

**Universidad de Monterrey**  
**Integración de Aplicaciones Computacionales**

**Heartguard: Sistema Predictivo para  
Emergencias Cardiovasculares**

**Reporte Formal**

**Segundo Entrega – Proyecto de Integración de Sistemas  
Computacionales**

**11 de octubre de 2025**

**PhD. Raúl Morales Salcedo**

**Equipo 1**

**Aldo Elio Peña Salas - 635861**

**Pablo Celedón Cabriaes - 597730**

*Damos nuestra palabra que hemos realizado esta actividad con integridad académica*

## **ÍNDICE**

<b>OBJETIVO DEL MÓDULO BACKEND (CMS).....</b>	<b>3</b>
Módulo 1.....	3
<b>Descripción detallada de las funcionalidades implementadas.....</b>	<b>3</b>
<b>Listado y explicación de queries SQL y Stored Procedures utilizados.....</b>	<b>4</b>
<b>Diseño del Diagrama ER y Modelo de Datos.....</b>	<b>6</b>
<b>Presentación de KPIs, métricas y gráficas.....</b>	<b>9</b>
<b>Capturas de pantalla del sistema en funcionamiento.....</b>	<b>10</b>
<b>Conclusiones.....</b>	<b>13</b>

# OBJETIVO DEL MÓDULO BACKEND (CMS)

## Módulo 1

El objetivo del Módulo 1 fue desarrollar un sistema Back End CMS (Content Management System) completamente funcional para la plataforma Heartguard: Sistema predictivo para emergencias cardiovasculares. Este módulo permite la administración integral de usuarios, pacientes, doctores, citas y alertas médicas, garantizando la integridad y trazabilidad de la información médica mediante autenticación por roles, procedimientos almacenados, y conexión directa a una base de datos PostgreSQL normalizada hasta tercera forma normal (3FN).

El CMS tiene como propósito ofrecer una interfaz de gestión interna que facilite las operaciones CRUD (crear, leer, actualizar y eliminar), el monitoreo de indicadores clave de desempeño (KPIs) y la generación de reportes analíticos con visualizaciones gráficas que representen el comportamiento del sistema y sus usuarios en tiempo real.

## Descripción detallada de las funcionalidades implementadas

El sistema Heartguard CMS fue desarrollado como una aplicación web monolítica utilizando el lenguaje Go (Golang), el motor de plantillas html/template, y la base de datos PostgreSQL 16 ejecutada en un contenedor Docker.

Todas las operaciones de manipulación de datos se realizan exclusivamente mediante Stored Procedures y Functions, asegurando la separación lógica de la capa de negocio y la capa de datos.

Las principales funcionalidades implementadas son:

- **Autenticación y roles de usuario:**

Se incluye un sistema de inicio de sesión con verificación de credenciales y control de acceso basado en roles, diferenciando entre *administrador* y *doctor*. El administrador tiene acceso total a las operaciones de gestión, mientras que el doctor solo puede consultar y actualizar registros de sus pacientes asignados.

- **Gestión de entidades principales (CRUD):**

Se implementaron formularios y tablas dinámicas para administrar las entidades del

sistema: *Usuarios, Pacientes, Doctores, Citas y Alertas*.

Cada acción se comunica con la base de datos mediante procedimientos almacenados como `sp_create_patient`, `sp_update_patient`, `sp_delete_patient`, y funciones de lectura como `fn_get_all_patients()`.

- **Reportes tabulares y exportación:**

El CMS incluye reportes filtrables por fecha, estado y tipo de contenido. Además, los resultados de pueden exportarse en formato CSV para análisis externos on la librería estándar `encoding/csv`.

- **Indicadores analíticos (KPIs):**

En el panel principal del administrador se despliegan valores estadísticos en tiempo real como número total de pacientes, doctores activos, citas programadas para el día y alertas registradas en las últimas 24 horas.

- **Visualización gráfica:**

Se integraron seis gráficas interactivas mediante **Chart.js**, representando métricas como:

1. Total de usuarios registrados.
2. Pacientes activos por mes.
3. Citas por tipo (`Appointment_Type`)
4. Distribución por género.
5. Alertas por nivel de riesgo.
6. Top 5 doctores por pacientes.

## Listado y explicación de queries SQL y Stored Procedures utilizados

El diseño de la base de datos se encuentra totalmente normalizado hasta la **Tercera Forma Normal (3FN)**, cumpliendo con las dependencias funcionales y las relaciones establecidas en el **diagrama entidad-relación (ER)** del proyecto.

A continuación se describen los principales procedimientos y funciones empleados:

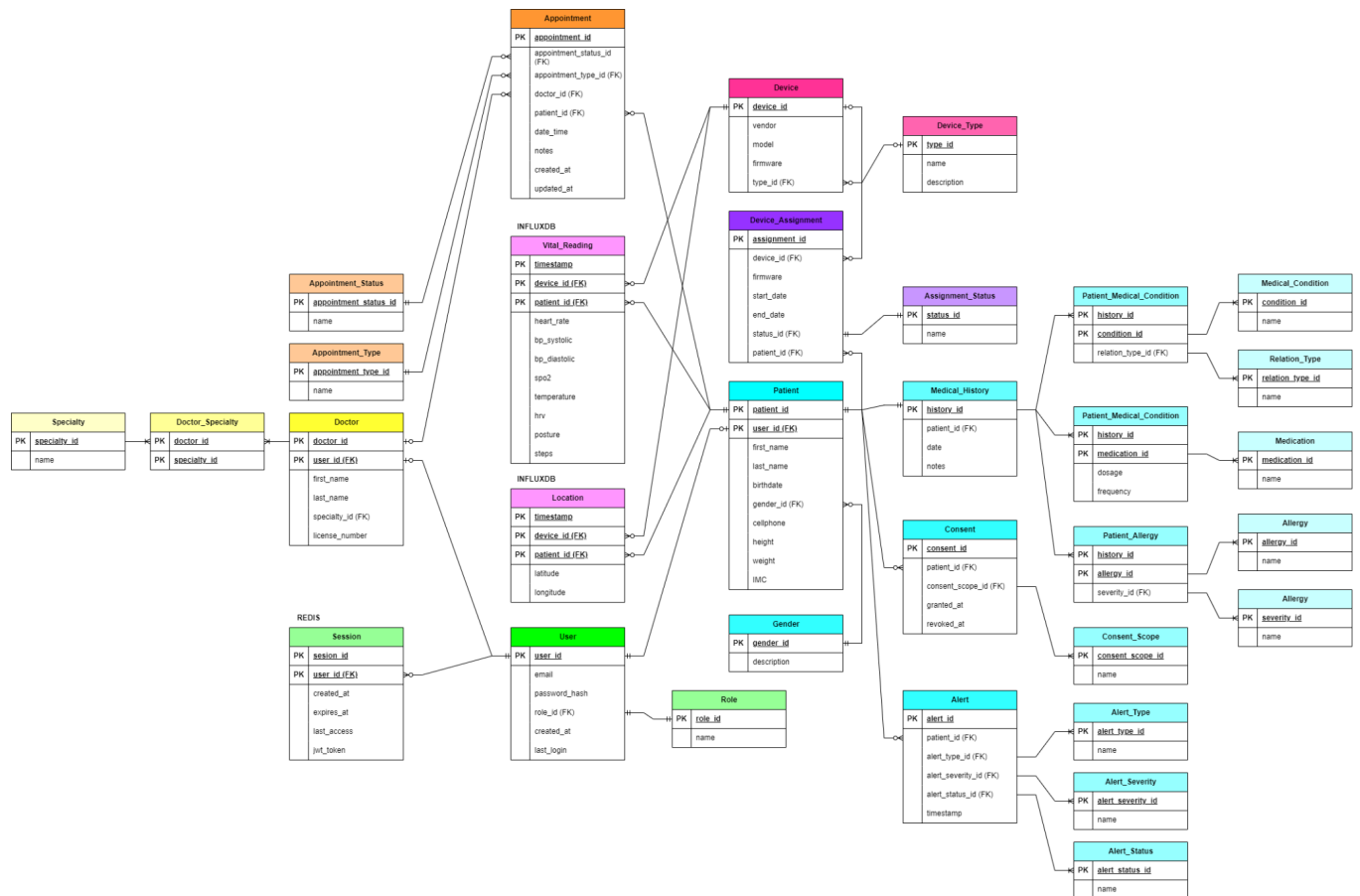
- **`sp_create_patient(first_name, last_name, birthdate, gender_id, cellphone, height, weight)`**

Inserta un nuevo paciente en la tabla patient y calcula su índice de masa corporal (IMC) automáticamente.

- **sp\_update\_patient(patient\_id, first\_name, last\_name, birthdate, gender\_id, cellphone, height, weight)**  
Permite modificar los datos de un paciente existente y recalcula el IMC si cambian peso o altura.
- **sp\_delete\_patient(patient\_id)**  
Realiza un *soft delete*, actualizando el campo deleted\_at sin eliminar el registro físico para conservar trazabilidad.
- **fn\_get\_all\_patients()**  
Devuelve una lista completa de pacientes activos, incluyendo nombre, fecha de nacimiento, género, teléfono, peso, altura e IMC.
- **sp\_create\_user(email, password\_hash, role\_id)**  
Inserta nuevos usuarios en el sistema con contraseñas encriptadas mediante bcrypt.
- **fn\_get\_user\_by\_email(email)**  
Retorna la información del usuario para la autenticación y validación de credenciales.
- **sp\_create\_appointment(patient\_id, doctor\_id, date, description)**  
Registra nuevas citas médicas y vincula los datos con el historial del paciente.
- **fn\_get\_alerts\_by\_status(status)**  
Recupera todas las alertas médicas clasificadas por tipo o nivel de riesgo.
- **sp\_create\_doctor**  
Inserta nuevo doctor y asigna especialidad.
- **fn\_count\_kpis\_dashboard**  
Devuelve KPIs del panel (totales).
- **sp\_create\_alert**

Inserta alerta en tabla Alert con severidad.

## Diseño del Diagrama ER y Modelo de Datos



El diseño de la base de datos de Heartguard evolucionó significativamente entre la versión inicial y la versión final. En la primera propuesta, el modelo se centraba únicamente en la entidad principal Usuario, vinculada a registros básicos de Signos Vitales, Ubicación GPS, KPIs diarios e Inferencias de Riesgo. Este diseño representaba una estructura inicial funcional, pero con un alcance limitado: las relaciones eran principalmente uno a muchos (1:N), sin un control explícito de especializaciones, roles, ni normalización avanzada. Además, las entidades estaban orientadas a un único tipo de usuario (paciente), sin distinción entre doctores, administradores ni dispositivos médicos, lo cual restringía la escalabilidad del sistema.

En la versión actual de nuestro diagrama, la arquitectura del modelo fue completamente reestructurada y normalizada hasta Tercera Forma Normal (3FN), introduciendo una jerarquía más clara y modular que distingue las funciones de cada componente del ecosistema. La nueva versión incorpora entidades clave como en los res pilares fundamentales: usuarios, pacientes y dispositivos médicos. En versiones anteriores, la base de datos se limitaba a unas pocas entidades (Usuario, Signos Vitales, KPI Diario, GPS e Inferencia de Riesgo), con relaciones simples y sin control de roles. La nueva versión amplía drásticamente su alcance mediante la introducción de más de veinte entidades relacionadas, cada una con una función clara dentro del ecosistema médico digital, logrando una representación completa del proceso clínico desde la monitorización del paciente hasta la gestión de citas, alertas y consentimientos médicos.

La entidad User actúa como punto central del sistema y representa a todos los usuarios registrados, incluyendo doctores, pacientes y administradores. Contiene información esencial de autenticación como el correo electrónico (email), la contraseña encriptada (password\_hash), el rol (role\_id) y fechas de registro y acceso. Este diseño permite implementar una autenticación basada en JWT Tokens y almacenamiento de sesiones en Redis, a través de la entidad Session, que guarda los tokens activos, fechas de creación y expiración, garantizando seguridad y trazabilidad.

El sistema distingue claramente entre usuarios administrativos y pacientes clínicos. Para ello, la entidad Patient está vinculada uno a uno con User, agregando información clínica como nombre, apellido, fecha de nacimiento, género, altura, peso e índice de masa corporal (IMC). Paralelamente, la entidad Doctor mantiene la relación entre los doctores y sus especialidades mediante la tabla intermedia Doctor\_Specialty, y además se conecta con las citas médicas (Appointment), permitiendo que cada doctor atienda múltiples pacientes.

La entidad Appointment (Cita Médica) documenta las consultas entre paciente y doctor, incluyendo la fecha, hora, notas y tipo de cita. Se apoya en catálogos complementarios como Appointment\_Status y Appointment\_Type para mantener la estandarización de estados y modalidades (presencial, virtual, seguimiento, etc.).

Otra parte esencial del modelo son las mediciones médicas y datos de monitoreo. La tabla Vital\_Reading almacena datos como frecuencia cardíaca, presión arterial, saturación de oxígeno, temperatura y pasos, asociados a un dispositivo (Device) y un paciente. Las entidades Device, Device\_Type y Device\_Assignment permiten controlar los dispositivos médicos registrados, su tipo (por ejemplo, smartwatch o monitor de presión), y las asignaciones a pacientes en periodos de tiempo específicos, incluyendo estado de activación.

Asimismo, se integran módulos para localización GPS a través de la entidad Location, la cual registra coordenadas y tiempos, sincronizada también con InfluxDB para análisis de movilidad. El sistema también incorpora la gestión de alertas en tiempo real mediante la entidad Alert, que notifica posibles riesgos médicos. Estas alertas se clasifican mediante catálogos de tipo (Alert\_Type), severidad (Alert\_Severity) y estado (Alert\_Status), ofreciendo control total sobre los eventos clínicos detectados.

A nivel de historial médico, la entidad Medical\_History documenta los antecedentes de cada paciente, mientras que las entidades Patient\_Medical\_Condition, Medication y Allergy registran las condiciones, tratamientos y alergias respectivas. Estas tablas se vinculan con catálogos como Relation\_Type, Condition, y Severity para mantener una estructura estandarizada. Además, el módulo de Consentimiento Médico (Consent) asegura el cumplimiento ético y legal mediante el registro de permisos firmados por los pacientes, con referencias al tipo de consentimiento (Consent\_Scope).

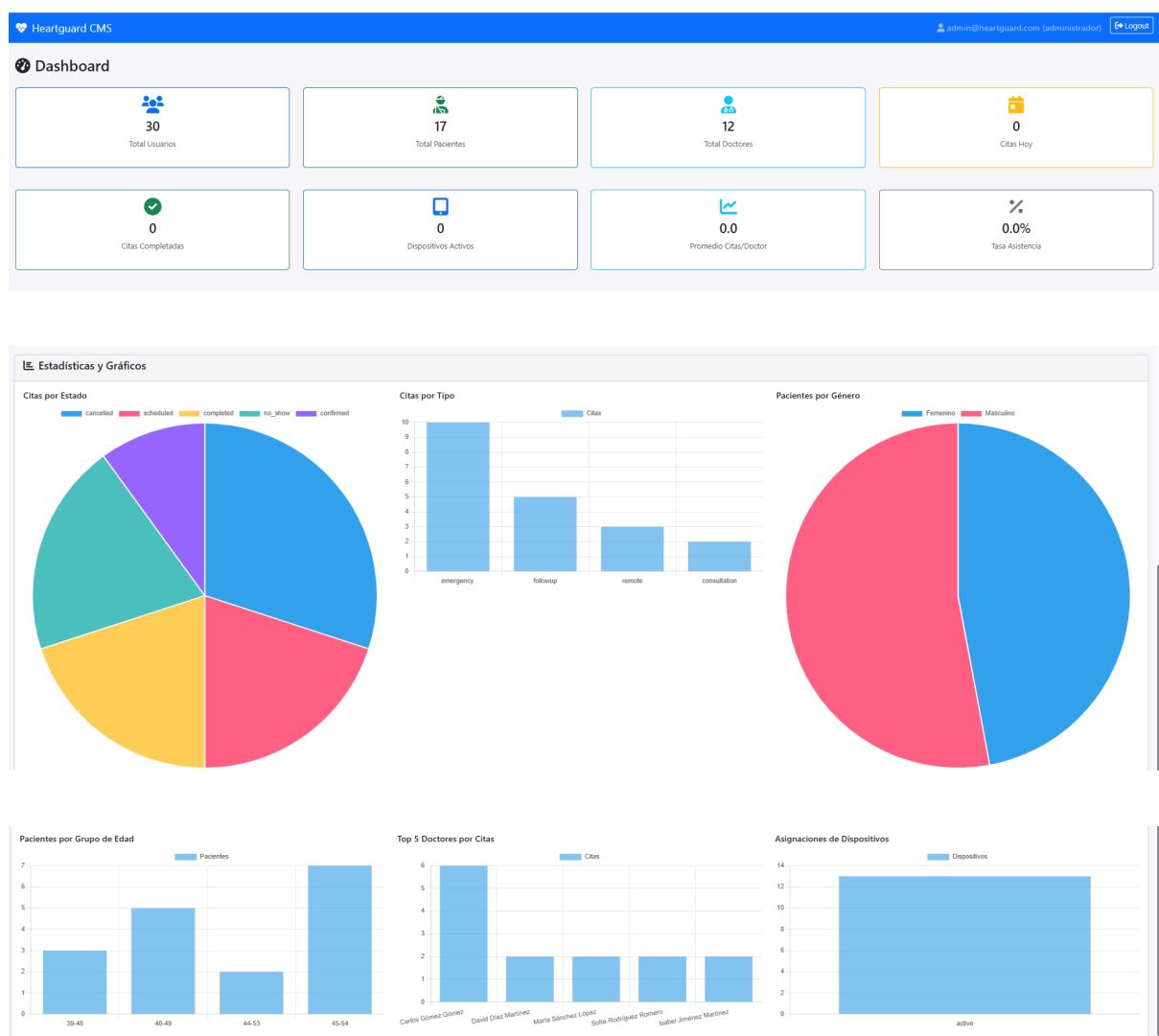
Finalmente, se añaden tablas de apoyo como Gender, Role, Specialty y Assignment\_Status, las cuales garantizan la coherencia de catálogos reutilizables y evitan duplicaciones. El resultado es un modelo robusto, escalable y clínicamente coherente, que refleja la complejidad real del sistema médico HeartGuard, capaz de integrar monitoreo, diagnóstico, trazabilidad y predicción en tiempo real.

En términos de cardinalidades, el nuevo modelo mantiene la consistencia referencial mediante claves primarias y foráneas claramente definidas, eliminando dependencias transitivas y asegurando que cada entidad represente un concepto único y bien delimitado. En conjunto, el rediseño permitió pasar de un modelo centrado en un único flujo de información a un sistema integral, modular y escalable,



alineado con los requisitos de un CMS clínico avanzado con autenticación, trazabilidad y análisis predictivo de riesgo.

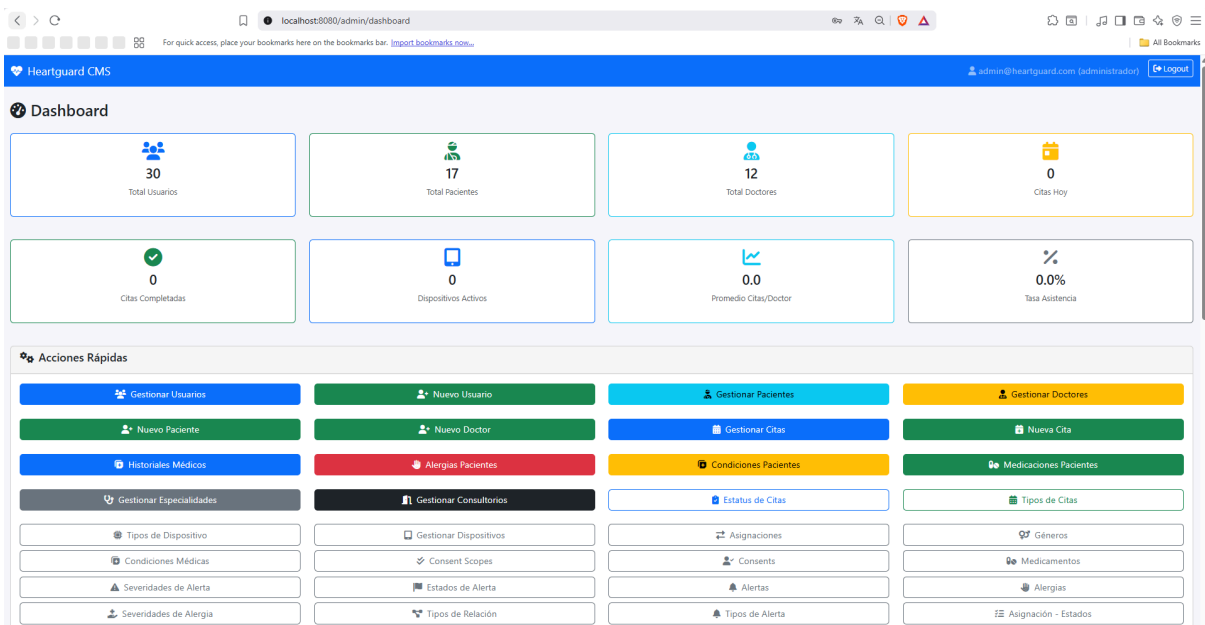
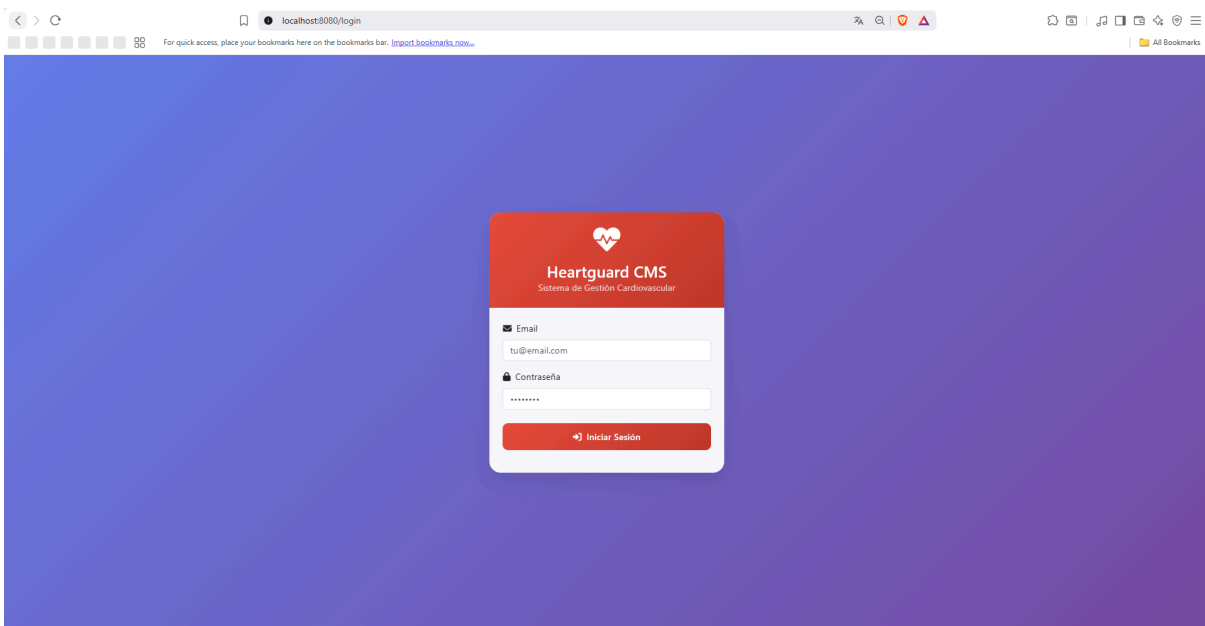
## Presentación de KPIs, métricas y gráficas

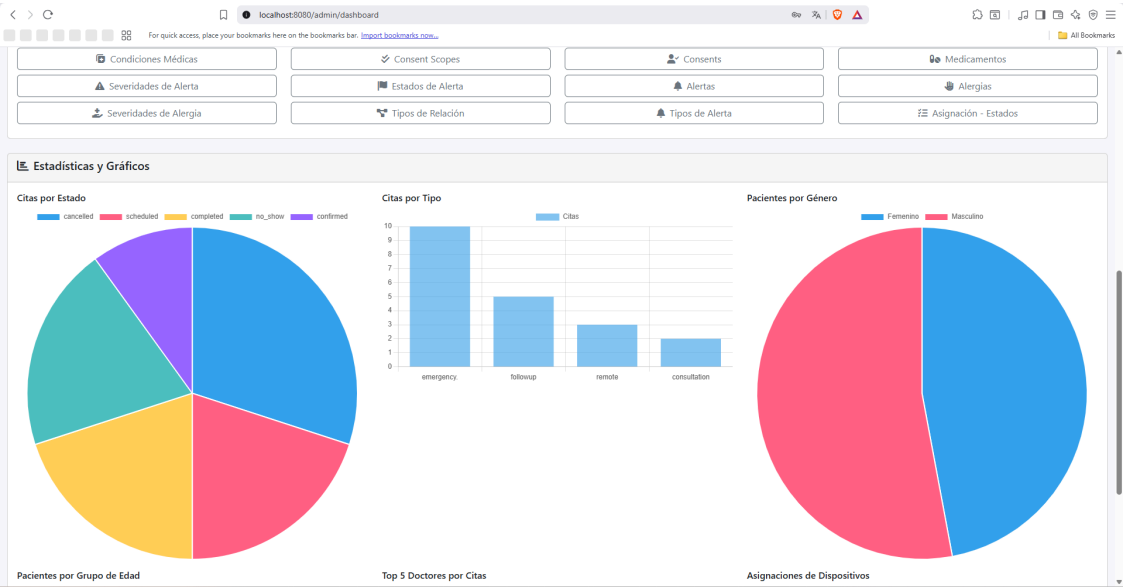


El sistema HeartGuard integra un módulo analítico dentro del CMS que permite visualizar en tiempo real indicadores clave de desempeño (KPIs) mediante seis gráficas interactivas desarrolladas con Chart.js. Estos indicadores reflejan el comportamiento general del sistema, facilitando la toma de decisiones médicas y administrativas. Las métricas implementadas incluyen el número total de pacientes registrados, doctores activos, citas programadas por mes, distribución de pacientes por género, alertas clasificadas por

nivel de riesgo y el top 5 de doctores con mayor número de consultas. Cada una de estas visualizaciones se alimenta directamente desde las funciones SQL y vistas derivadas del sistema, garantizando que los datos mostrados provengan de fuentes actualizadas y verificadas en la base de datos. Este componente analítico fortalece el carácter predictivo de la plataforma, al permitir observar tendencias, detectar anomalías y anticipar escenarios clínicos críticos.

## Capturas de pantalla del sistema en funcionamiento





Gestión de Doctores

Exportar a Excel   Nuevo Doctor

Nombre	Especialidad	Licencia	Consultorio	Fecha Registro	Acciones
Javier Ruiz Rodríguez	Cardiología, Cardiología Intervencionista	DOC002	N/A	10/10/2025	
Juan Martínez Ruiz	Endocrinología, Medicina Interna	DOC005	N/A	10/10/2025	
David Díaz Martínez	Cardiología, Electrofisiología	DOC004	N/A	10/10/2025	
Sofía Rodríguez Romero	Cardiología	DOC007	N/A	10/10/2025	
Carmen Jiménez Rodríguez	Cardiología	DOC011	N/A	10/10/2025	
Isabel Jiménez Martínez	Cardiología	DOC012	N/A	10/10/2025	
Carlos Gómez Gómez	Cardiología, Cardiología Intervencionista	DOC001	N/A	10/10/2025	
María Sánchez López	Cardiología	DOC010	N/A	10/10/2025	
Ana Díaz Díaz	Cardiología	DOC009	N/A	10/10/2025	
Miguel Pérez Rodríguez	Endocrinología, Medicina Interna	DOC006	N/A	10/10/2025	
Luis Romero Hernández	Cardiología, Electrofisiología	DOC003	N/A	10/10/2025	
Laura Hernández Díaz	Cardiología	DOC008	N/A	10/10/2025	

Nuevo Doctor

Nombre

Aldo

Apellido

Peña

Especialidad(es)

☐ Cardiología

☐ Cardiología Intervencionista

☐ Electrofisiología

☐ Endocrinología

☐ Medicina Interna

☐ Neumología

☒ Traumatología

Licencia Médica

DOC6358

Cancelar

Crear Doctor

Editar Doctor

Nombre

Aldo

Apellido

Peña

Especialidad(s)

☐ Cardiología

☐ Cardiología Intervencionista

☐ Electrofisiología

☐ Endocrinología

☐ Medicina Interna

☐ Neumología

☒ Traumatología

Licencia Médica

DOC6358

Cancelar

Actualizar Doctor

Gestión de Doctores

Exportar a Excel

+ Nuevo Doctor

localhost:8080 wants to save

Downloads

Organize

New folder

Home

Gallery

Aldo - Personal

Aldo

Datos adjuntos

Desktop

Documents

Documents

doctores (1).xlsx

Save as type: Microsoft Excel Worksheet (\*.xlsx)

Save

Cancel

Nombre	Especialidad	Licencia Médica	Fecha Registro	Acciones	
Luis Romero Hernández	Cardiología, Electrofisiología	DOC003	N/A	10/10/2025	<div><div></div><div></div><div></div></div>
Javier Ruiz Rodríguez	Cardiología, Cardiología Intervencionista	DOC002	N/A	10/10/2025	<div><div></div><div></div><div></div></div>
Laura Hernández Díaz	Cardiología	DOC008	N/A	10/10/2025	<div><div></div><div></div><div></div></div>

Doctor ID	User ID	First Name	Last Name	Specialty	License	Created At
28b8c3e6-22f2239a	Juan	Martínez	F	Endocrinología	DOC005	2025-10-10T23:42:55-06:00
7b8a7ae5-8e9338ac	David	Díaz	Martí	Cardiología	DOC004	2025-10-10T23:42:55-06:00
83869318-85982c83	Sofía	Rodríguez		Cardiología	DOC007	2025-10-10T23:42:55-06:00
9755017e-10e14e9a	Carmen	Jiménez	R	Cardiología	DOC011	2025-10-10T23:42:55-06:00
56d3729b-46cb789b	Isabel	Jiménez	A	Cardiología	DOC012	2025-10-10T23:42:55-06:00
9ef44612-036fc22f	Carlos	Gómez	G	Cardiología	DOC001	2025-10-10T23:42:55-06:00
bec022f7-95a1ab01	Maria	Sánchez	L	Cardiología	DOC010	2025-10-10T23:42:55-06:00
c52262ab-ee990974	Ana	Díaz		Cardiología	DOC009	2025-10-10T23:42:55-06:00
cd035a15-619ee1f4	Miguel	Pérez	Rod	Endocrinología	DOC006	2025-10-10T23:42:55-06:00
cde58011-ae35cb92	Luis	Romero	H	Cardiología	DOC003	2025-10-10T23:42:55-06:00
26a589b0-f113b199	Javier	Ruiz	Rodr	Cardiología	DOC002	2025-10-10T23:42:55-06:00
e516d0e9-b9b0f089	Laura	Hernández		Cardiología	DOC008	2025-10-10T23:42:55-06:00
39erty8rhi-345110561	Pablo	Celedon		Traumatología	DOC977	2025-10-10T23:42:55-06:00

La interfaz del CMS fue diseñada para ser intuitiva, funcional y adaptable, desarrollada mediante plantillas HTML, CSS y Bootstrap, procesadas por el motor html/template de Go. El diseño prioriza la usabilidad y el acceso rápido a la información, permitiendo a los administradores y doctores navegar entre módulos como Pacientes, Doctores, Citas, Alertas y Reportes desde un panel lateral unificado. Cada sección del sistema incorpora formularios dinámicos y tablas interactivas que permiten realizar operaciones CRUD de forma fluida, mientras que el panel principal centraliza los KPIs y gráficas con una disposición visual limpia y jerarquizada. La interfaz está pensada para ofrecer una experiencia profesional similar a los paneles clínicos modernos, donde el personal médico puede acceder en segundos al estado general del sistema, analizar métricas y gestionar información en tiempo real de manera eficiente.

## Conclusiones

El desarrollo del Módulo 1 representó una etapa fundamental para la consolidación del ecosistema **Heartguard**, estableciendo las bases tecnológicas del sistema de gestión interna.

La arquitectura implementada en **Go + PostgreSQL** permitió obtener un CMS robusto, seguro y de alto rendimiento, con una clara separación entre la lógica de presentación y la lógica de negocio.

El uso de **Stored Procedures** garantizó consistencia en las operaciones y minimizó errores en la manipulación de datos y se comprobó el rendimiento estable del servidor en contenedores Docker.

Entre los principales retos enfrentados destacan:

- La integración de procedimientos almacenados con las consultas de Go mediante el paquete pgx.
- La gestión de sesiones seguras con roles diferenciados.
- La compatibilidad entre las plantillas HTML y los datos obtenidos desde funciones SQL.

Además de retos como la integración del backend Go con PostgreSQL, la conexión simultánea a Redis e InfluxDB para sesiones y métricas, y el diseño de un panel de control intuitivo.

Como lección aprendida, el equipo reforzó la importancia de planificar desde el diseño del modelo ER hasta la lógica de negocio, manteniendo separación de responsabilidades y documentación clara del código.