



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 1

Neiva, 5 de junio de 2023

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El suscrito:

Cristian Samuel Carrera Guzmán, con C.C. No. 1075313098, Autor de la tesis y/o trabajo de grado titulado “Diseño e implementación de una plataforma web para el registro y publicación de resultados de la prueba de la hormona antidiurética (ADH).” presentado y aprobado en el año 2023 como requisito para optar al título de ingeniero electrónico; Autorizo al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.
- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “Los derechos morales sobre el trabajo son propiedad de los autores”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

Cristian Samuel Carrera Guzmán:

Firma:

Vigilada Mineducación



TÍTULO COMPLETO DEL TRABAJO: DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA WEB PARA EL REGISTRO Y PUBLICACIÓN DE RESULTADOS DE LA PRUEBA ANTIDIURÉTICA (ADH).

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Carrera Guzmán	Cristian Samuel

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Cortés-Cabezas	Albeiro

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
----------------------------	--------------------------

PARA OPTAR AL TÍTULO DE: Ingeniero electrónico

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Ingeniería electrónica

CIUDAD: Neiva, Huila

AÑO DE PRESENTACIÓN: 2023

NÚMERO DE PÁGINAS: 43



TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas X Fotografías___ Grabaciones en discos___ Ilustraciones en general X Grabados___ Láminas___
Litografías___ Mapas___ Música impresa___ Planos___ Retratos___ Sin ilustraciones___ Tablas o Cuadros X

SOFTWARE requerido y/o especializado para la lectura del documento:

MATERIAL ANEXO:

PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o Meritoria):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>
1. hormona antidiurética (ADH)	Antidiuretic hormone (ADH)
2. MERN	MERN
3. MongoDB	MongoDB
4. Express.js	Express.js

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)



Actualmente el departamento del Huila no posee un sistema para la adquisición y publicación de datos estadísticos médicos eficiente y abierta al público, por ende, se planteó diseñar e implementar un aplicativo web para el registro y publicación de resultados para el caso específico de la prueba de la hormona antidiurética (ADH). El aplicativo se desarrolló bajo el framework MERN y algunas librerías como Bootstrap, Handlebars, Passport.js, Bcrypt.js y Connect-flash. Se obtuvo un aplicativo web altamente escalable y flexible, con dos colecciones en la base de datos no relacional (MongoDB); se usaron las rutas del servidor Express.js y se muestran tres agrupaciones estadísticas al público. Este proyecto abre la posibilidad de llevar un seguimiento más cercano y detallado de diferentes aspectos de salud de los huilenses; gracias a esto, los profesionales de la salud podrán crear estrategias de prevención de enfermedades, al ser altamente escalable es factible agregar más exámenes y tipos de agrupación/evaluación de datos.

ABSTRACT: (Máximo 250 palabras)



Currently the department of Huila does not have a system for the acquisition and publication of medical statistical data in an efficient manner and open to the public, therefore, it was proposed to design and implement a web application for the registration and publication of results for the specific case of the antidiuretic hormone (ADH) test. The application was implemented under the MERN framework and some libraries such as Bootstrap, Handlebars, Passport.js, Bcrypt.js and connect-flash. A highly scalable and flexible web application was obtained, with two collections in the non-relational database (MongoDB); Routes from Express.js were used and three statistical groupings were displayed to the public. This project opens the possibility of carrying out a closer and more detailed follow-up of different aspects of the health of huilenses; Thanks to this, health professionals will be able to create disease prevention strategies, being highly scalable it is feasible to add more tests and types of data grouping/evaluation.

APROBACION DE LA TESIS

Nombre Jurado: Johan Julian Molina Mosquera

Firma:

Nombre Jurado: Martin Diomedes Bravo Obando

Firma:

**DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA WEB PARA EL REGISTRO Y
PUBLICACIÓN DE RESULTADOS DE LA PRUEBA DE LA HORMONA
ANTIDIURÉTICA (ADH).**

CRISTIAN SAMUEL CARRERA GUZMÁN

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA, COLOMBIA
2022**

**DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA WEB PARA EL
REGISTRO Y PUBLICACIÓN DE RESULTADOS DE LA PRUEBA DE LA
HORMONA ANTIDIURÉTICA (ADH).**

**CRISTIAN SAMUEL CARRERA GUZMÁN
Cod. 20161147202**

**Trabajo de grado para optar
al título de ingeniero electrónico**

**Director:
MSc. PhD. Albeiro Cortés Cabezas**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA, COLOMBIA
2022**

Nota de aceptación

Firma del director

Firma del Jurado

Firma del Jurado

Neiva, abril de 2023

RESUMEN

Actualmente el departamento del Huila no posee un sistema para la adquisición y publicación de datos estadísticos médicos eficiente y abierta al público, por ende, se planteó diseñar e implementar un aplicativo web para el registro y publicación de resultados para el caso específico de la prueba de la hormona antidiurética (ADH). El aplicativo se desarrolló bajo el framework MERN y algunas librerías como Bootstrap, Handlebars, Passport.js, Bcrypt.js y Connect-flash. Se obtuvo un aplicativo web altamente escalable y flexible, con dos colecciones en la base de datos no relacional (MongoDB); se usaron las rutas del servidor Express.js y se muestran tres agrupaciones estadísticas al público. Este proyecto abre la posibilidad de llevar un seguimiento más cercano y detallado de diferentes aspectos de salud de los huilenses; gracias a esto, los profesionales de la salud podrán crear estrategias de prevención de enfermedades, al ser altamente escalable es factible agregar más exámenes y tipos de agrupación/evaluación de datos.

Palabras clave: hormona antidiurética (ADH), MERN, MongoDB, Express.js.

ABSTRACT

Currently the department of Huila does not have a system for the acquisition and publication of medical statistical data in an efficient manner and open to the public, therefore, it was proposed to design and implement a web application for the registration and publication of results for the specific case of the antidiuretic hormone (ADH) test. The application was implemented under the MERN framework and some libraries such as Bootstrap, Handlebars, Passport.js, Bcrypt.js and connect-flash. A highly scalable and flexible web application was obtained, with two collections in the non-relational database (MongoDB); Routes from Express.js were used and three statistical groupings were displayed to the public. This project opens the possibility of carrying out a closer and more detailed follow-up of different aspects of the health of huilenses; Thanks to this, health professionals will be able to create disease prevention strategies, being highly scalable it is feasible to add more tests and types of data grouping/evaluation.

Key words: Antidiuretic hormone (ADH), MERN, MongoDB, Express.js.

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	11
2. OBJETIVOS.....	13
2.1. Objetivo general.....	13
2.2. Objetivos especificos	13
3. ANTECEDENTES.....	14
4. FUNDAMENTOS BÁSICOS	16
4.1. Hormona antidiurética (ADH).....	16
4.2. Secreción inadecuada de la hormona antidiurética.....	16
4.3. Prueba de la hormona antidiurética	16
5. METODOLOGÍA.....	18
5.1. Estructura del proyecto.....	18
5.2. Modelado de la base de datos	20
5.3. Cifrado de contraseña	21
5.4. Motor de plantillas	22
5.5. Direccionamiento.....	22
5.5.1. Rutas para el primer uso de la plataforma	23
5.5.2. Enrutamiento de datos estadísticos	24
5.5.3. Funciones y permisos para cada rol en la aplicación web	27
5.5.4. Rutas que no necesitan autenticación de usuario.....	28
5.5.5. Funciones de administrador.....	28
5.5.6. Rutas del médico	29
5.5.7. Rutas del empleado	32
5.5.8. Rutas del paciente	35
6. RESULTADOS	36
7. CONCLUSIONES	40
8. TRABAJO FUTURO	41

REFERENCIAS 42

TABLA DE FIGURAS

Figura 1. Importación de paquetes y dependencias	18
Figura 2. Diagrama de flujo general de la plataforma	19
Figura 3. Diagrama de caso de uso de la plataforma.....	20
Figura 4. Inicialización de la plataforma en el archivo index.js usando Express js	20
Figura 5. Aplicación de Bcrypt para el cifrado de contraseña en la colección usuarios ..	22
Figura 6. Declaración de los archivos de enrutamiento en el archivo index.js	22
Figura 7. Diagrama de flujo de la creación del primer usuario	23
Figura 8. Formulario para la creación del primer usuario.....	24
Figura 9. Diagrama de flujo del despliegue de los gráficos estadísticos	25
Figura 10. Acceso a la plataforma desde el banner	26
Figura 11. Acceso a la plataforma desde la barra lateral	27
Figura 12. Funciones disponibles para cada rol.....	27
Figura 13. Diagrama de flujo para la generación de ordenes médicas	30
Figura 14. Formulario de búsqueda de usuario por cédula.....	31
Figura 15. Botón de generación de orden médica	31
Figura 16. Formulario de generación	31
Figura 17. Visualización de orden generada.....	32
Figura 18. Diagrama de flujo para la adición de resultados	33
Figura 19. Despliegue del botón "Acción"	33
Figura 20. Formulario para la adición de resultados	34
Figura 21. Formulario para la cancelación de resultados.....	34
Figura 22. Listado de exámenes cancelados.....	34
Figura 23. Visualización de exámenes realizados por paciente.....	35
Figura 24. Visualización de examen individual	35
Figura 25. Grafica en la plataforma web con los resultados de prueba agrupados por nivel.....	37
Figura 26. Visualización de la agrupación del total de exámenes y resultados fuera del	

rango común por cada municipio	37
Figura 27. diagrama circular de exámenes completados agrupados por rango de edad	38
Figura 28. Dashboard con datos de la plataforma	39

LISTA DE TABLAS

Tabla 1. Dependencias y paquetes utilizados en el aplicativo web.....	15
Tabla 2. Propiedades y descripción del esquema "ExamSchema"	16
Tabla 3. Rutas para la creación del primer administrador.....	18
Tabla 4. Posibles valores para la propiedad "group"	18
Tabla 5. Rutas para la visualización de datos estadísticos.....	19
Tabla 6. Handlers para las rutas estadísticas	19
Tabla 7. Helpers usados para la validación de usuario.....	20
Tabla 8. Rutas al que tienen acceso todos los roles	20
Tabla 9. Rutas para las funciones del administrador	21
Tabla 10. Descripción de los controladores de las funciones del administrador	21
Tabla 11. Rutas para las funciones del médico	22
Tabla 12. Descripción de los controladores de las funciones del médico	22
Tabla 13. Rutas para las funciones del empleado	23
Tabla 14. Descripción de los controladores de las funciones del empleado	23
Tabla 15. Distribución de roles ficticios para la realización de pruebas	24

1. INTRODUCCIÓN

La hormona antidiurética (ADH), o arginina vasopresina, es una hormona que se produce en el hipotálamo, se almacena y libera a través de la neurohipófisis presente en la mayoría de los mamíferos, incluyendo a los humanos. Su función es contraer los vasos sanguíneos y ayudar a los riñones en el control de sal y agua para la regulación de la presión arterial y orina producida (Instituto Nacional del Cáncer, 2022).

Determinadas afecciones pueden influir sobre la cantidad de ADH que produce el cuerpo, como ciertos tipos de tumores, niveles bajos de sodio en los fluidos corporales, diabetes insípida, entre otras. Los niveles normales de ADH para una persona adulta varían de 1 a 5 pg/ml. Los resultados que están por encima del rango normal pueden significar que el paciente presenta tumores en el pulmón, tumores en el sistema nervioso central, un desequilibrio de líquido postquirúrgico o, incluso, porfiria. Los niveles bajos de ADH pueden significar que el paciente presenta diabetes insípida, daño en la glándula pituitaria o polidipsia primaria que se manifiesta en el cuerpo bajo sed excesiva debido a problemas en el hipotálamo (UC San Diego Health, 2022).

Una forma común de monitoreo de estas afecciones sobre la salud de las personas es a través de estadísticas. Investigadores y organizaciones de gobierno (tanto privadas, como sin fines de lucro) recopilan estadísticas de salud y las utilizan para aprender sobre salud pública y atención médica —se entiende por estadísticas de salud a las cifras que resumen la información relacionada con la salud—. Algunos ejemplos de los tipos de estadísticas pueden ser: cuántas personas en una determinada zona presentan una enfermedad, cuántas personas contrajeron una enfermedad en un determinado período de tiempo y cuántas personas de cierto grupo tienen una enfermedad. Los grupos pueden basarse en ubicación, raza, grupo étnico, sexo, edad, profesión, nivel de ingresos o nivel de educación (MedlinePlus, 2022).

Por lo tanto, resulta importante contar con una herramienta de visualización de estas estadísticas para hacer seguimiento y tomar oportunas decisiones sobre la salud de una población. En el caso particular de las enfermedades generadas por un desnivel de la hormona ADH, el departamento del Huila no cuenta con una herramienta donde se muestren las estadísticas relacionadas con los resultados de las pruebas de ADH, por consiguiente, complica el seguimiento a la salud de los huilenses.

Es por esto que se planteó diseñar e implementar un aplicativo web en el cual se suministren estadísticas del examen de ADH y su correspondiente resultado en el departamento del Huila. En el presente documento se explica el desarrollo del aplicativo web utilizando el marco de trabajo MERN (MongoDB, Express.js, React y Node.js). Como resultado se muestran tres agrupaciones estadísticas abiertas al público de manera gráfica: i) la agrupación de resultados por nivel de densidad de

ADH (bajo, normal y alto), ii) el total de exámenes realizados junto con la cantidad de resultados anormales (aquellos que presentan baja o alta densidad de ADH) en cada municipio del Huila y iii) el total de exámenes realizados divididos por rango de edad.

2. OBJETIVOS

2.1. Objetivo general

Implementar un aplicativo web que permita al personal médico y usuarios del departamento del Huila ordenar exámenes de ADH, registrar su resultado, cancelar órdenes médicas y llevar agrupaciones estadísticas abiertas al público.

2.2. Objetivos específicos

- Estructurar la aplicación web teniendo en cuenta cuáles son los servicios y funciones que se quieren incluir.
- Modelar las colecciones en la base de datos no relacional MongoDB.
- Diseñar las vistas de la aplicación web con el fin de que la experiencia de usuario sea agradable, permitiendo una navegación fácil e intuitiva.
- Realizar el tratamiento de datos de los exámenes, resultados y usuarios para el análisis y agrupación estadística.
- Utilizar el framework MERN para la implementación de la aplicación web.

3. ANTECEDENTES

En esta sección se mencionan investigaciones relacionadas al desarrollo e implementación de sistemas de gestión de historial clínico.

Maradei, Espinel, & Diaz (2008) diseñaron una herramienta llamada “Web ANTROPOS 2.0”, la cual almacena y genera información estadística antropométrica de una población en específico. La información puede ser consultada por cualquier usuario interesado en el diseño de productos o espacios de trabajos para el desarrollo y construcción de mejores condiciones de vida. ANTROPOS 2.0 fue desarrollado según la ingeniería de usabilidad, se desarrolló bajo la arquitectura cliente/ servidor utilizando PHP y el administrador de bases de datos MySQL. El lenguaje PHP se ha visto reemplazado por JavaScript (JS) en los últimos años, ya que el aporte de la comunidad de desarrolladores ha sido mayor para JS, un claro ejemplo es el surgimiento de herramientas como: Node, React, el administrador de paquetes NPM y librerías de código abierto. Sin embargo, la arquitectura cliente/ servidor aún es vigente. Por lo tanto, en el presente proyecto se utilizó el marco de trabajo MERN, debido a que tiene como base al servidor Express.js, que trabaja bajo la arquitectura cliente/ servidor y su lenguaje de desarrollo es JavaScript.

Carrión (2015) presentó la tesis titulada “Desarrollo de una aplicación web basada en el modelo vista controlador para la gestión de las historias clínicas de los pacientes en el centro de salud de San Jerónimo”, su aplicativo web, denominado SGHC, es una herramienta para la gestión automatizada de todo lo referente con el expediente médico de un paciente en el centro de salud de San Jerónimo. El autor afirma que, gracias a la implementación de este proyecto, se logró reducir la búsqueda de un expediente de 9 minutos a 15 segundos y una reducción de registro de una historia clínica de 15 minutos a 8 minutos, además de reducir a cero la pérdida de las historias clínicas. Lo anterior confirma un aumento en la efectividad y productividad dentro de un establecimiento si se desarrollan soluciones digitales para suplir los procesos mecánicos.

Chiroles, Pacheco, Reyes, & Sisto (2017) implementaron una aplicación web para el módulo de servicio quirúrgico que gestione la información contenida en el anuncio e informe operatorio como parte del servicio quirúrgico en el Hospital Clínico Docente León Cuervo Rubio en la provincia Pinar del Río, Cuba. El aplicativo web fue desarrollado con el framework Angular 2 y cuenta con las siguientes funciones: gestionar anuncio operatorio, gestionar informe operatorio, emitir reporte de anuncios operatorios pendientes, emitir reporte de informes operatorios pendientes, emitir reporte de anuncios operatorios aprobados, emitir reporte de informes operatorios aprobados y emitir reporte de operaciones pendientes.

Calderón (2019) implementó una aplicación web con el fin de mejorar el

tratamiento y seguimiento de la población de pacientes que son autorizados para el traslado a hospitales de mayor complejidad. Con este proyecto el autor propone automatizar el registro y el proceso de autorización y evitar diligenciar dos veces los formularios de ingreso de las dos entidades donde se atiende al paciente. En este proyecto se usó la metodología RUP.

Mochãoa, y otros (2021) diseñaron una plataforma web denominada *IPOscore*, el propósito de esta aplicación es para el análisis y predicción de complicaciones quirúrgicas posoperatorias en el campo de la oncología. *IPOscore* podría llegar a ser considerado como un sistema de soporte para decisiones médicas, específicamente en el sector oncológico. Este sistema incluye una base de datos para almacenar/gestionar los datos clínicos recopilados en un formato estructurado, herramientas de análisis y visualización de datos, y modelos predictivos de aprendizaje automático para predecir los resultados postoperatorios en pacientes con cáncer.

4. FUNDAMENTOS BÁSICOS

4.1. Hormona antidiurética (ADH)

También es conocida como arginina vasopresina, o argipresina. Esta hormona se encarga de controlar la cantidad de agua que recuperan los riñones después de haber filtrado los desechos de la sangre, es decir, regula el equilibrio hídrico en el organismo. La arginina vasopresina es producida en el hipotálamo y se almacena en la hipófisis. Cuando el cuerpo detecta aumento de la osmolalidad de la sangre o disminuciones del volumen sanguíneo, se genera secreción de ADH por parte de la hipófisis. Los riñones responden a la acción de la ADH conservando el agua, lo que produce orina más concentrada. El agua retenida ayuda a la dilución de la sangre, lo que ocasiona disminución de la osmolalidad y aumento en la presión de la sangre. En el caso de que este mecanismo no sea suficiente para restaurar el balance hídrico del organismo, se pone en marcha el mecanismo de la sed, para que la persona beba más agua (Labtestsonline, 2023).

4.2. Secreción inadecuada de la hormona antidiurética

Se conoce como síndrome de secreción inadecuada de la hormona antidiurética (SSIHA) cuando el cuerpo produce densidades de ADH menores a 1pg/ml o mayores a 5pg/ml (MedlinePlus, 2022), algunas de las razones por las que se produce el SSIHA son:

- Consumo de alimentos como fármacos anticonvulsivos, antidepresivos, fármacos contra el cáncer, opiáceos (con menos frecuencia) y fármacos para el corazón, la diabetes y la presión arterial.
- Cirugía bajo anestesia general.
- Trastornos cerebrales provenientes de una lesión, infecciones o accidente cerebrovascular
- Tuberculosis, cáncer, infecciones crónicas y enfermedades pulmonares como la neumonía.

En el caso de disminución en la secreción de ADH, el organismo pierde gran cantidad de agua por los riñones, lo que significa que la orina emitida será muy diluida y la sangre estará concentrada. Consecuentemente, el paciente sentirá sed, aumento de la frecuencia de orina y, en el caso de no reponer la falta de agua bebiendo más cantidad, se producirá deshidratación e hipernatremia (aumento de la concentración de sodio en sangre).

Si se produce un aumento en la cantidad de ADH, el cuerpo retendrá mucha agua, aumentará el volumen sanguíneo y podrían aparecer náuseas, dolores de cabeza, desorientación, somnolencia e hiponatremia (Labtestsonline, 2023).

4.3. Prueba de la hormona antidiurética

La prueba de ADH se debe realizar cuando la concentración de sodio en la sangre está disminuida o se presenta una sed persistente, necesidad frecuente de orinar,

vomito, deshidratación, dolor de cabeza, náuseas y confusión. El examen se realiza a partir de una muestra de sangre venosa del brazo. se realiza por tres posibles razones:

- Para detectar, diagnosticar y determinar la causa de una deficiencia, resistencia o exceso de hormona antidiurética (ADH).
- Evaluar una baja concentración de sodio en sangre (hiponatremia).
- Distinguir los dos tipos de diabetes insípida.

Para la toma de muestra de ADH no se necesita ninguna preparación especial. Sin embargo, es importante consultar al médico en el caso de que la prueba sea parte de una prueba de restricción hídrica o de sobrecarga hídrica (Labtestsonline, 2023).

5. METODOLOGÍA

5.1. Estructura del proyecto

Para la implementación de este aplicativo se usó el framework MERN, ya que permite escalabilidad al ser sumamente flexible. Para la creación del marco y entorno de trabajo fue necesaria la instalación de módulos, paquetes y dependencias. En la Tabla 1 se muestran las dependencias y paquetes usados para la creación del entorno en el que se trabajó y en la Figura 1 se muestra cómo se importan en el archivo index.js.

Tabla 1. Dependencias y paquetes utilizados en el aplicativo web

Módulo	Descripción
Express	Permite la integración con motores de renderización de “vistas” para generar respuestas mediante la introducción de datos en plantillas y establece el puerto de conexión.
connect-flash	Paquete que permite la comunicación entre vistas y muestra mensajes o alertas.
bcrypt.js	Es una función criptográfica de hash, que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija, independientemente de la longitud de entrada.
Handlebars	Es un lenguaje de plantillas simple. Utiliza una plantilla y un objeto de entrada para generar HTML u otros formatos de texto.
Express-session	Almacena los datos de sesión en el servidor; sólo guarda el ID de sesión en la propia cookie, no los datos de sesión.
Method-override	Crea una nueva función de middleware para anular la propiedad req.method con un nuevo valor, gracias a este módulo es posible usar el método PUT para editar o eliminar documentos de la base de datos.
Mongoose	Biblioteca de programación orientada a objetos de JavaScript que crea una conexión entre MongoDB y el entorno de tiempo de ejecución de JavaScript de Node.js.
Passport	Middleware de autenticación para Node.js flexible y modular, se puede colocar discretamente en cualquier aplicación web basada en Express.
Nodemon	Herramienta que ayuda a desarrollar aplicaciones basadas en Node.js al reiniciar automáticamente la aplicación del nodo cuando se detectan cambios en los archivos del directorio.
Chart.js	Biblioteca de JavaScript de código abierto para la visualización de datos, que admite 8 tipos de gráficos.

Figura 1. Importación de paquetes y dependencias

```
server.js | m
const express = require('express');
const path = require('path');
const exphbs = require('express-handlebars')
const methodOverride = require("method-override")
const morgan = require('morgan')
const flash = require('connect-flash');
const session = require('express-session');
const passport = require('passport');
```

En la Figura 2 se puede observar el diagrama de flujo general simplificado de la

plataforma. En la plataforma puede haber cuatro tipos de usuarios: médico, empleado, paciente y administrador, en la Figura 3 se muestra el diagrama de caso de uso de la plataforma, se pueden ver las funciones de cada uno de los perfiles. Finalmente, en la Figura 4 se muestra la inicialización del proyecto, esta se realiza en el archivo index.js.

Figura 2. Diagrama de flujo general de la plataforma

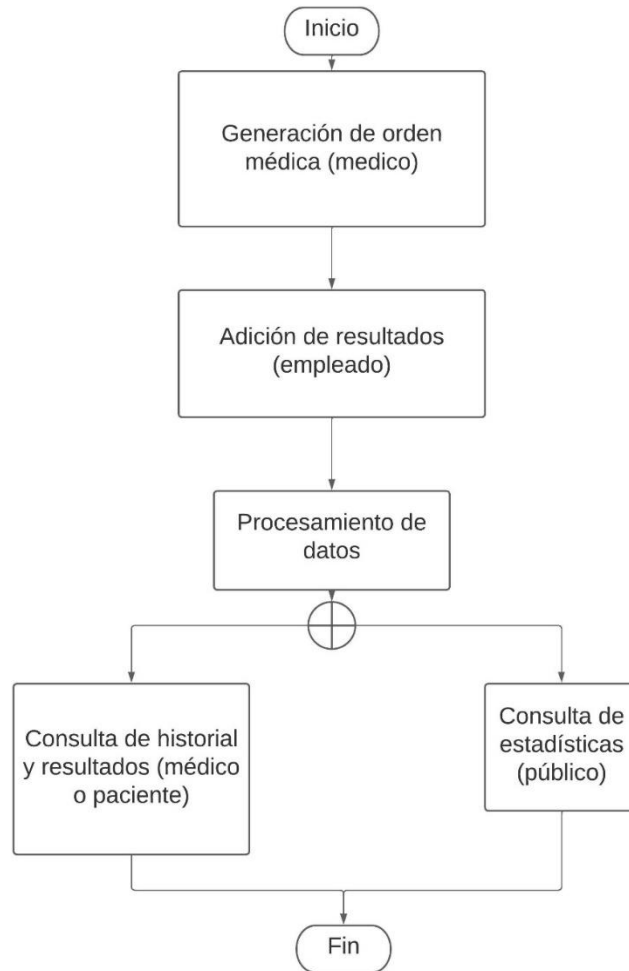


Figura 3. Diagrama de caso de uso de la plataforma

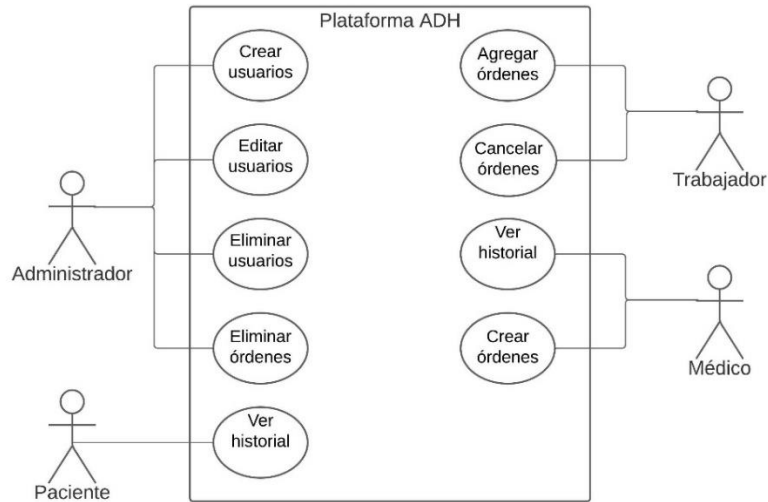


Figura 4. Inicialización de la plataforma en el archivo index.js usando Express js

```

1  require('dotenv').config();
2  const app = require('./server');
3  require('./database');
4
5
6  app.listen(app.get('port'), () => {
7    console.log('Server on port:', app.get('port'))
8  })
9
10

```

5.2. Modelado de la base de datos

Mongoose funciona con el uso de esquemas. Cada esquema se asigna a una colección en MongoDB y define la forma de los documentos dentro de esa colección. Los esquemas tienen una estructura JSON. Para la implementación de este proyecto se desarrollaron dos esquemas, el primero es el esquema de usuarios, “UserSchema”, en donde se almacena la información de todos los usuarios sin importar el tipo de usuario (medico, paciente, empleado o administrador). Sus propiedades de tipo *string* son las siguientes: *identification_type*, *identification*, *name*, *lastname*, *sec_lastname*, *date_of_birth*, *gender*, *blood_type*, *rh*, *marital_status*, *EPS*, *home_phone*, *mobile_phone*, *work_phone*, *address*, *city*, *role*, *email*, *password*, *contact_name*, *contact_lastName*, *contact_sec_lastName*, *contact_relationship* y *contact_phone*. La única propiedad que es distinta a *string*, es *last_login_date* y el tipo de dato es *date*.

El segundo esquema es para los exámenes, “ExamSchema”, en donde se almacena tanto información de los exámenes, como información del paciente al que se le toma la prueba. En la Tabla 2 se pueden observar las propiedades del esquema de los exámenes.

Tabla 2. Propiedades y descripción del esquema "ExamSchema"

Propiedades	Tipo de dato	Descripción
Id	ObjectId	Por defecto, Mongoose añade un ID único a cada documento.
Result	Number	El resultado una vez se ha tomado el examen.
Description	String	Breve análisis con base al resultado.
date_of_exam	String	Fecha en la que se programó el examen.
Dx	String	Dictamen por el cual se programa el examen.
hour_of_exam	String	Hora en la que se programa el examen.
status	String	Estado en el que se encuentre la orden (pendiente, cancelado, completo).
whyCancel	String	Razón por la que se cancela el examen.
nivel	String	El nivel en el que se encuentra el examen (bajo, normal, alto).
user	String	ObjectId del paciente.
Patient	Array	Contiene información del paciente extraída de la colección "users", guarda las propiedades "name", "lastName", "identification", "city" y "date_of_birth".
Age	Number	Edad del paciente.
Group	String	Grupo según rango de edad del paciente.

Para que sea posible trabajar bajo la definición de esquema, es necesario convertir el esquema en un modelo, para ello, se usa el comando de Mongoose:

```
mongoose.model(modelName, schema)
```

En donde modelName será el nombre del modelo y schema es el esquema previamente implementado (Mongoose, 2022) De esta manera, los esquemas "UserSchema" y "ExamSchema" pasan a ser los modelos "User" y "Exam".

5.3. Cifrado de contraseña

El cifrado de contraseña se realiza con *bcrypt*, es necesario crear un método dentro del esquema de usuarios que cifre y retorne la contraseña ya encriptada. la sintaxis de encriptación es la siguiente:

```
bcrypt.hash(string, salt)
```

En donde String es el texto que se quiere encriptar y salt, una cadena de texto aleatorio que hace la contraseña indescifrable (NPM JS, 2022). Así, la variable password del primer esquema, es encriptada y sobrescrita antes de guardarse en la colección Users. En la Figura 5 se observa la aplicación de Bcrypt para el cifrado de la contraseña en la colección Users, se usó un conjunto de 10 bytes para la codificación de la cadena de texto.

Figura 5. Aplicación de Bcrypt para el cifrado de contraseña en la colección usuarios

```
121
122   UserSchema.methods.encryptPassword = async password => {
123     const salt = await bcrypt.genSalt(10);
124     return await bcrypt.hash(password, salt);
125   };
```

5.4. Motor de plantillas

Para la visualización de la plataforma se usó *Handlebars*, ya que permite compilar código tanto del Front-end, como del Back-end y también permite el uso de estructuras de decisión como ciclos y condicionales. Para el estilo de la plataforma, se usó Bootstrap, ya que esta librería ofrece optimización de código y diseño responsivo. Finalmente, para la tipografía se usó la librería Font Awesome, en donde se encuentra una amplia variedad de fuentes e iconos.

5.5. Direccionamiento

El direccionamiento hace referencia a la determinación de cómo responde una aplicación a una solicitud del cliente en un determinado punto final (*endpoint*). Se entiende como *endpoint* a la combinación de un URI (o una vía de acceso) y un método de solicitud HTTP específico (GET, POST, PUT o DELETE) (Expressjs, 2022). En Express.js cuando se establece una ruta, se debe relacionar dicha ruta con un controlador (o handler), cada ruta puede tener una o varias funciones de controlador. La definición de ruta tiene la siguiente estructura:

```
app.METHOD(PATH, HANDLER)
```

Donde:

- app es una instancia de express.
- METHOD es un método de solicitud HTTP.
- PATH es una vía de acceso en el servidor.
- HANDLER es la función que se ejecuta cuando se correlaciona la ruta.

La plataforma web cuenta con un total de 45 rutas, que serán mostradas a medida que se explique los permisos de cada rol y las funciones que presenta la aplicación web. Las rutas están distribuidas en 3 archivos de enrutamiento, que son declarados en el archivo index.js, tal como se muestra en la Figura 6.

Figura 6. Declaración de los archivos de enrutamiento en el archivo index.js

```
51
52   //Routes
53   app.use(require('./routes/index.routes'));
54   app.use(require('./routes/users.routes'));
55   app.use(require('./routes/exams.routes'));
56
57
```

5.5.1. Rutas para el primer uso de la plataforma

Cuando se usa la aplicación por primera vez, el primer perfil que se debe crear es el del administrador, en la Tabla 3, se observan las rutas para la creación de dicho usuario. Como se muestra en la Figura 7, a esta ruta solo se puede acceder mientras no existan documentos en la colección “users”, ya que, si existe algún registro, el servidor redirigirá al usuario a la ruta de inicio (/). El formulario para la creación del primer usuario se muestra en la Figura 8. Una vez creado el primer perfil de administrador, la plataforma es completamente usable.

Tabla 3. Rutas para la creación del primer administrador

Ruta	Metodo	Función
/users/createAdminForm	Get	createAdminForm
/users/createAdmin	Post	createAdmin

Figura 7. Diagrama de flujo de la creación del primer usuario

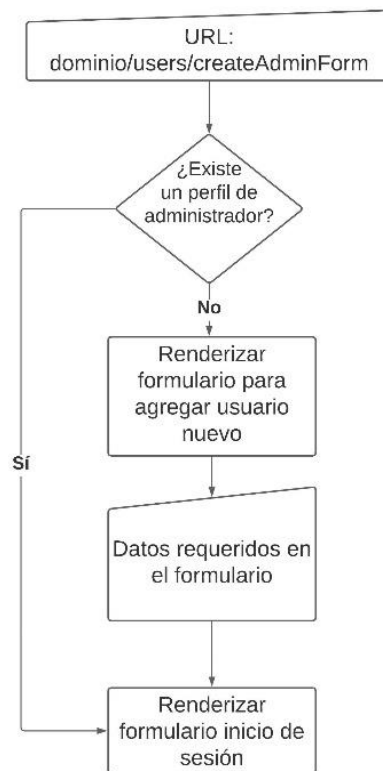


Figura 8. Formulario para la creación del primer usuario

The screenshot shows a web browser window with the URL 'localhost:3000/users/createAdminForm'. The page title is 'Agregar el primer administrador'. On the left, there is a blue sidebar with the 'ADH' logo and 'hormona antidiuretica' text, and navigation links for 'Inicio' and 'Estadísticas'. The main content area contains the following form sections:

- Información personal:** Includes a dropdown for 'Elija un tipo de identificación', a text field for 'Número de identificación', and text fields for 'Nombres', 'Primer apellido', and 'Segundo apellido'.
- Información general:** Includes a date picker for 'Fecha de nacimiento', a dropdown for 'Tipo de sangre', a dropdown for 'Estado civil', a dropdown for 'Genero', a dropdown for 'Rh', and a dropdown for 'EPS'.
- Datos de localización:** Includes text fields for 'Teléfono de la casa', 'Teléfono del trabajo', 'Ciudad', 'Teléfono móvil', 'Dirección', and 'Departamento'.
- Información de sesión:** Includes a dropdown for 'Administrador de la plataforma', a text field for 'Contraseña', a text field for 'Email', and a text field for 'Confirmar contraseña'.
- Información de contacto:** Includes text fields for 'Nombres del contacto', 'Primer apellido del contacto', 'Segundo apellido del contacto', 'Relación o parentesco', and 'Teléfono del contacto'.

5.5.2. Enrutamiento de datos estadísticos

En la aplicación se hace la publicación de datos estadísticos resumidos en tres gráficos:

El primero es un gráfico de barras, en el cual se observa la cantidad de resultados agrupados por nivel (bajo, normal o alto). Para realizar esta agrupación, se realiza una consulta a la base de datos, específicamente a la colección "exams", y se encuentran los documentos cuyo valor de "status" sea "completed".

El segundo gráfico es un diagrama de barras con la cantidad de exámenes realizados y la cantidad de resultados correspondientes fuera del rango normal (densidad de ADH menor a 1 pg/ml o mayor a 5pg/ml) por cada municipio. Para lograrlo, se hace una consulta a la colección "exams" y se buscan los documentos en los cuales la propiedad "status" sea "completed" y la propiedad "result" sea menor a 1 o mayor a 5.

El tercero, es un diagrama circular, se muestra la cantidad de exámenes realizados por grupos de edad. Para realizar esta agrupación, al momento de registrar un nuevo usuario, el aplicativo resta la fecha de nacimiento del usuario a la fecha en la que se está haciendo el registro, el resultado será asignado a la propiedad "age" en la colección users que determinará a qué grupo será asignado. Finalmente, se realiza una consulta a la colección "exams" para encontrar los resultados con la propiedad "status" igual a "completed" y se agrupa con los rangos expuestos en la Tabla 4.

En la Figura 9 se explica el proceso y las queries a las dos colecciones para la visualización de datos estadísticos.

Figura 9. Diagrama de flujo del despliegue de los gráficos estadísticos

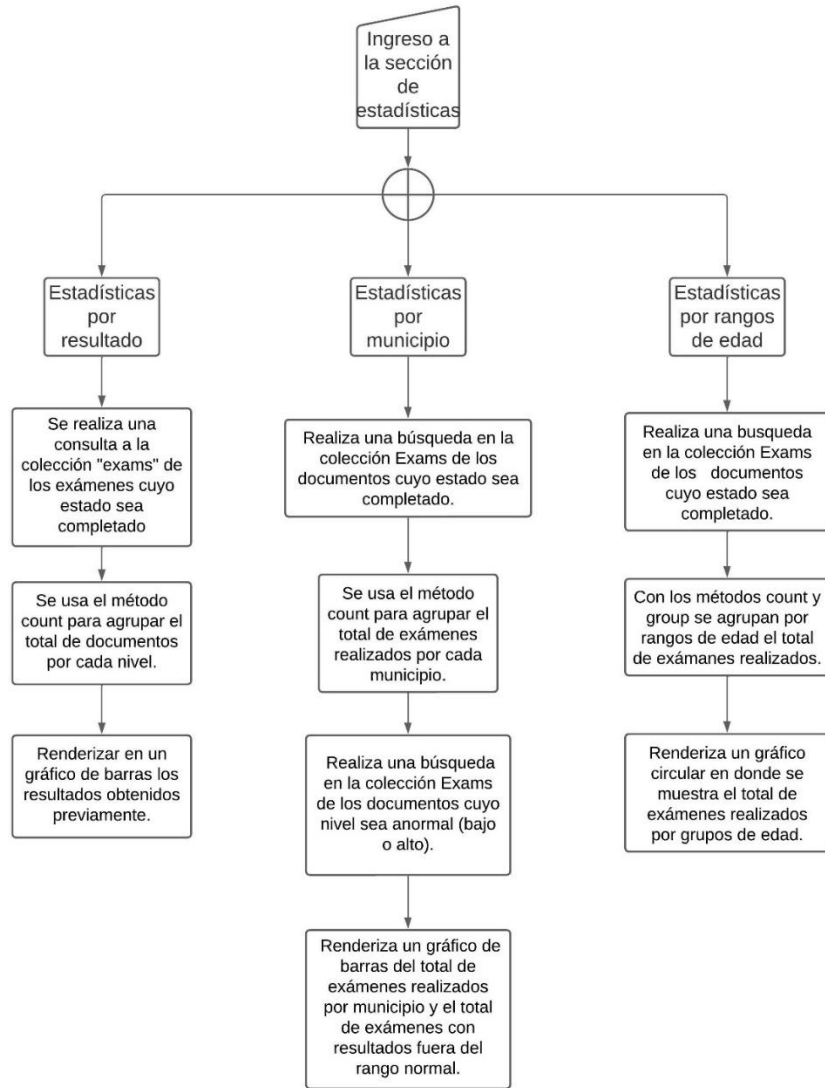


Tabla 4. Posibles valores para la propiedad "group"

Nombre del valor	Rango de edad
"Group 1"	18 y 24 años
"Group 2"	25 y 34 años
"Group 3"	35 y 44 años
"Group 4"	45 y 54 años
"Group 5"	55 y 64 años
"Group 6"	Mayor a 65 años

En la Tabla 5 se observan las rutas de los datos estadísticos y en la Tabla 6, la descripción de los controladores asociados a las rutas. Los gráficos son de acceso público ya que no se muestra información sensible o personal.

Tabla 5. Rutas para la visualización de datos estadísticos

Ruta	Método	Handler
/exams/stadistics/user	Get	renderStadisticsPerUser
/exams/stadistics/examsPerCity	Get	renderStadisticsPerCity
/exams/stadistics/examsPerAge	Get	renderStadisticsPerAge

Tabla 6. Handlers para las rutas estadísticas

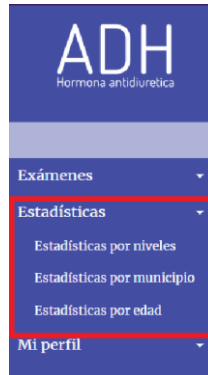
Handler	Descripción
renderStadisticsPerUser	Consulta a la colección Exams y busca los exámenes cuyo estado sea completado (status: done) y con el método count cuenta el total de documentos por cada nivel. Renderiza en un gráfico el total de resultados por niveles (bajo, normal, alto).
renderStadisticsPerCity	Realiza una búsqueda en la colección Exams de los documentos cuyo estado sea completado (status: done) y con el método count entrega el total de exámenes realizados por cada municipio. Realiza una búsqueda en la colección Exams de los documentos cuyo nivel sea bajo (nivel: bajo) y con el método count entrega el total de documentos de esta búsqueda. Renderiza un gráfico de barras del total de exámenes realizados por municipio y el total de exámenes con resultados fuera del rango normal.
renderStadisticsPerAge	Realiza una búsqueda en la colección Exams de los documentos cuyo estado sea completado (status: done) y con el método count entrega el total de exámenes realizados por la propiedad group. Renderiza un gráfico circular en donde se muestra el total de exámenes realizados por grupos de edad.

Para acceder a las estadísticas anteriormente mencionadas, se da click a cualquiera de las opciones del banner (Figura 10) o a través de la sección de estadísticas en la barra lateral (Figura 11), las gráficas estadísticas se muestran en la sección de resultados.

Figura 10. Acceso a la plataforma desde el banner



Figura 11. Acceso a la plataforma desde la barra lateral



5.5.3. Funciones y permisos para cada rol en la aplicación web

En la Figura 12, se puede observar un resumen de las funciones que tiene cada rol dentro del aplicativo. Para asignar permisos, es necesario la ayuda de Helpers que, como su nombre lo indica, brindan ayuda en los procesos intermedios de la aplicación, particularmente en la autenticación del usuario y su correspondiente rol. Por lo cual, se crearon los helpers que se muestran en la Tabla 7 para la identificación del rol del usuario que ha iniciado sesión y permiten o niegan el acceso a rutas específicas.

Figura 12. Funciones disponibles para cada rol

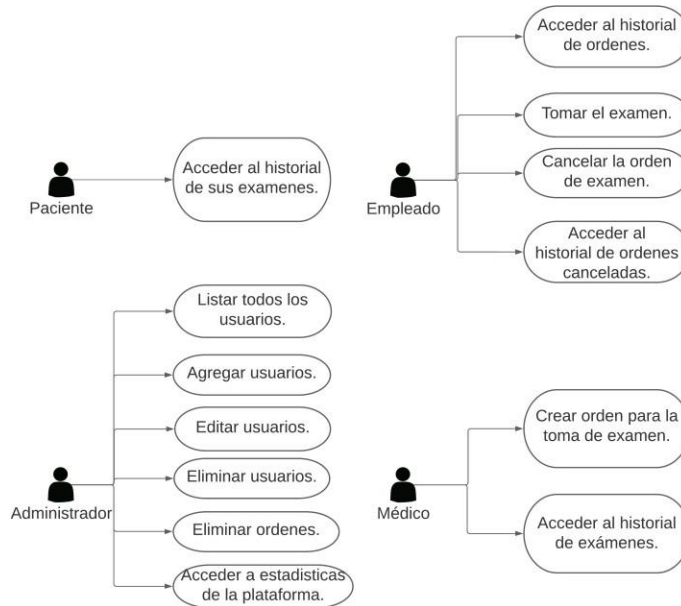


Tabla 7. Helpers usados para la validación de usuario

Helper	Descripción
isAuthenticated	Verifica si se ha iniciado sesión.
isAdmin	Verifica si el rol del usuario es "Administrador".
isLab_Staff	Verifica si el rol del usuario es "Empleado".
isPhysician	Verifica si el rol del usuario es "Medico".

5.5.4. Rutas que no necesitan autenticación de usuario

Además de las rutas de los gráficos estadísticos, las rutas que se encuentran en la Tabla 8 están disponibles para cualquier tipo de rol, tienen funciones comunes y necesarias en la mayoría de las aplicaciones web, como inicio de sesión, cambio de contraseña, cierre de sesión, entre otros.

Tabla 8. Rutas al que tienen acceso todos los roles

Ruta	Metodo	Función
/users/signInForm	Get	signInForm
/users/signIn	Post	signIn
/users/signInError	Get	signInError
/users/started	Get	started
/users/logout	Get	logout
/users/myProfile	Get	myProfile
/users/changePasswd	Get	changePasswdForm
/users/changePasswd	Post	changePasswd
/users/forgotPasswdForm	Get	forgotPasswordForm
/users/forgotPassword	Post	forgotPassword

5.5.5. Funciones de administrador

Las funciones del administrador están enfocadas en la creación, edición y eliminación de los usuarios del aplicativo. El administrador es el único perfil que puede eliminar las órdenes médicas que no han sido tomadas y también tiene acceso a una ruta en donde se muestran las estadísticas generales de la plataforma. En la Tabla 9 se muestran todas las rutas del administrador y en la Tabla 10 la explicación detallada de los handlers correspondientes a la ruta.

Tabla 9. Rutas para las funciones del administrador

Ruta	Metodo	Controlador
/users/createUserForm	Get	createUserForm
/users/createUser	Get	createUser
/users/seeAllPatientsAdmin	Get	seeAllPatientsAdmin
/users/seeAllPhysicianAdmin	Get	seeAllPhysicianAdmin
/users/seeAllLabStaffAdmin	Get	seeAllPhysicianAdmin
/users/seeAllAdmin	Get	seeAllAdmin
/users/seeAllUsersAdmin	Get	seeAllUsersAdmin
/users/findUserByIdentification	Get	findUserByIdentificationForm
/users/findUserByIdentification	Post	findUserByEmail
/users/findUserByEmail	Get	findUserByEmailForm
/users/findUserByEmail	Post	findUserByEmail
/users/seeUserAdminForm/:id	Get	seeUserAdminForm
/users/editUserFormAdmin/:id	Get	editUserFormAdmin
/users/editUserAdmin/:id	Put	editUserAdmin
/users/deleteUserAdmin/:id	Delete	deletUserAdmin
/staff/delete/orders	Get	DeleteORdersList
/exams/staff/delete/:id	Delete	deleteExam
/exams/stadistics/general	Get	generalStadistics

Tabla 10. Descripción de los controladores de las funciones del administrador

Controlador	Descripción
createUserForm	Renderiza el formulario para la creación de un nuevo usuario.
createUser	Crea un nuevo usuario usando el metodo Post.
seeAllPatientsAdmin	Consulta a la base de datos para encontrar a los usuarios con el rol "Paciente", renderiza una vista y muestra en una lista la consulta.
seeAllPhysicianAdmin	Consulta a la base de datos para encontrar a los usuarios con el rol "Medico", renderiza una vista y muestra en una lista la consulta.
seeAllAdmin	Consulta a la base de datos para encontrar a los usuarios con el rol "Admin", renderiza una vista y muestra en una lista la consulta.
seeAllUsersAdmin	Consulta a la base de datos para encontrar a todos los usuarios, sin importar el rol, renderiza una vista y muestra en una lista la consulta.
findUserByIdentificationForm	Renderiza el formulario para la búsqueda por número de identificación.
findUserByIdentification	Usa el método post para realiza la consulta por identificación.
findUserByEmailForm	Renderiza el formulario para la búsqueda por correo electrónico.
findUserByEmail	Usa el método post para realiza la consulta por correo electrónico.
editUserFormAdmin	Renderizar el formulario para la actualización de usuarios.
editUserAdmin	Actualiza los usuarios usando el metodo Put.
deletUserAdmin	Elimina usuarios.
DeleteORDersList	Renderiza el listado de las órdenes pendientes con la opción "eliminar", solo el administrador tiene acceso a este listado.
deleteExam	Elimina la orden seleccionada a través del metodo delete.

5.5.6. Rutas del médico

El médico tiene la función de crear órdenes médicas para la toma de exámenes de ADH, visualizar la lista de exámenes realizados y ver el registro completo de un examen específico. En la Tabla 11 se observan las rutas disponibles para el rol de médico y en la Tabla 12 se explica la función de cada handler.

Tabla 11. Rutas para las funciones del médico

Ruta	Metodo	Controlador
/exams/findUserByIdentification	Get	findUserByIdentificationForm
/exams/findUserByIdentification	Post	findUserByIdentification
/exams/new-exam	Post	CreateNewExam
/exams-done	Get	renderDoneExam
/exams/exam/:id	Get	renderDoneSingleExam

Tabla 12. Descripción de los controladores de las funciones del médico

Controlador	Descripción
findUserByIdentificationForm	Renderiza el formulario para buscar pacientes con su número de cédula.
findUserByIdentification	Realiza la búsqueda del número de cédula a través del método post.
CreateNewExam	Crea una nueva de orden de examen
renderDoneExam	Hace una consulta a la base de datos y renderiza los exámenes con estado realizado ("Done").
renderDoneSingleExam	Renderiza la información del examen que ya fue tomado, muestra el diagnostico diferencial, el resultado, comentarios adicionales, nombre del paciente, documento, entre otros.

El médico tiene dos funciones: generar órdenes médicas y visualizar el historial de los pacientes. En la Figura 13 se observa el diagrama de flujo para la generación de órdenes médicas. El formulario para la búsqueda del paciente por cédula se muestra en la Figura 14, si el paciente se encuentra en el sistema, el médico deberá dar click en el botón "Generar orden" (ver Figura 15) y se desplegará un formulario como el de la Figura 16, el médico ingresa los datos solicitados, da click en el botón "generar" y finalmente se muestra la orden como en la Figura 17.

Figura 13. Diagrama de flujo para la generación de ordenes médicas

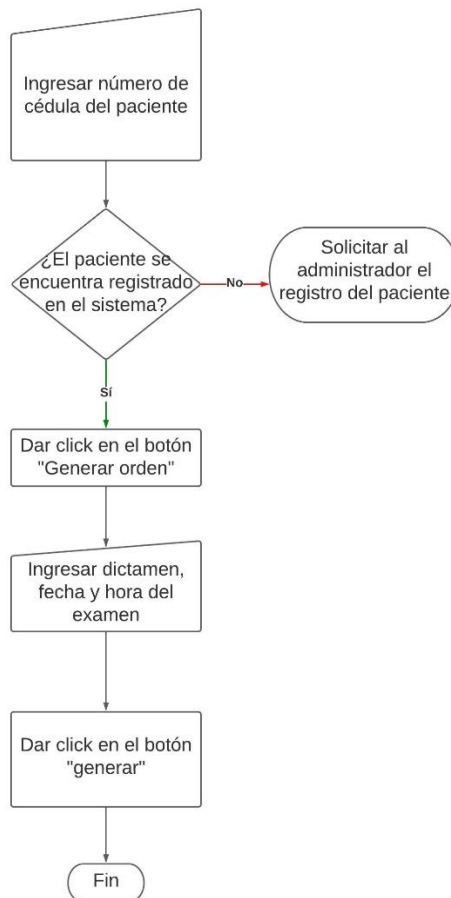


Figura 14. Formulario de búsqueda de usuario por cédula

The screenshot shows the ADH web application interface. On the left is a dark blue sidebar with the ADH logo and navigation links: Inicio, Exámenes, Estadísticas, and Mi perfil. The main content area is white and features a search form titled 'Búsqueda de usuario por cédula'. The form includes a text input field labeled 'Número de identificación' and a 'Buscar' button. In the top right corner, there is a 'Cerrar sesión' button. The footer contains logos for the 'SECRETARÍA DE SALUD DEPARTAMENTO DEL HUILA', 'SIGUENOS', 'Otras entidades', and 'GETI'.

Figura 15. Botón de generación de orden médica

Usuarios

Nombres	Primer apellido	Segundo apellido	Identificación	Correo electrónico	Tipo	Agendar
Bertha Cecilia	Guzmán	Guzmán	26449873	berthacecilia@gmail.com	Paciente	Generar orden

Figura 16. Formulario de generación

Orden de examen "Prueba de ADH"

Secretaría de salud
 Departamento del Huila
 Nit. 900434629-1
 Calle 9 N° 15- 25
 Tel. 8715907- 3118848270

Paciente: Bertha Cecilia Guzmán.
Documento: 26449873.

Código 914102	Prestación Hormona Antidiurética	Muestra Plasma con EDTA Congelado: 8 mL. Enviar 2 tubos
-------------------------	--	---

Dx:

Fecha de examen: Hora de examen:

Figura 17. Visualización de orden generada

Módulo toma de examen

Secretaría de salud

Departamento del Huila

Nit. 900434629-1
Calle 9 N° 15- 25
Tel. 8715907- 3118848270

Paciente: Bertha Guzmán.
Documento: 26449873.
Fecha programada: 2022-08-17.
Hora programada: 2:00 p.m..

Dictamen previo:
Dx

Resultado:	Nivel:
4.3 pg/ml	normal

Observaciones
El paciente se encuentra bien.

Volver

5.5.7. Rutas del empleado

El empleado es el que revisa las órdenes que están pendientes y puede tomar el examen o cancelar la orden, en la Tabla 13 se muestran las rutas para el empleado y en la Tabla 14, la descripción de cada uno de los handlers.

Tabla 13. Rutas para las funciones del empleado

Ruta	Metodo	Controlador
/staff/orders	Get	renderCreatedOrders
/exams/edit/:id	Get	renderEditForm
/exams/edit/:id	Put	updateExam
/exams-canceled	Get	renderCanceledOrders
/exams/cancelOrder/:id	Get	cancelOrderForm
/exams/cancelOrder/:id	Put	cancelOrder

En la Figura 18 se muestra el diagrama de flujo para la adición de resultados o cancelación de órdenes médicas. Para cualquiera de las dos opciones, el trabajador debe dar click en el botón “acción” de la lista de órdenes (ver Figura 19), si se da click en “Tomar examen” se despliega el formulario de la Figura 20, si se da click en el botón de “Cancelar orden” se despliega el formulario de la Figura 21. En la Figura 22 se muestra una vista de la lista de las órdenes canceladas.

Tabla 14. Descripción de los controladores de las funciones del empleado

Controlador	Descripción
renderCreatedOrders	Realiza una consulta a la base de datos y renderiza el total de exámenes para el día actual, su respectivo listado y un listado de los exámenes pendientes posteriores.
renderEditForm	Renderiza el formulario para agregar los resultados correspondiente a la orden.
updateExam	Actualiza el documento correspondiente a la orden de la que se tomó el examen.
renderCanceledOrders	Realiza una consulta a la base de datos y renderiza un listado de las órdenes que fueron canceladas por los empleados.
cancelOrderForm	Renderiza el formulario para la cancelación de exámenes.
cancelOrder	Cambia el estado de la orden creada a cancelada a través del metodo update.

Figura 18. Diagrama de flujo para la adición de resultados

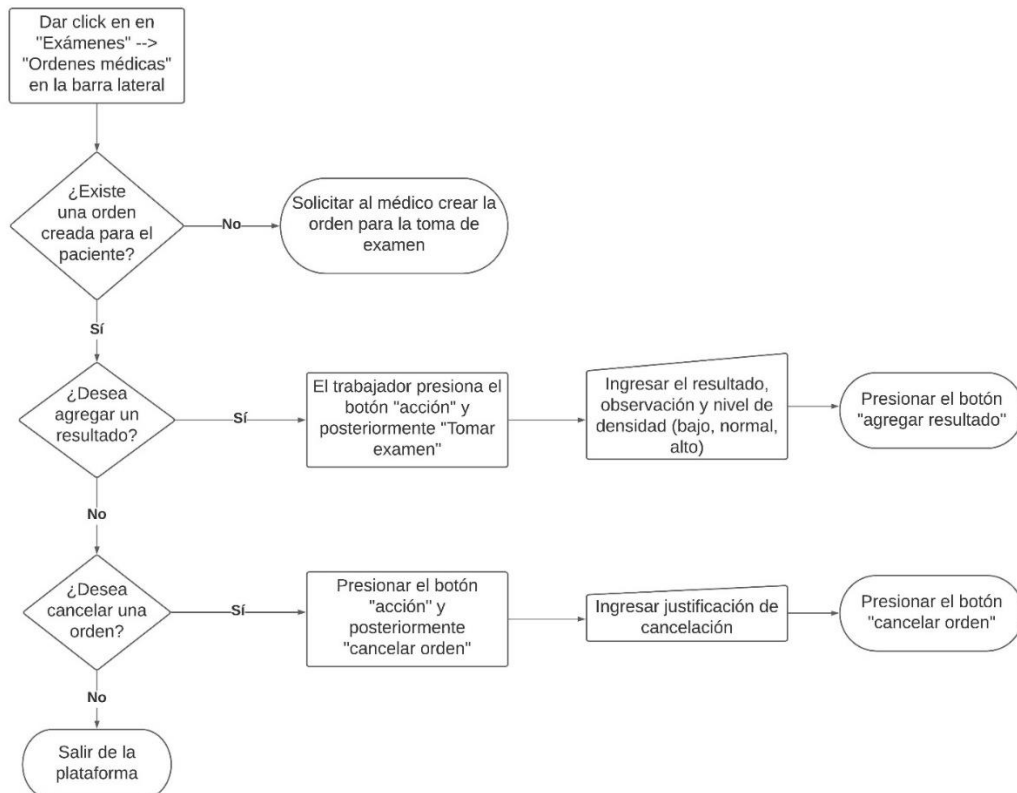


Figura 19. Despliegue del botón "Acción"

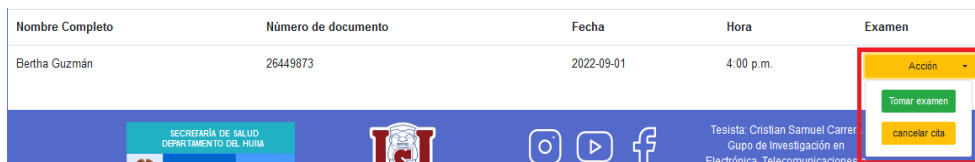


Figura 20. Formulario para la adición de resultados

Módulo toma de examen

Secretaría de salud
Departamento del Huila
 Nit. 900434829-1
 Calle 9 N° 15- 25
 Tel. 8715907- 3118848270

Paciente: Bertha Guzmán.
Documento: 26449873.
Fecha programada: 2022-09-01.
Hora programada: 4:00 p.m..

Dictamen previo:
 El paciente requiere de este examen porque...

Publicación de resultados:

Resultado en pg/ml

Observaciones

Nivel Bajo

Guardar

Figura 21. Formulario para la cancelación de resultados

Cancelación de examen.

Secretaría de salud
Departamento del Huila
 Nit. 900434829-1
 Calle 9 N° 15- 25
 Tel. 8715907- 3118848270

Paciente: Bertha Guzmán.
Documento: 26449873.
Fecha programada: 2022-09-01.
Hora programada: 4:00 p.m..

Cancelar cita

Razón por la cual cancela la cita:

Cancelar

Figura 22. Listado de exámenes cancelados

Nombre Completo	Número de documento	Fecha	Hora	Razón
Bertha Guzmán	26449873	2022-07-28	6:00 p.m.	Just because.
Bertha Guzmán	26449873	2022-07-27	1:00 p.m.	El paciente no asistió.

5.5.8. Rutas del paciente

Solo hay una ruta disponible para los pacientes, /exams-patient, la cual utiliza el método GET para mostrarle al usuario los exámenes realizados y que coincidan con su objectID. En la Figura 23 se muestra el listado de un usuario de prueba, si se da click en el botón “Ver examen”, se despliega el historial de dicho examen como se observa en la Figura 24.

Figura 23. Visualización de exámenes realizados por paciente

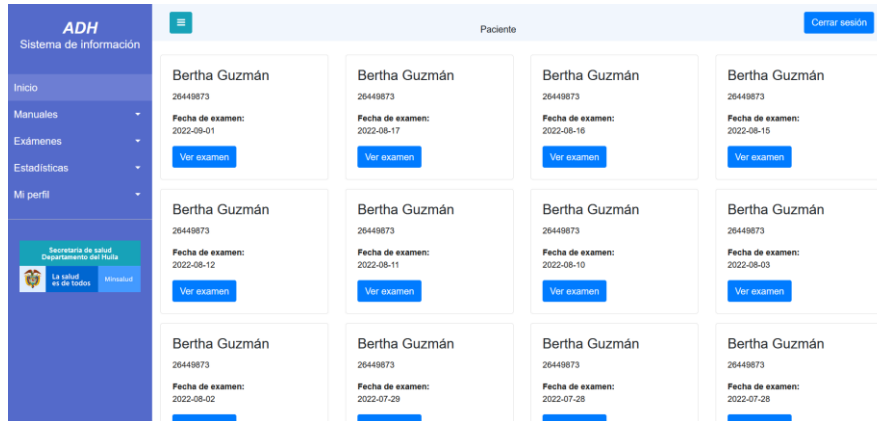


Figura 24. Visualización de examen individual



6. RESULTADOS

Como resultado de este trabajo se implementó un aplicativo web bajo el framework MERN (MongoDB, Express, React y Node) capaz de digitalizar el proceso de ordenamiento, publicación y distribución de resultados. El aplicativo cuenta con una base de datos no relacional, implementada con Mongo DB, y contiene dos colecciones: “Users” y “Exams”. Esta base de datos fue utilizada debido a que presenta una mayor escalabilidad con un menor costo computacional, a diferencia de las bases de datos relacionales. Adicionalmente, las bases tipo noSQL son más versátiles y soportan el crecimiento horizontal; lo cual es conveniente para este proyecto ya que el aplicativo web propuesto podría ser considerado como un producto mínimo viable (MVP) y será necesario seguir con su desarrollo. Respecto al uso de Express.js, este entorno de trabajo permitió la unificación de un solo lenguaje para el front-end y el back-end, facilitando el proceso de desarrollo. Finalmente, mediante React y Handlebars fue posible actualizar y renderizar de manera eficiente los componentes con el fin de implementar una aplicación ligera.

Para confirmar el correcto funcionamiento de la plataforma se crearon 12 perfiles de prueba distribuidos como se muestra en la Tabla 15. De los pacientes, 4 fueron registrados como residentes de la ciudad de Neiva, 2 del municipio de Algeciras, 1 del municipio de Campoalegre y 1 del municipio de Rivera.

Tabla 15. Distribución de roles ficticios para la realización de pruebas

Rol	Cantidad
Médico	2
Empleado	1
Administrador	1
Paciente	8

Se crearon 35 órdenes para toma de examen y se agregaron sus correspondientes resultados, de los cuales 2 son considerados como baja densidad, 9 como alta densidad y 24 como nivel estable. Los resultados se muestran en la plataforma en un gráfico de barras (como se observa en la Figura 25).

Usando los mismos exámenes anteriormente mencionados, se implementó un segundo diagrama de barras en donde se muestra el total de exámenes realizados y el total de resultados fuera del rango estable por municipio, tal como se observa en la Figura 26. Voluntariamente no se asignó orden de examen al paciente registrado en Rivera, por lo tanto la gráfica solo muestra resultados de Neiva, Campoalegre y Algeciras.

Figura 25. Grafica en la plataforma web con los resultados de prueba agrupados por nivel

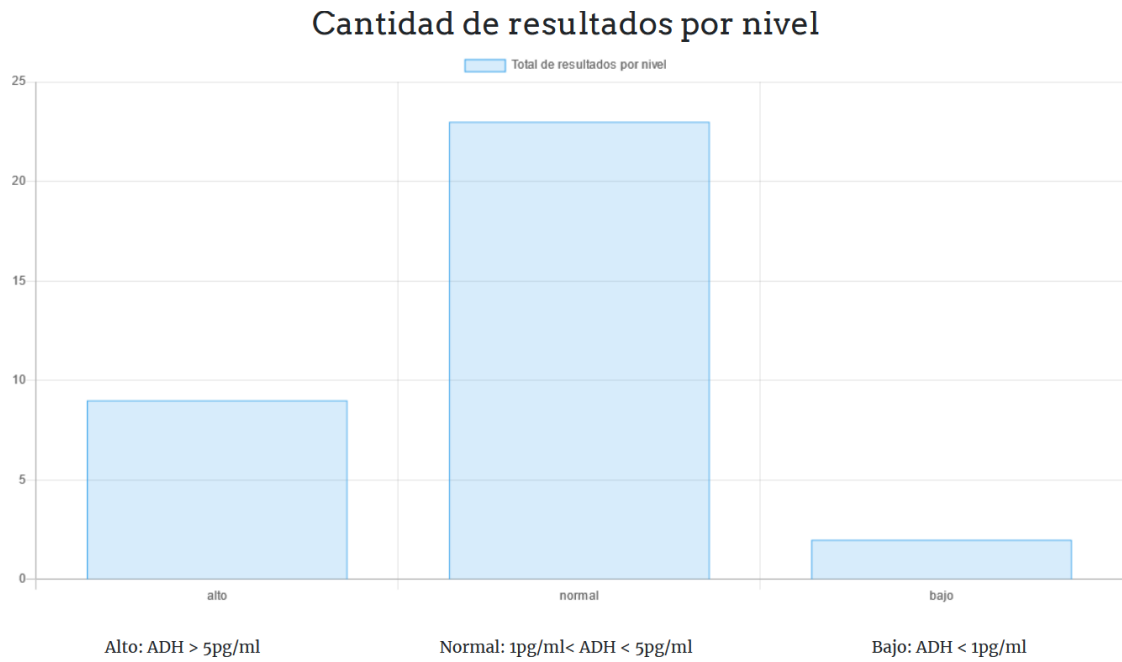
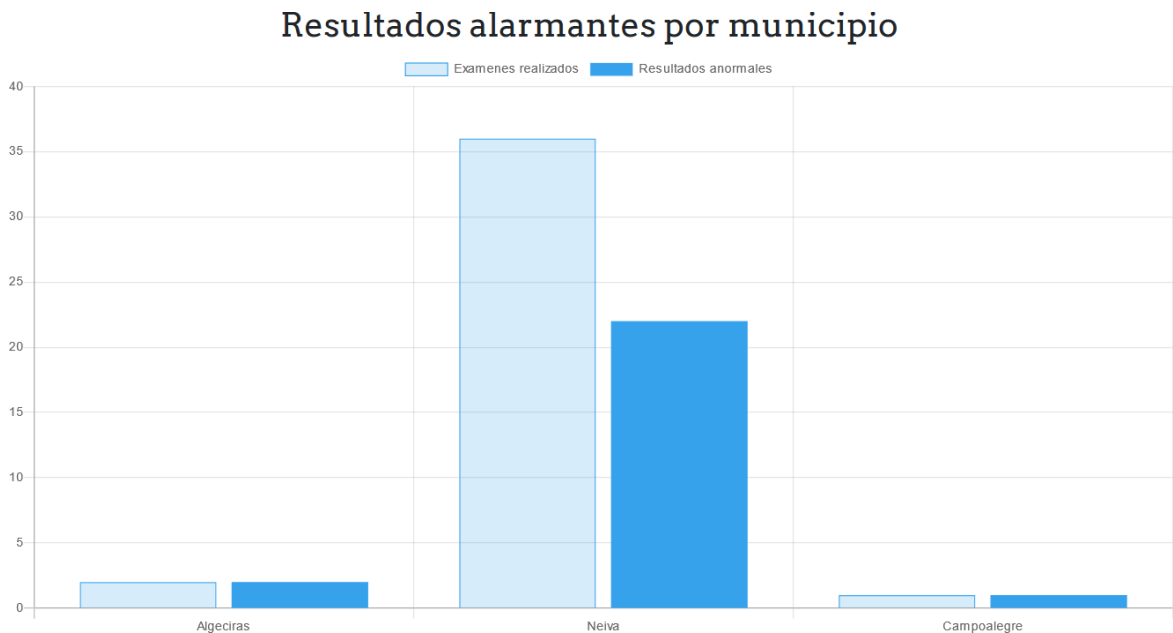


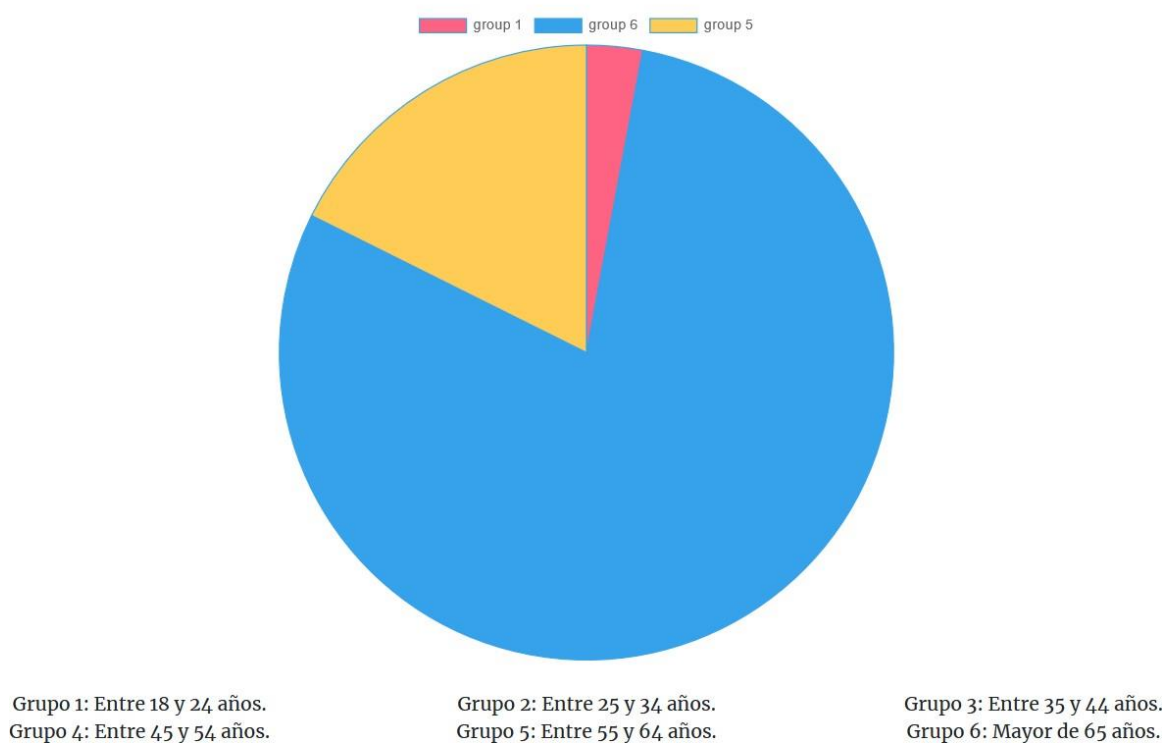
Figura 26. Visualización de la agrupación del total de exámenes y resultados fuera del rango común por cada municipio



El último gráfico que se muestra en la plataforma, es un gráfico circular, en donde se agrupan los resultados por rango de edad, como se muestra en la *Figura 27*.

Figura 27. diagrama circular de exámenes completados agrupados por rango de edad

Exámenes completados agrupados por edad



Se debe tener en cuenta que en la plataforma es posible asignarle una orden y resultado al mismo usuario más de una vez, por lo tanto los gráficos mostrados anteriormente son consecuencia de la asignación repetida a cualquiera de los 8 pacientes registrados.

Con el fin de obtener no solo datos estadísticos sobre los pacientes, sino también sobre la plataforma, se implementó un dashboard al que solo tiene acceso el administrador; en el cual se muestra el número de exámenes realizados en el presente mes y en el mes anterior, la cifra de exámenes cancelados, órdenes pendientes, pacientes registrados, y órdenes emitidas. El dashboard se observa en la *Figura 28*.

Figura 28. Dashboard con datos de la plataforma



En resumen, con el aplicativo web desarrollado en esta tesis se introduce la cuantificación y seguimiento de resultados médicos en el departamento del Huila. Lo anterior, mediante la agrupación de los resultados de cada paciente por edad, municipio de residencia, género y otras características de forma automatizada. Por lo tanto, las causas y tratamientos preventivos de ciertas enfermedades podrán identificarse fácilmente; lo cual podría promover la disminución de las tasas de incidencia y mortalidad de dichas enfermedades en el departamento del Huila. Por otra parte, la explicación del uso del aplicativo web se encuentra en el manual del paciente, manual del personal médico y manual del programador.

7. CONCLUSIONES

- Se implementó un aplicativo web capaz de digitalizar los procesos de ordenamiento y agendamiento de citas, toma de exámenes de la prueba de ADH y la publicación de su correspondiente resultado. Con este tipo de aplicativos se busca mejorar los índices de salud en el departamento del Huila, ya que permite hacer rastreo y seguimiento de enfermedades teniendo en cuenta variables como edad, municipio, género, etc. con lo que será más fácil tomar medidas de prevención.
- Se realizó una correcta estructuración del aplicativo web asignando tareas, permisos y funciones a los distintos roles que intervienen en la plataforma. Los accesos y permisos fueron posibles gracias al uso del módulo *Passport.js* y los middlewares de autorización.
- Se implementó una base de datos no relacional (MongoDB) con dos colecciones; una destinada para la información de los usuarios y otra, para la información de las órdenes médicas y exámenes. Gracias a la flexibilidad y fácil escalabilidad que ofrecen las bases de datos no relacionales, se podrá seguir desarrollando este proyecto sin verse en la obligación de realizar grandes cambios en los documentos ya existentes.
- El aplicativo web realiza el tratamiento de datos para que cualquier persona con acceso a internet pueda ver tres agrupaciones estadísticas distintas: total de resultados agrupados por nivel (bajo, normal, alto); total de exámenes realizados junto con el total de exámenes con resultados fuera del rango normal por municipio y, finalmente; total de exámenes realizados agrupados por rangos de edad. Además el administrador tiene acceso a un panel con estadísticas y cifras detalladas de la plataforma.
- Se trabajó bajo el framework MERN, lo que permitió abarcar todo el ciclo de desarrollo, desde el front-end hasta el back-end

8. TRABAJO FUTURO

Es imprescindible continuar en el avance tecnológico y en especial en la cuantificación y seguimiento de variables médicas para llevar más control y crear medidas de prevención en el sector de la salud, por esta razón este proyecto puede continuar en su proceso de evolución y mejora. A continuación, se plantean posibles mejoras que se pueden implementar a futuro.

- Añadir una sección en donde se encuentren los pacientes que tienen resultados fuera del rango normal de ADH y que estén o hayan empezado tratamiento médico para controlarlo.
- Adicionar más variedad de exámenes a este aplicativo web, de esta manera se obtiene un panorama más amplio de la salud en el departamento del Huila y las posibles medidas de prevención y corrección.
- Migrar la plataforma web a aplicativo móvil para que la interacción sea más rápida y personalizada.
- La sección de estadísticas por municipio podría ser más amigable si en lugar de un diagrama de barras se mostrara un mapa político del Huila interactivo y cuando se coloque el puntero del mouse, muestre las estadísticas en cada municipio.

REFERENCIAS

- Calderón, D. O. (2019). *Aplicación web de referencias y contrareferencias de pacientes del seguro integral de salud*. Lima, Perú.
- Carrión, V. (2015). *Desarrollo de una aplicación web basada en el modelo vista controlador para la gestión de las historias clínicas de los pacientes en el centro de salud de San Jerónimo*. Andahuaylas.
- Chart.js. (2023). *chartjs.org*. Obtenido de <https://www.chartjs.org/docs/latest/>
- Chiroles, M., Pacheco, Y., Reyes, R., & Sisto, A. (2017). Implementación de una aplicación web para el módulo servicio quirúrgico de la aplicación Behique. *Revista de ciencias médicas de Pinar del Río*.
- Express.js. (2017). *Express- Node.js web application*. Obtenido de <https://expressjs.com/>
- Expressjs. (2022). *Direccionamiento básico de Express*. Obtenido de <https://expressjs.com/es/starter/basic-routing.html>
- INSTITUTO NACIONAL DEL CÁNCER. (2022). *Diccionarios del NCI*. Obtenido de <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/vasopresina>
- Instituto Nacional del Cáncer. (2022). *Instituto Nacional del Cáncer*. Obtenido de <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/hormona-antidiuretica>
- Labtestsonline. (2023). *Labtestsonline*. Obtenido de <https://www.labtestsonline.es/tests/adh>
- Maradei, M. F., Espinel, F., & Diaz, L. (2008). Diseño De Una Plataforma Informática Para El Almacenamiento y Análisis de Datos Antropométricos. *Antropos 2.0. Revista UIS Ingenierías*.
- MedlinePlus. (2022). *Díabetes insípida: MedlinePlus*. Obtenido de <https://medlineplus.gov/spanish/ency/article/000377.htm>
- MedlinePlus. (2022). *MedlinePlus*. Obtenido de <https://medlineplus.gov/spanish/ency/article/000314.htm>
- MedlinePlus. (2022). *Estadísticas de salud*. Obtenido de <https://medlineplus.gov/spanish/healthstatistics.html>

Mochãoa, H., Gonçalves, D., Alexandre, L., Castro, C., Valério, D., Barahona, P., . . . Costa, R. S. (2021). IPOscore: An interactive web-based platform for postoperative surgical complications analysis and prediction in the oncology domain. *Elseiver*.

MongoDB. (2022). *MongoDB Atlas*. Obtenido de https://www.mongodb.com/cloud/atlas/lp/try4?utm_source=google&utm_campaign=gs_americas_colombia_search_core_brand_atlas_desktop&utm_term=mongo&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624317&adgroup=115749712463&gclid=CjwKCAjw6raYBhB7Ei

Mongoose. (2022). *Mongoose js*. Obtenido de <https://mongoosejs.com/docs/guide.html>

Mongoose. (2022). *Mongoose: Schemas*. Obtenido de <https://mongoosejs.com/docs/guide.html>

Mozilla Developer Network Web docs. (2022). *Qué es JavaScript*. Obtenido de https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

NPM JS. (2022). *bcryptjs- npm*. Obtenido de <https://www.npmjs.com/package/bcryptjs>

Oracle Colombia. (2022). *¿Qué es una base de datos?* Obtenido de <https://www.oracle.com/co/database/what-is-database/>

UC San Diego Health. (2022). *UC San Diego Health, Health Library*. Obtenido de https://myhealth.ucsd.edu/RelatedItems/167,antidiuretic_hormone_ES