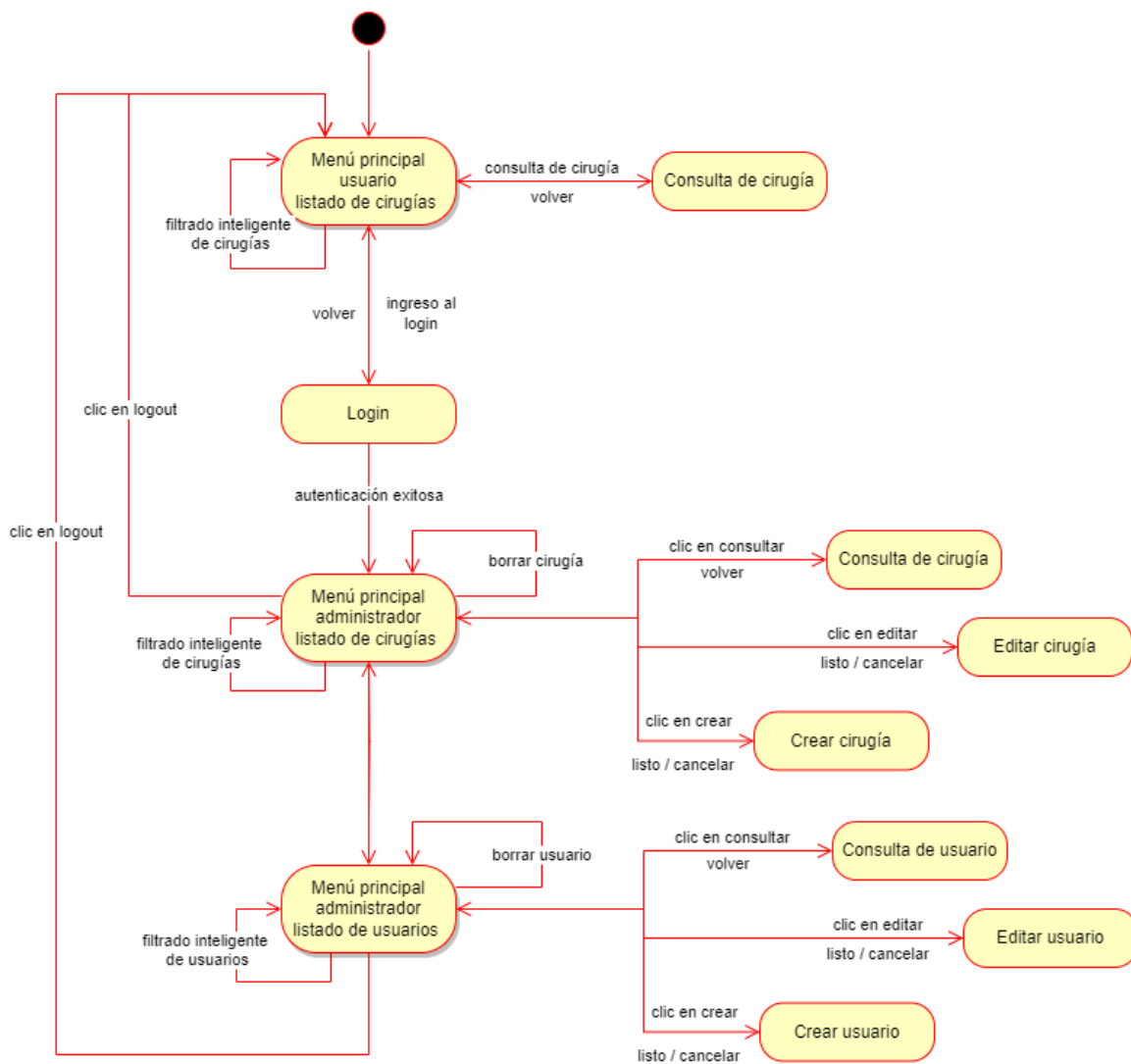
Nota. Esta tabla muestra la descripción detallada de las vistas de la aplicación; donde se especifican los campos y botones de cada una. Tabla elaborada por el Autor.

6.4.2. Creación interfaz de usuario.

Para la creación de la interfaz de usuario se utilizará la biblioteca de JavaScript React ya que nos ofrece una forma eficiente y modular de desarrollar aplicaciones web por medio

de componentes y el uso del Virtual DOM que permite mejorar el rendimiento y experiencia del usuario; aparte también cuenta una gran comunidad de desarrolladores y herramientas de terceros, lo que lo convierte en una excelente opción para la creación de aplicaciones modernas y escalables. Para detallar de forma más clara la funcionalidad y el comportamiento de la aplicación, se realizará un mapa de navegación. Véase Figura 11

Figura 11. Mapa de navegación del aplicativo.



Nota. Representación visual que muestra cómo los usuarios pueden navegar por las diferentes páginas o secciones de la aplicación web. Figura elaborada por el Autor.

La creación del front con react y tailwind conserva la misma estructura que se muestra en la maqueta véase Figuras 25 a la 32 en el apartado anexos. Por ejemplo, la Figura 12 con la vista del administrador.

Figura 12. Vista administrador elaborada con React y Tailwind CSS.



Nota. Vista administrador conservando los parámetros de la maqueta. Figura elaborada por el Autor.

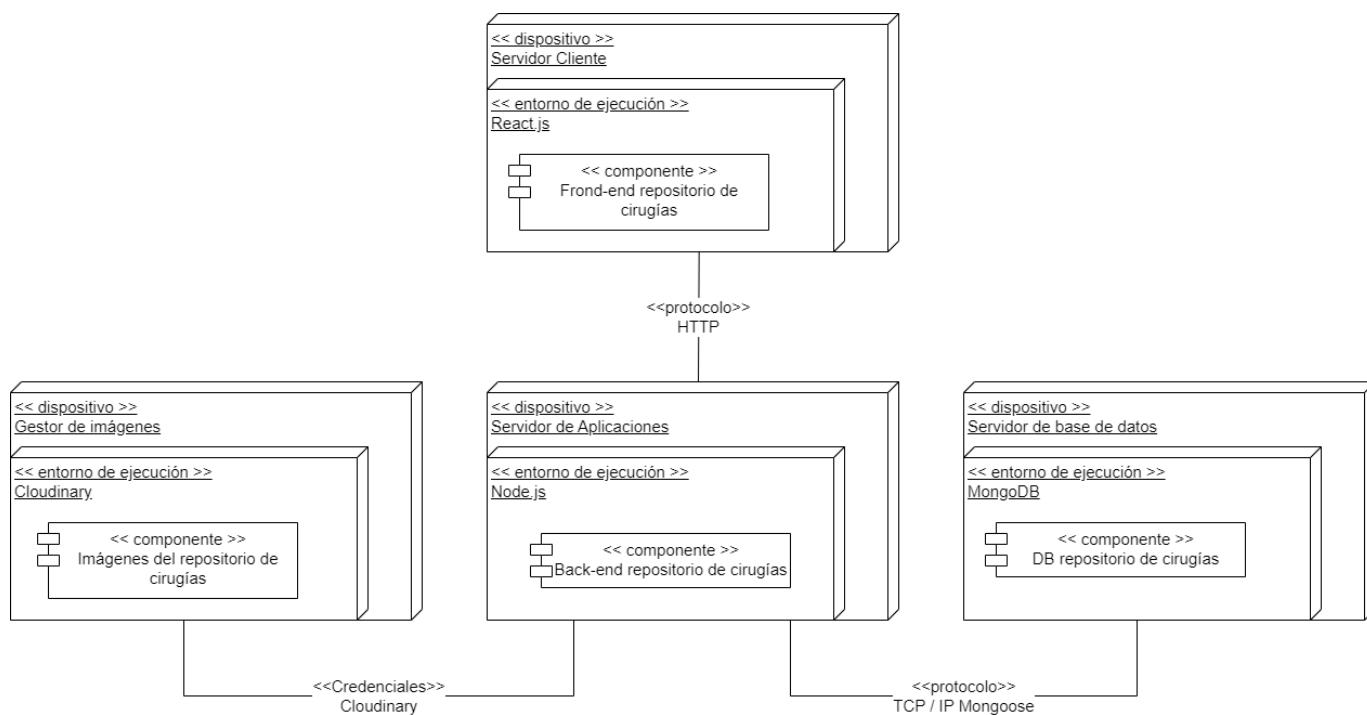
6.5.Integración

Para la integración de la aplicación web se utilizará el stack MERN ya que ofrece una solución completa para el desarrollo de aplicaciones web escalables, basándose en su mayoría en el lenguaje Javascript, ayudando a construir una aplicación funcional y de fácil despliegue, también se utilizarán algunas herramientas que nos facilitarán un desarrollo ágil y eficiente de la plataforma como lo son Mongoose, Cloudinary, Axios y Formik de la siguiente manera.

- El backend es el responsable de manejar la conexión a la base de datos utilizando Mongoose, que es una biblioteca de JavaScript que proporciona una interfaz para MongoDB. El backend también utiliza Cloudinary para el almacenamiento de imágenes.
- El frontend utiliza Axios para hacer solicitudes de API al backend. Axios es una biblioteca de JavaScript que permite a los navegadores web y a Node.js realizar solicitudes HTTP.
- Para la elaboración y validación de formularios de usuario y cirugías, se utiliza Formik. Formik es una biblioteca de React que facilita la creación de formularios con validación de entrada.

Esta integración permite que el frontend y el backend trabajen juntos para facilitar la creación, almacenamiento y validación de datos de usuario y cirugías. Mongoose y Cloudinary en el backend y Axios y Formik en el frontend como se puede observar en la Figura 13.

Figura 13. Diagrama de arquitectura de implementación.



Nota. Representación visual de los componentes y sistemas que se utilizan en la aplicación. Figura elaborada por el Autor.

6.6. Validación

Es importante realizar la validación de datos en cualquier proyecto, especialmente en aplicaciones web donde los usuarios pueden interactuar y enviar información. La validación de datos ayuda a garantizar la integridad de los datos ingresados, mejora la usabilidad de la plataforma y brinda una mejor experiencia de usuario. En el contexto de este proyecto, se utilizó Formik, una biblioteca de JavaScript, para facilitar la validación de formularios y asegurar la consistencia de los datos ingresados.

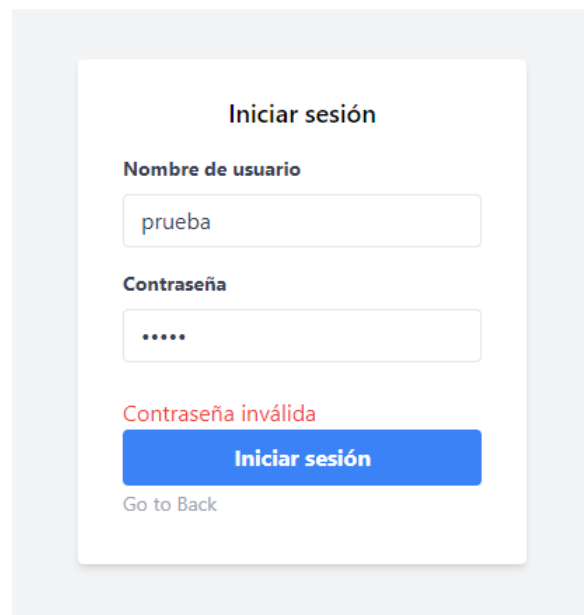
Hay varios beneficios de utilizar Formik para la validación de datos en un proyecto:

- **Facilidad de uso:** Formik simplifica en gran medida el proceso de validación de formularios en aplicaciones web. Proporciona una API intuitiva y fácil de entender que permite definir reglas de validación y mensajes de error de forma clara y concisa. Esto hace que sea más rápido y sencillo implementar la validación de datos en la plataforma.
- **Manejo automático de estados y eventos:** Formik se encarga automáticamente de la gestión de estados y eventos de formularios. Esto significa que no es necesario escribir código adicional para rastrear el estado de los campos del formulario, manejar eventos de cambio o controlar el envío de datos. Formik se encarga de estas tareas de manera

transparente, lo que simplifica el desarrollo y reduce la cantidad de código necesario.

- Validación en tiempo real: Formik permite realizar validaciones en tiempo real mientras los usuarios completan los formularios. Esto significa que los errores se muestran de inmediato a medida que se ingresan los datos incorrectos, brindando retroalimentación inmediata al usuario. Esta funcionalidad ayuda a prevenir errores y garantiza que los datos ingresados cumplan con los criterios requeridos como se puede observar en las Figuras 14 a la 16.

Figura 14. Validación de autenticación de usuario.



La imagen muestra un formulario web con el título "Iniciar sesión". Hay dos campos de entrada: "Nombre de usuario" con el texto "prueba" y "Contraseña" con caracteres ocultos por puntos. Debajo de los campos, hay un mensaje de error en rojo que dice "Contraseña inválida". En la parte inferior del formulario, hay un botón azul que dice "Iniciar sesión" y un enlace de texto "Go to Back" en gris.

Nota. Validación y mensaje de error al ingresar un usuario o contraseña incorrecta. Figura elaborada por el Autor.

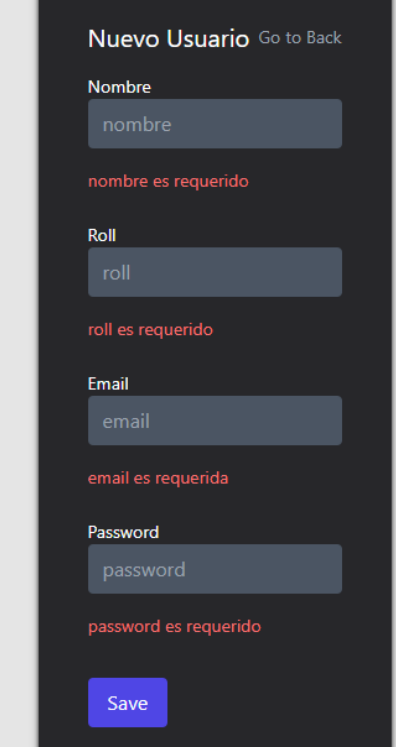
Figura 15. Validación del formulario de cirugía.

The image shows a mobile application interface for creating a new surgery. The form is titled "Nueva Cirugía" and has a "Go to Back" button in the top right corner. The form contains several input fields, each with a validation error message displayed below it in red text:

- Nombre:** The input field contains the placeholder text "nombre". Below it, the error message "nombre es requerido" is displayed.
- Descripción:** The input field contains the placeholder text "descripción". Below it, the error message "descripción es requerido" is displayed.
- Técnica:** The input field contains the placeholder text "técnica". Below it, the error message "técnica es requerida" is displayed.
- Variante:** The input field contains the placeholder text "variante". Below it, the error message "variante es requerido" is displayed.
- Imagen:** This field is a file selector. It contains a button labeled "Seleccionar archivo" and the text "Sin archivos seleccionados".
- Referencia:** The input field contains the placeholder text "referencia". Below it, the error message "referencia es requerida" is displayed.

Nota. Validación y mensaje de error al tratar de ingresar una nueva cirugía sin llenar los parámetros obligatorios. Figura elaborada por el Autor.

Figura 16. Validación del formulario de usuario.



The image shows a mobile application interface for creating a new user. The title is "Nuevo Usuario" with a "Go to Back" link. There are four input fields, each with a validation message below it:

- Nombre:** The input field contains the text "nombre". Below it, the message "nombre es requerido" is displayed in red.
- Roll:** The input field contains the text "roll". Below it, the message "roll es requerido" is displayed in red.
- Email:** The input field contains the text "email". Below it, the message "email es requerida" is displayed in red.
- Password:** The input field contains the text "password". Below it, the message "password es requerido" is displayed in red.

At the bottom of the form, there is a blue "Save" button.

Nota. Validación y mensaje de error al tratar de ingresar un nuevo usuario sin llenar los parámetros obligatorios. Figura elaborada por el Autor.

6.7.Control de versiones

Es de vital importancia contar con un repositorio confiable y seguro para almacenar y gestionar el código fuente de la aplicación. En este caso, se recomienda utilizar plataformas de control de versiones como GitHub. Un repositorio en GitHub ofrece numerosos beneficios, como la posibilidad de mantener un historial completo de cambios, facilitar la colaboración entre diferentes desarrolladores, realizar seguimiento de problemas y solicitudes de mejora, y garantizar una copia de respaldo del proyecto en caso de cualquier eventualidad.

En este sentido, me complace informarles que el proyecto se encuentra alojado en mi repositorio de GitHub. Pueden acceder a él a través del siguiente enlace: GitHub - https://github.com/SnNig1/surgeries_repository. Allí encontrarán el código fuente y podrán realizar un seguimiento detallado de las actualizaciones y contribuciones realizadas durante el desarrollo del proyecto. El uso de GitHub como plataforma de alojamiento y control de versiones es una práctica estándar en el desarrollo de software, y contribuye significativamente a la transparencia y trazabilidad del proyecto.

7. Recomendaciones

La aplicación entregada tiene como objetivo principal tener un repositorio web enfocado al área de la salud para brindar ayuda en la planeación de técnicas quirúrgicas a usar en los diferentes pacientes según el criterio de los médicos. Sin embargo, para la aplicación creada se espera tener otros módulos como lo son odontología, cirugía estética, ortopedia, entre otros. Con esto se busca que la aplicación tenga cabida en el área de la salud para brindar soporte principalmente a los estudiantes y docentes de estas áreas.

El control de permisos para los roles de usuario fue implementado de tal forma que solo existen administrador, con todos los privilegios de lectura y escritura y usuario con el privilegio de lectura. Sin embargo, la aplicación en su backend, está diseñada para que permita crear más “tipos de usuario” y además poder asignar, modificar o eliminar privilegios de acceso a cada tipo de usuario. Este último control de permisos descrito se recomienda implementar cuando la aplicación sea mucho más robusta.

Mejorar la seguridad de la aplicación: A medida que la aplicación crezca y se amplíe para incluir otros módulos, es importante tener en cuenta la seguridad de los datos y la protección de la privacidad de los usuarios. Se recomienda implementar medidas adicionales de seguridad, como la autenticación de dos factores, encriptación de datos sensibles, protección contra ataques de inyección de código y auditorías de seguridad

periódicas. Estas medidas ayudarán a garantizar la confidencialidad e integridad de la información y a fortalecer la confianza de los usuarios en la plataforma.

8. Conclusiones

El desarrollo de esta aplicación permitió el aprendizaje de un conjunto de tecnologías que se han vuelto muy populares en el desarrollo de aplicaciones web modernas y escalables a través del stack MERN. Esto incluye aprender MongoDB para almacenar grandes cantidades de datos de manera eficiente y flexible, Express y Node.js para crear un servidor web robusto y escalable, y React para crear interfaces de usuario interactivas y dinámicas.

La aplicación creada proporciona una solución tecnológica para la planeación de técnicas quirúrgicas en el área de la salud, que puede ser de gran utilidad tanto para estudiantes como para docentes en el campo médico. Además, al contar con un repositorio web, se puede acceder a la información de manera más eficiente y colaborativa.

La aplicación se diseñó para ser escalable y flexible, lo que facilita la incorporación de nuevos módulos en el futuro, como odontología, cirugía estética, ortopedia y otros. Esto permitirá que la aplicación tenga cabida en el área de la salud para brindar soporte a una variedad de especialidades.

La aplicación tiene como objetivo brindar soporte a los estudiantes y docentes en el área de la salud, lo que puede contribuir a mejorar la calidad de la enseñanza y el

aprendizaje en este campo. Además, la aplicación también puede ser útil para los profesionales de la salud en el proceso de planeación de técnicas quirúrgicas para sus pacientes, lo que puede tener un impacto positivo en su atención médica.

Bibliografía

- Abelló, D. (2021). Análisis comparativo de 2 técnicas quirúrgicas en el tratamiento de la hernia incisional subxifoidea. Estudio observacional. *Cirugía Española*, (99), de <https://www.elsevier.es/es-revista-cirugia-espanola-36-articulo-analisis-comparativo-2-tecnicas-quirurgicas-S0009739X20302827>
- Amazon. (2023). ¿Qué es NoSQL?. aws.amazon.com Recuperado de <https://aws.amazon.com/es/nosql/>
- Blog. (2020). Illustrative diagram of the MERN stack [Image]. Recuperado de <https://www.bocasay.com/how-does-the-mern-stack-work/>
- Castro Romero. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. Universidad Pedagógica y Tecnológica de Colombia Recuperado de <https://www.redalyc.org/pdf/4139/413940772003.pdf>
- Díaz Alfonso, Leonor R, Machado Héctor, Elvira, García Álvarez, Hernán, & Alegre Núñez, Juan. (2000). Estudio comparativo de 2 técnicas quirúrgicas para la cirugía del pterigión primario. *Revista Cubana de Oftalmología*, 13(2) Recuperado en 07 de febrero de 2023, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21762000000200004&lng=es&tlng=es
- ELSEVIER. (2023). Learning medicine made simple. osmosis.org Recuperado de <https://www.osmosis.org/>
- Genbeta. (2017). clasificación de las bases de datos según el teorema CAP [Image]. Recuperado de <https://www.genbeta.com/desarrollo/nosql-clasificacion-de-las-bases-de-datos-segun-el-teorema-cap>
- H. Gracia, O. Yanes. (2012). Bases de datos NoSQL. *TELEM@TICA*, (11), pp. 21 – 33.
- Joyanes, L. (2019). *Inteligencia De Negocios Y Analítica De Datos*. México: Alfaomega. Recuperado de <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/74/74>
- La resolución 5261 del Ministerio de Salud, de 05 de Agosto de 1994 https://www.minsalud.gov.co/Normatividad_Nuevo/RESOLUCI%C3%93N%205261%20DE%201994.pdf
- Lecturio GmbH. (2023). Lecturio . Internet Recuperado de <https://www.lecturio.com>

Mehra Monika, Manish Kumar. (2021). MERN Stack Web Development. Arya Institute of Engineering & Technology, Jaipur, Rajasthan, India. Recuperado el 05 de marzo del 2023.

Metronic. (2022). Touch Surgery™. iOS and Android stores Recuperado de <https://www.touchsurgery.com/download-the-app>

Novick, A. (2010). *Técnicas quirúrgicas en urología*. : Journal, de <https://medilibro.com/product/tecnicas-quirurgicas-en-urologia/>

Pacherres Muñoz, Liliana Raquel. (2018). PAGINAS WEB. FACULTAD DE CIENCIAS Recuperado de <https://repositorio.une.edu.pe/bitstream/handle/20.500.14039/5006/P%C3%A1ginas%20web.pdf?sequence=1&isAllowed=y#page=26&zoom=100,109,878>

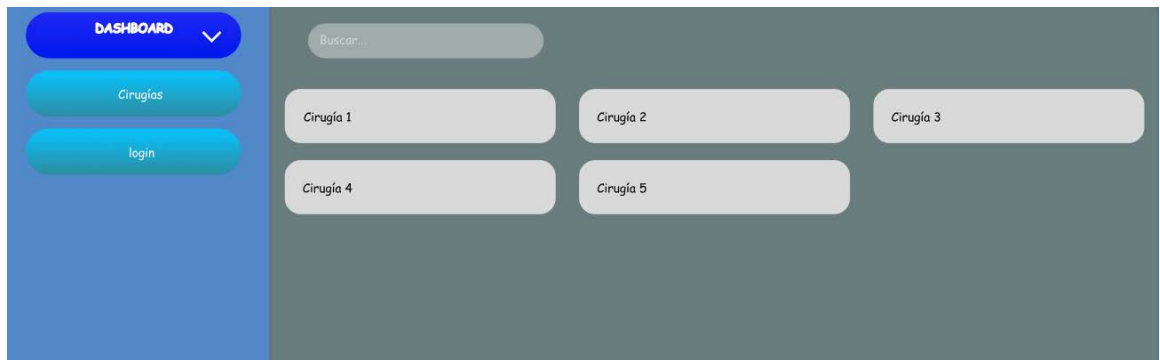
Sanchit Aggarwal, Jyoti Verma. (2018). Comparative analysis of MEAN stack and MERN stack. Department of Information Technology, Northern India Engineering College, New Delhi. Recuperado el 05 de marzo del 2023

Toledo-Valdez, Cielo, Rivera-Barreno, Ramón, Talamantes-Lima, Ilse, Bustos-Varela, Jocelyn, García-Herrera, Ricardo, & Rodríguez-Alarcón, Carlos. (2021). Revisión sistemática de las diferentes técnicas quirúrgicas de contracepción en gatas. *Abanico veterinario*, 11, e203. Epub 08 de noviembre de 2021. Recuperado de <https://doi.org/10.21929/abavet2021.27>

Visible Body. (2022). Atlas de anatomía humana. Android Store Recuperado de https://play.google.com/store/apps/details?id=com.visiblebody.atlas&hl=es_CO&gl=US&pli=1

Anexo A

Figura 17. Vista del usuario.



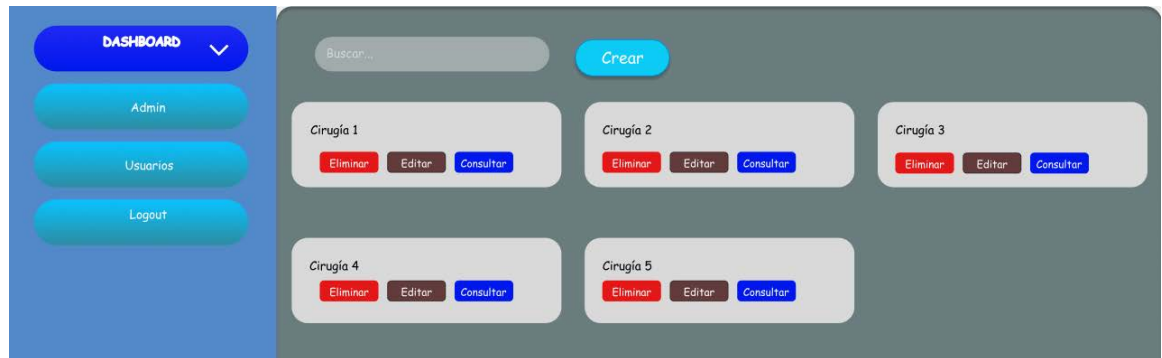
Nota. La vista del usuario contará con una sidebar, una barra de búsqueda y el listado de cirugías. Figura elaborada por el Autor.

Figura 18. Vista del login.



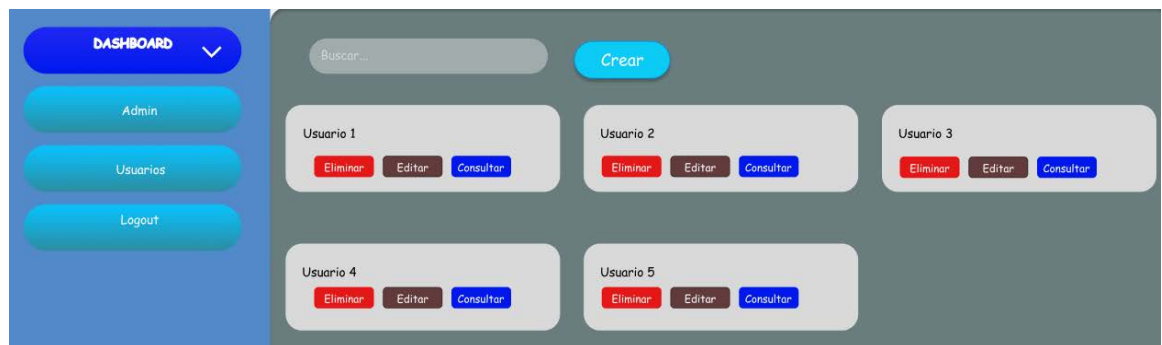
Nota. La vista del login contará con sus respectivos campos para la validación del ingreso y un botón para volver. Figura elaborada por el Autor.

Figura 19. Vista del Administrador en el apartado Admin.



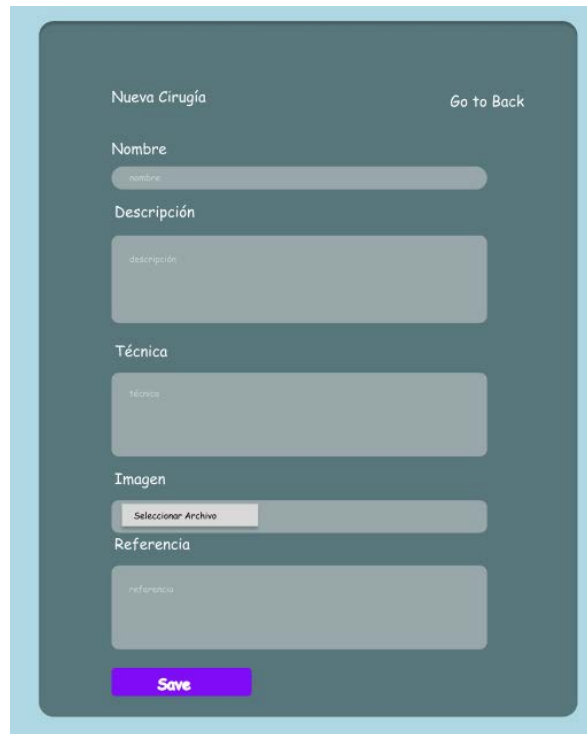
Nota. La vista del Administrador del apartado Admin contará con sus respectivos campos para la realización del CRUD de las cirugías. Figura elaborada por el Autor.

Figura 20. Vista del Administrador en el apartado Usuarios.



Nota. La vista del Administrador del apartado Usuarios contará con sus respectivos campos para la realización del CRUD de los usuarios. Figura elaborada por el Autor.

Figura 21. Vista del Administrador en el formulario Cirugías.



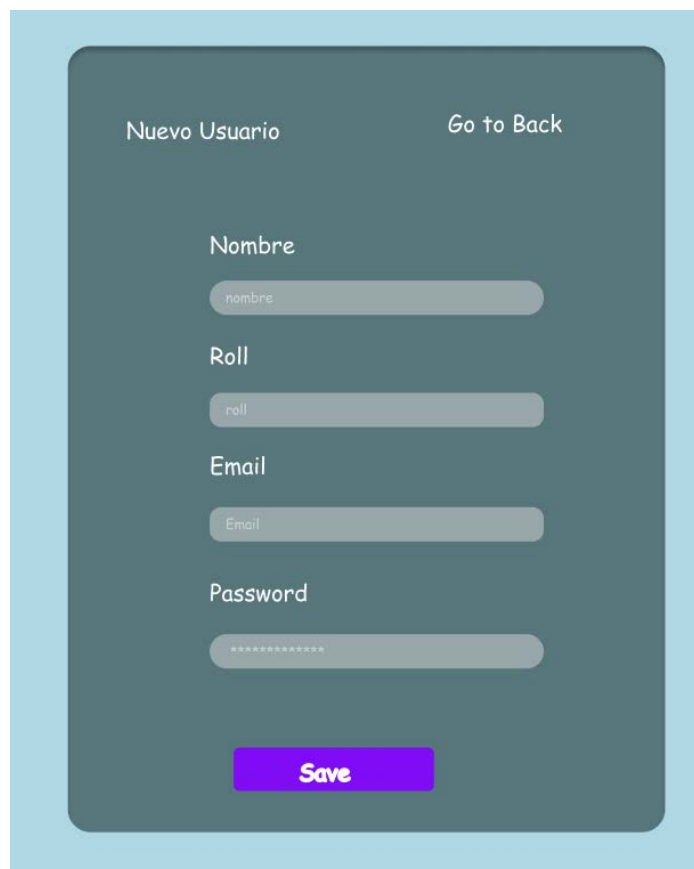
The image shows a mobile application interface for adding a new surgery. The form is titled 'Nueva Cirugía' and has a 'Go to Back' link in the top right corner. The form contains the following fields:

- Nombre:** A text input field with a placeholder 'nombre'.
- Descripción:** A larger text input field with a placeholder 'descripción'.
- Técnica:** A text input field with a placeholder 'técnica'.
- Imagen:** A file selection field with a button labeled 'Seleccionar Archivo'.
- Referencia:** A text input field with a placeholder 'referencia'.

At the bottom of the form is a purple 'Save' button.

Nota. La vista del Administrador del formulario cirugías, contará con sus respectivos campos para la realización de la inserción de nuevas cirugías y la actualización de las mismas. Figura elaborada por el Autor.

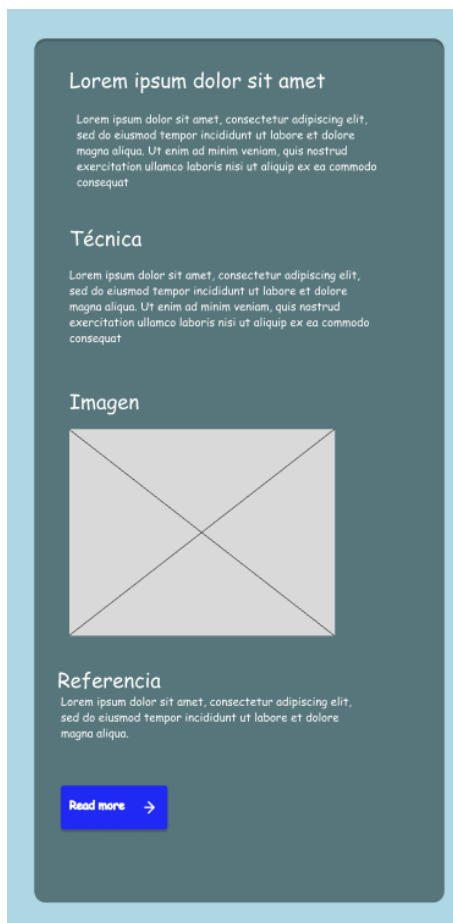
Figura 22. Vista del Administrador en el formulario Usuarios.



The image shows a mobile application interface for creating a new user. The form is titled "Nuevo Usuario" and has a "Go to Back" button in the top right corner. It contains four input fields: "Nombre" (with placeholder text "nombre"), "Roll" (with placeholder text "roll"), "Email" (with placeholder text "Email"), and "Password" (with placeholder text "*****"). A purple "Save" button is located at the bottom of the form.

Nota. La vista del Administrador del formulario usuarios, contará con sus respectivos campos para la realización de la inserción de nuevos usuarios y la actualización de los mismos. Figura elaborada por el Autor.

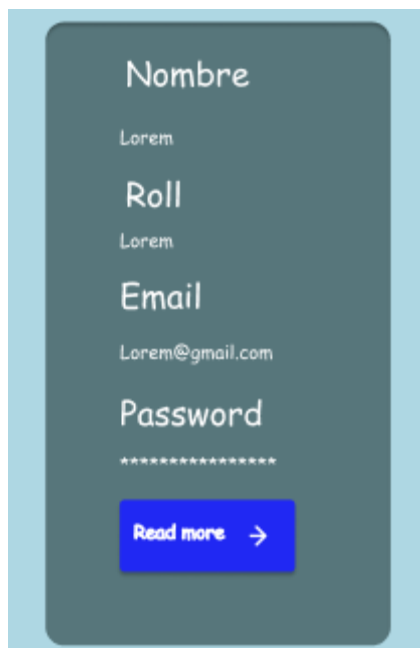
Figura 23. Vista del Administrador al consultar una cirugía.



Nota. La consulta de una cirugía en específico, contará con la anterior estructura.

Figura elaborada por el Autor.

Figura 24. Vista del Administrador al consultar un usuario.

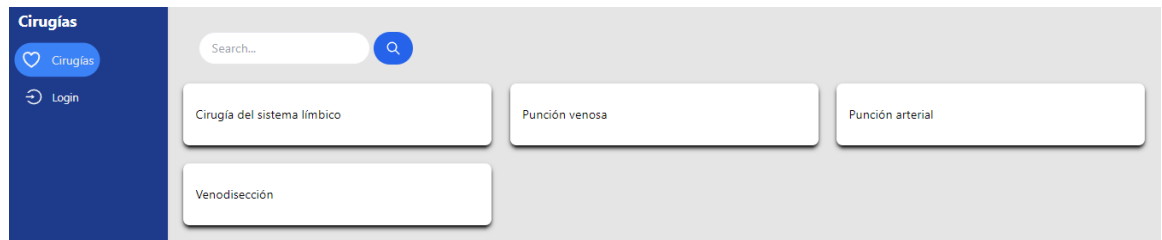


Nota. La consulta de un usuario en específico, contará con la anterior estructura.

Figura elaborada por el Autor.

Anexo B

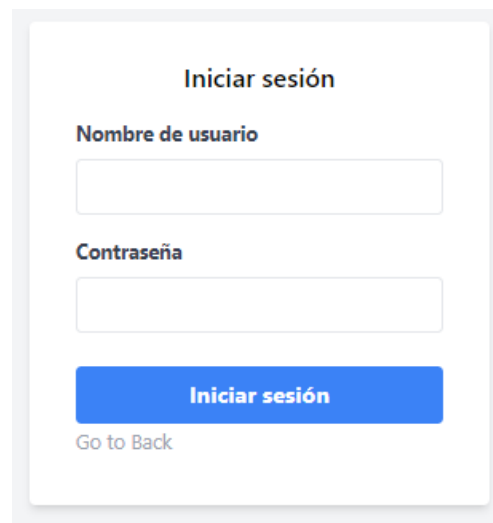
Figura 25. Vista del usuario con React y Tailwind.



Nota. La consulta de una cirugía en específico, contará con la anterior estructura.

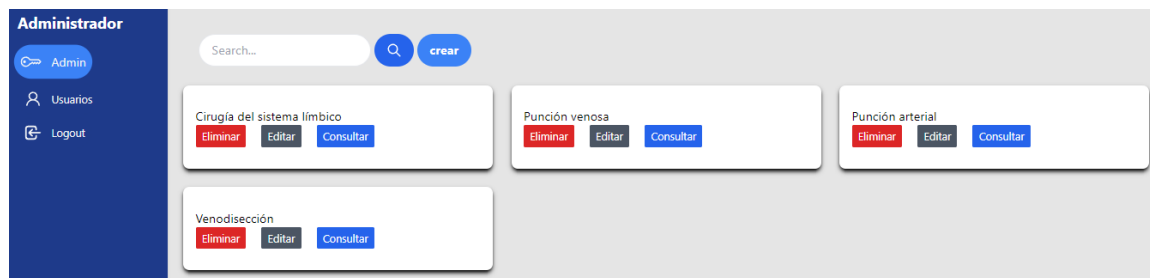
Figura elaborada por el Autor.

Figura 26. Vista del login con React y Tailwind.



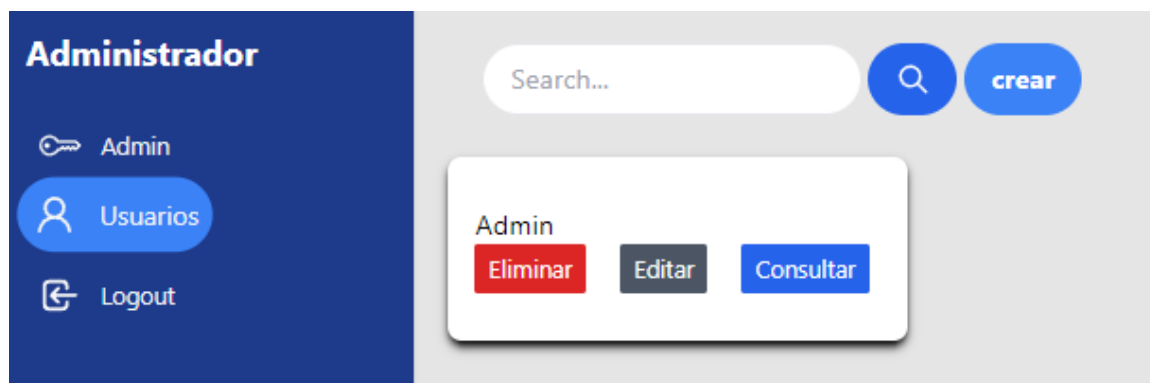
Nota. La vista del login contará con sus respectivos campos para la validación del ingreso y un botón para volver. Figura elaborada por el Autor.

Figura 27. Vista del Administrador en el apartado Admin con React y Tailwind.



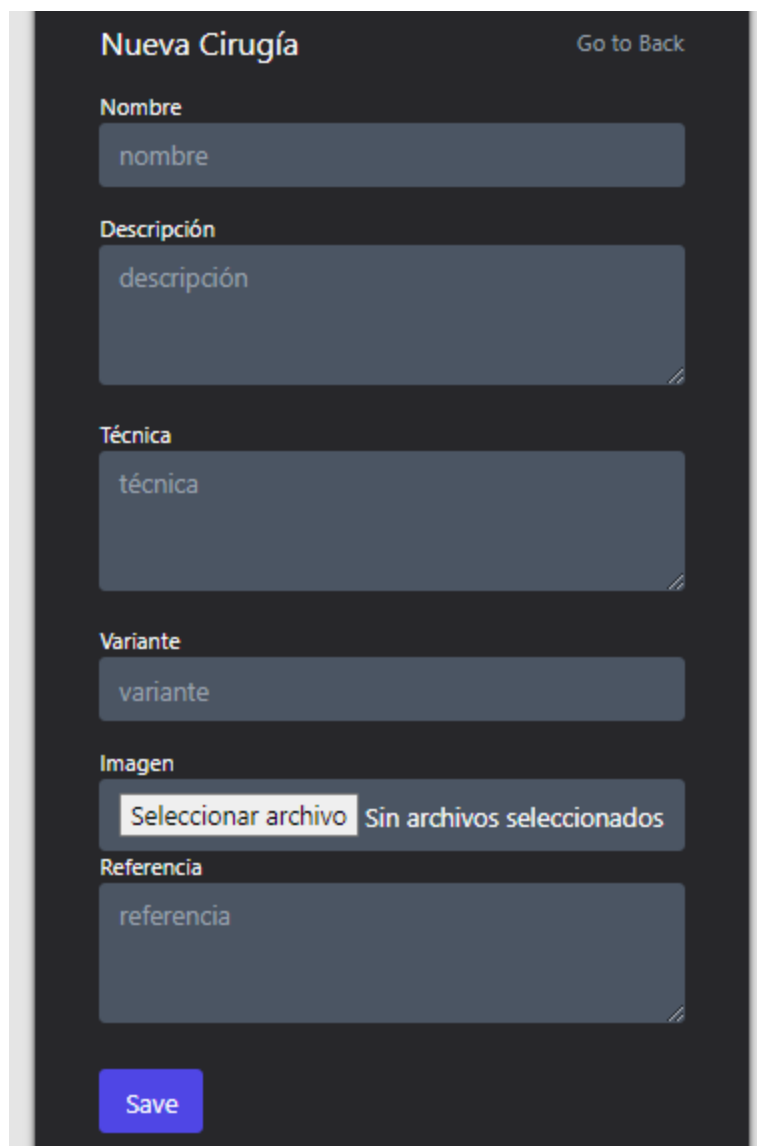
Nota. La vista del Administrador del apartado Admin contará con sus respectivos campos para la realización del CRUD de las cirugías. Figura elaborada por el Autor.

Figura 28. Vista del Administrador en el apartado Usuarios con React y Tailwind.



Nota. La vista del Administrador del apartado Usuarios contará con sus respectivos campos para la realización del CRUD de los usuarios. Figura elaborada por el Autor.

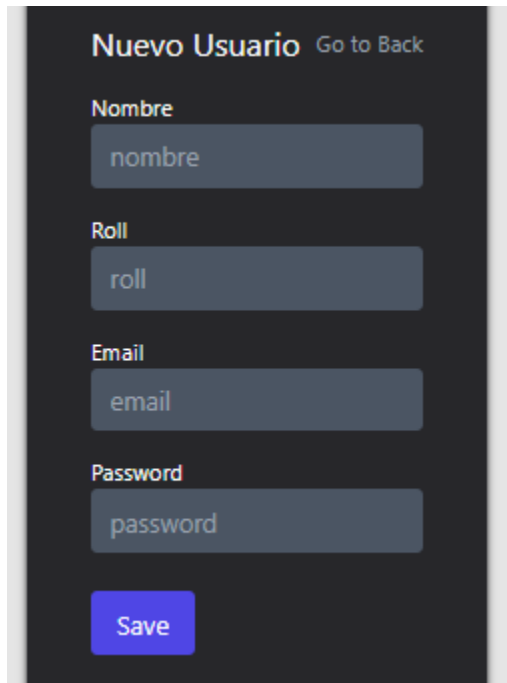
Figura 29. Vista del Administrador en el formulario Cirugías con React y Tailwind.



The image shows a mobile application interface for creating a new surgery. The title is "Nueva Cirugía" and there is a "Go to Back" link in the top right corner. The form consists of several input fields: "Nombre" (text), "Descripción" (text area), "Técnica" (text area), "Variante" (text), "Imagen" (file selection with a button "Seleccionar archivo" and status "Sin archivos seleccionados"), and "Referencia" (text area). A blue "Save" button is located at the bottom left of the form.

Nota. La vista del Administrador del formulario cirugías, contará con sus respectivos campos para la realización de la inserción de nuevas cirugías y la actualización de las mismas. Figura elaborada por el Autor.

Figura 30. Vista del Administrador en el formulario Usuarios con React y Tailwind.



The image shows a dark-themed user creation form titled "Nuevo Usuario" with a "Go to Back" link. It contains four input fields: "Nombre" (placeholder: nombre), "Roll" (placeholder: roll), "Email" (placeholder: email), and "Password" (placeholder: password). A blue "Save" button is located at the bottom.

Nota. La vista del Administrador del formulario usuarios, contará con sus respectivos campos para la realización de la inserción de nuevos usuarios y la actualización de los mismos. Figura elaborada por el Autor.

Figura 31. Vista del Administrador al consultar una cirugía con React y Tailwind.

Punción arterial

La punción arterial se utiliza en el análisis de gases arteriales (gasometría arterial), para vigilancia de la presión arterial, o bien, para administrar medicamentos o medios de contraste. Una punción arterial obtiene mejores resultados cuando se realiza en las arterias radial o femoral.

Técnica

a) Asegúrese de mantener condiciones de asepsia y antisepsia. b) Palpe el pulso arterial (localización de la arteria) (figuras 6-6a y 6-6b). c) Inserte la aguja sobre la arteria, atravesando la piel con el bisel hacia arriba justo por debajo del pulso palpado con anterioridad. La vena femoral se localiza en sentido medial, la arteria femoral se encuentra lateral a la vena en el triángulo femoral, justo por debajo del ligamento inguinal (figura 6-7). d) Cuando observe sangre dentro de la aguja extraiga un poco la jeringa (figura 6-8). En caso de que se desee monitorear, deslice un catéter de teflón dentro de la arteria. e) Fije el catéter con cinta adhesiva o sutura (figura 6.9). f) Al retirar la aguja comprima el sitio de la punción por lo menos durante tres minutos (figura 6-10). NB: Si se coloca un catéter para monitoreo por más de 6 horas, es necesario heparinizarlo y limpiarlo con solución.

Variante

normal








Referencia

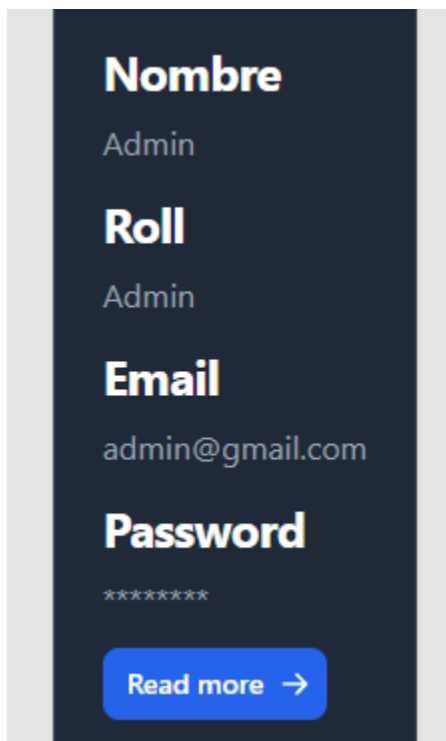
Tomado Procedimientos quirúrgicos básicos Pág. 141

[Read more →](#)

Nota. La consulta de una cirugía en específico, contará con la anterior estructura.

Figura elaborada por el Autor.

Figura 32. Vista del Administrador al consultar un usuario con React y Tailwind.



Nota. La consulta de un usuario en específico, contará con la anterior estructura.

Figura elaborada por el Autor.