

UNIVERSIDAD DE MONTERREY



Reporte - Primera Entrega Proyecto Final

MATERIA: Integración de Aplicaciones Computacionales

Eduardo Garza Briceño - 611441

Jorge Enrique Serangeli Andrade - 596711

Juan Carlos Mendoza Castillo - 598701

Maestro:

Raul Morales Salcedo

“Damos nuestra palabra que hemos realizado esta actividad con integridad académica”

Índice

1) Introducción y contexto del proyecto.....	3
1.1) Definición del Alcance.....	4
1.3) Identificación y Validación de Tecnologías.....	5
2) Cronograma.....	6
2.1) Asignación de Roles y Responsabilidades.....	7
Eduardo.....	7
Jorge.....	7
Juan Carlos (Líder).....	7
Colaboración conjunta.....	7
3) Arquitectura del Software.....	8
4) Arquitectura de Hardware.....	9
5) Diseño de datos: diagrama ER y modelo.....	10
Tablas.....	10
Diagrama E-R.....	13
Stored Procedures y Consultas Complejas.....	14
6) GUI: mockups y justificación de diseño.....	16
Admin de admin (superadmin):.....	16
Admin de familia (cabeza de familia):.....	22
Usuario final (móvil):.....	26
Referencias Bibliográficas.....	28

1) Introducción y contexto del proyecto

HeartGuard es una plataforma de monitoreo y alerta temprana de eventos cardiovasculares en tiempo real, diseñada inicialmente para uso personal y familiar. Su objetivo es reducir la ocurrencia de eventos críticos y mejorar la respuesta ante emergencias, integrando en un solo sistema la captura, análisis y alertamiento de datos fisiológicos clave.

Actualmente, los pacientes dependen de consultas esporádicas o de dispositivos aislados que no consolidan ni procesan la información en tiempo real, lo que eleva el riesgo de una atención tardía. HeartGuard busca resolver este problema al ofrecer un ecosistema integrado que conecte señales fisiológicas (frecuencia cardiaca, presión arterial, oxígeno en sangre), patrones de actividad y geolocalización, generando alertas inmediatas y útiles para la prevención.

El repositorio del proyecto se encuentra disponible en:

[GitHub – HeartGuard](#)

1.1) Definición del Alcance

HeartGuard está concebido como un sistema de uso personal y domiciliario, pensado para que un individuo pueda monitorear su salud cardiovascular y, al mismo tiempo, compartir ese monitoreo dentro de un entorno familiar organizado.

Alcance inicial

- Gestión de organizaciones familiares: un usuario administrador (ej. padre/madre) crea la organización y puede agregar, gestionar y monitorear a otros miembros.
- Monitoreo en tiempo real con:
 - Frecuencia cardíaca
 - Presión arterial
 - Patrones de actividad
 - Ubicación GPS
- Infraestructura tecnológica:
 - Postgres como nuestra base de datos principal para la información estructurada y permanente del sistema.
 - InfluxDB para almacenamiento de series de tiempo.
 - Redis para consultas rápidas.
 - Procesamiento con IA para detección de anomalías en tiempo real.
- Alertamiento: notificaciones inmediatas vía app Android tanto al usuario como a sus familiares.
- Visualización: panel que muestre métricas en vivo, evolución histórica y nivel de riesgo por persona.

Funcionalidades opcionales (futuras etapas)

- Integración con dispositivos médicos certificados.
- Ampliación de perfiles de usuario (paciente, familiar, médico).
- Escalamiento a entornos hospitalarios y ambulancias.
- Exportación de reportes personalizados.

1.2) Entregables Principales

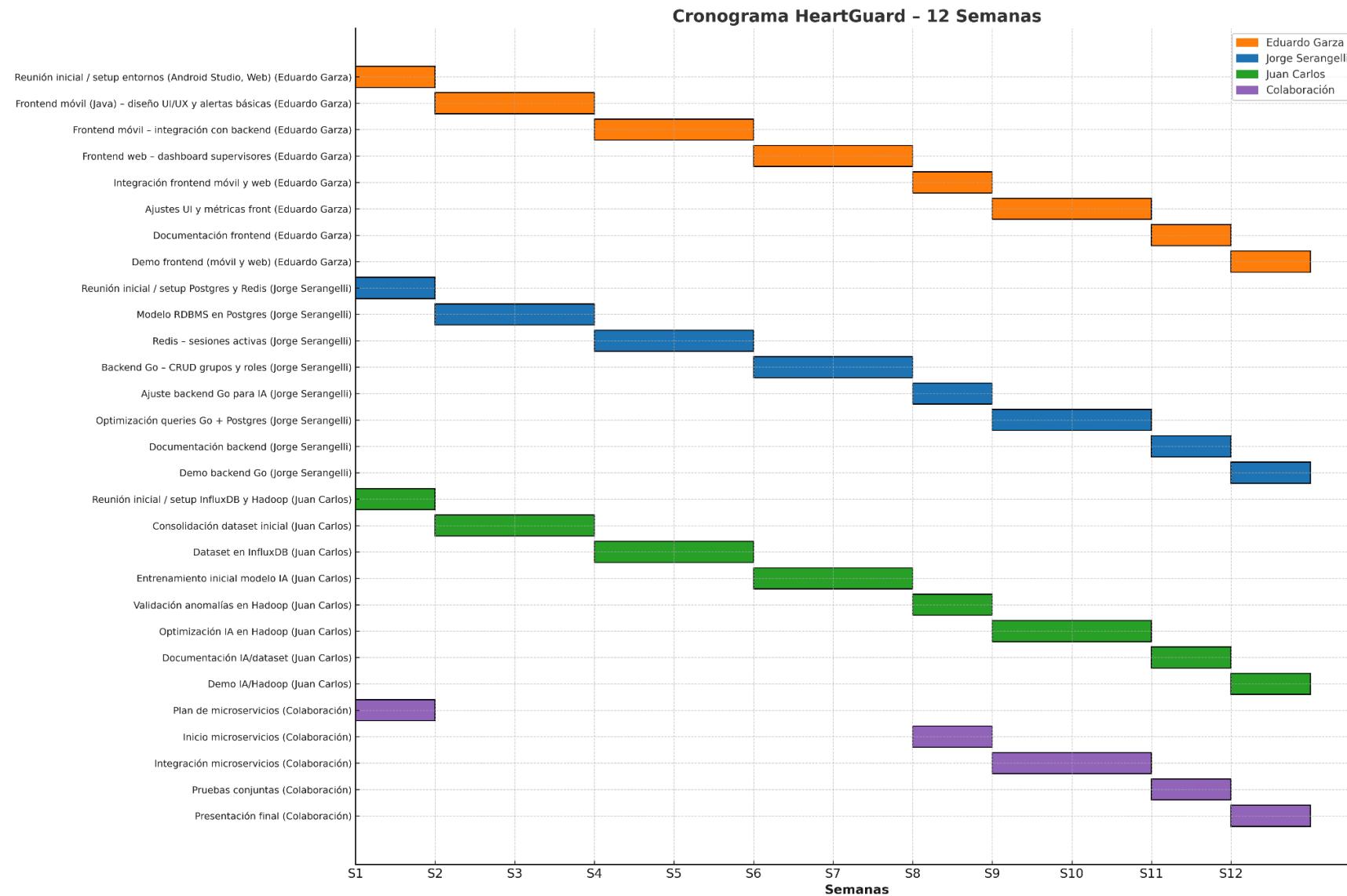
- Documento de comprensión y alcance.
- Cronograma de actividades.

- Prototipo funcional integrado.
- Documentación técnica y reporte final.

1.3) Identificación y Validación de Tecnologías

Tecnología	Propósito	Justificación	Acciones necesarias
Go	Backend para operaciones CRUD (uso exclusivo de administrador)	Lenguaje eficiente y escalable para APIs	Instalar Go, diseñar endpoints REST
Flask (Python)	Microservicios de IA	Flexibilidad para ML y prototipos	Crear entorno virtual, endpoints, etc
InfluxDB	BD de series de tiempo	Ideal para datos continuos	Instalar, configurar inserción de datos
Redis	Caché de alta velocidad	Minimiza latencia en consultas y alertas	Configurar persistencia y pruebas
Hadoop	Procesamiento distribuido	Permite analizar grandes volúmenes de datos	Configuración básica
IA	Creación y entrenamiento de un modelo de IA	Permite analizar qué tipo de alerta se le enviará al usuario	Configurar con app
Android	Aplicación móvil	Amplia compatibilidad con usuarios	Configurar Android Studio y librerías
Leap Motion	Panel de visualización interactivo	Innovación en la interacción de datos	Instalar SDK y pruebas de UI
HTML, CSS y JS	Interfaz web del panel y visualización	Estándar universal, ligero y compatible	Definir estructura, estilos y lógica dinámica

2) Cronograma



2.1) Asignación de Roles y Responsabilidades

Con el fin de lograr un desarrollo equilibrado y organizado del proyecto, se definieron los siguientes roles y responsabilidades para cada integrante del equipo, a continuación se muestran:

Eduardo

- Desarrollo de frontend móvil en Kotlin, exclusivamente para los usuarios para ver sus alertas y recibir alertas a tiempo real.
- Desarrollo de frontend web exclusivamente para aquellos usuarios que tienen una organización o pertenecen a una donde pueden supervisar a los usuarios de bajo nivel (Ejemplo: pacientes) dentro de una organización.

Jorge

- Creación del modelo RDBMS en Postgres para datos pertinentes y la creación de Redis para mantener sesiones activas.
- Desarrollo del backend en Go para operaciones CRUD de administración.

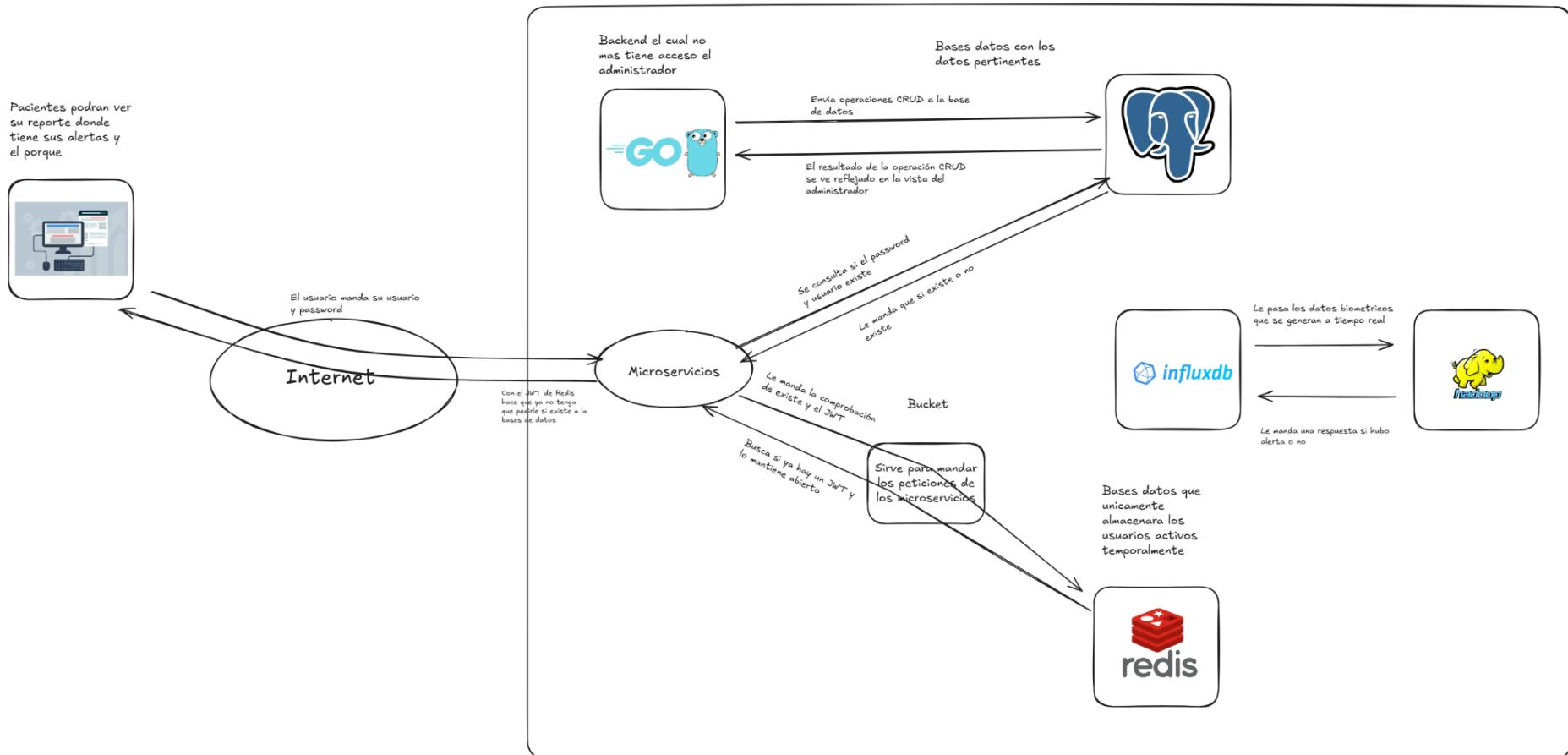
Juan Carlos (Líder)

- Consolidación de dataset para entrenamiento de modelo en donde se pondrá en InfluxDB.
- Crear modelo de IA para la detección de riesgos.
- Desarrollo de frontend móvil en Kotlin.

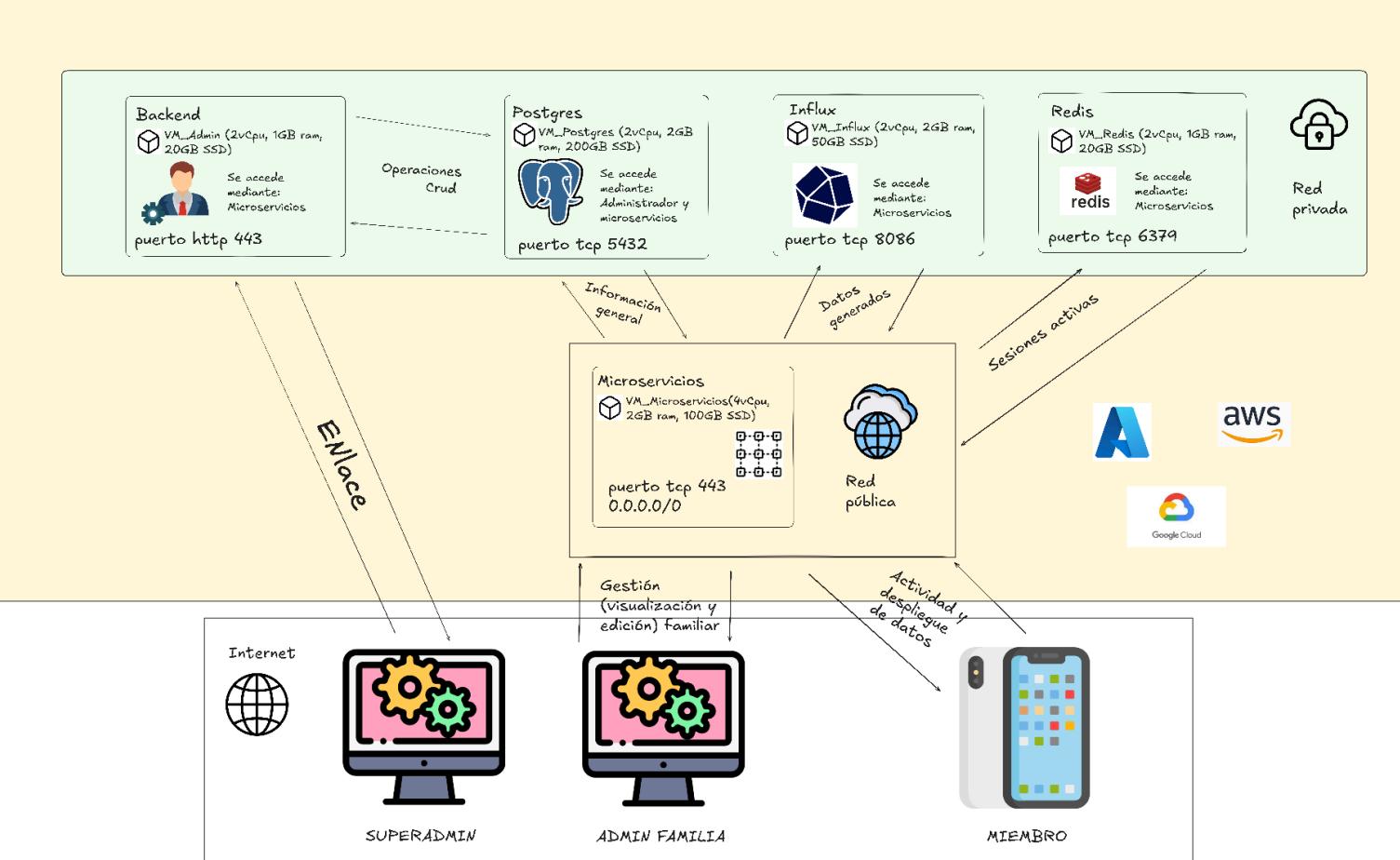
Colaboración conjunta

- Realización de microservicios para integrar los distintos servicios
- Pruebas conjuntas

3) Arquitectura del Software



4) Arquitectura de Hardware



5) Diseño de datos: diagrama ER y modelo

Tablas

1) roles

- Propósito: catálogo de perfiles de acceso.
- PK: id SERIAL.
- Campos clave: nombre (UNIQUE), descripcion, permisos JSONB (feature flags), fecha_creacion.
- Restricciones: nombre NOT NULL y único.
- Índices: idx_usuarios_rol_id (se usa desde usuarios).

2) familias

- Propósito: agrupador lógico de usuarios (núcleos familiares).
- PK: id SERIAL.
- Campos clave: nombre_familia, codigo_familia (UNIQUE) para invitaciones, fecha_creacion.
- Índices: idx_familias_codigo para búsquedas por código.

3) usuarios

- Propósito: identidad de personas autenticables y vinculadas a una familia.
- PK: id SERIAL.
- FK: rol_id → roles(id) (NOT NULL), familia_id → familias(id) (opcional).
- Campos clave: nombre, email (UNIQUE), password_hash, última ubicación (latitud, longitud), ultima_actualizacion, fecha_creacion.
- Restricciones: email UNIQUE, rol_id obligatorio.
- Índices: idx_usuarios_email, idx_usuarios_rol_id, idx_usuarios_familia_id, idx_usuarios_fecha_creacion.

4) alertas

- Propósito: eventos clínicos/operativos generados para usuarios.
- PK: id SERIAL.
- FK: usuario_id → usuarios(id) ON DELETE CASCADE, atendido_por → usuarios(id).

- Campos clave: tipo, descripcion, nivel (CHECK en {bajo, medio, alto, critico}), fecha, atendida, fecha_atencion, geolocalización (latitud, longitud).
- Restricciones: tipo y nivel NOT NULL, validación de nivel por CHECK.
- Índices: idx_alertas_usuario_id, idx_alertas_fecha DESC, parciales idx_alertas_atendida (solo no atendidas) y idx_alertas_nivel (alto/critico).

5) catalogos

- Propósito: catálogo genérico (tipos de alerta, niveles, estados...).
- PK: id SERIAL.
- Campos clave: tipo, clave, valor, descripcion, activo, fecha_creacion.
- Restricciones: UNIQUE(tipo, clave); tipo, clave, valor NOT NULL.
- Índices: idx_catalogos_tipo, idx_catalogos_tipo_clave.

6) logs_sistema

- Propósito: auditoría de acciones del sistema/usuarios.
- PK: id SERIAL.
- FK: usuario_id → usuarios(id) (nullable para eventos del sistema).
- Campos clave: accion, detalle JSONB/TEXT, fecha.
- Índices: idx_logs_usuario_id, idx_logs_fecha DESC, idx_logs_accion.

7) microservicios

- Propósito: registro de endpoints backend y su estado.
- PK: id SERIAL.
- Campos clave: nombre, url, estado (CHECK en {activo, inactivo, error}), ultima_verificacion, version, descripcion, fecha_creacion.
- Índices: idx_microservicios_estado, idx_microservicios_nombre.

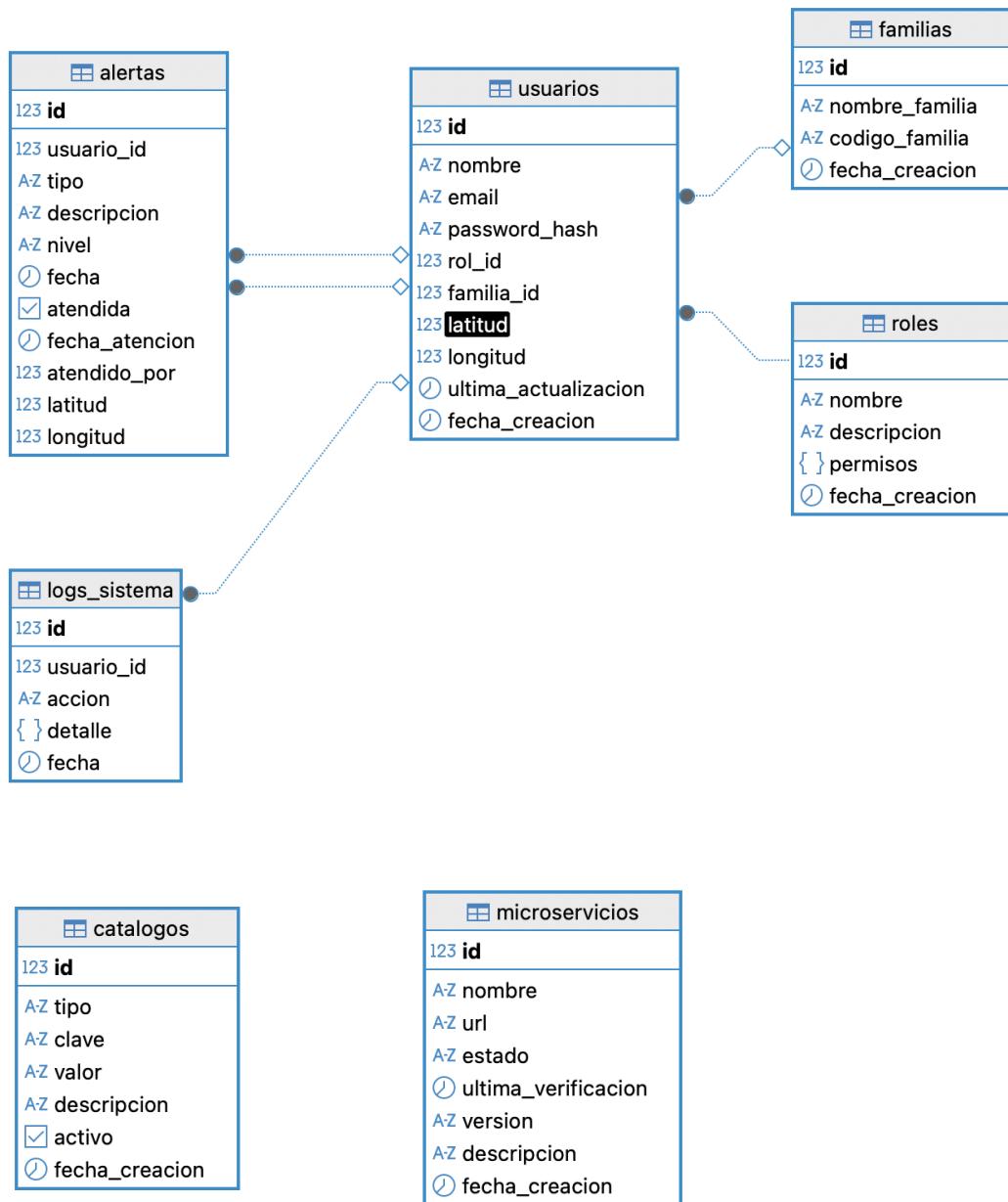
Vistas (KPIs)

- v_usuarios_completos: une usuarios + roles + familias para lecturas rápidas de perfil.
- v_alertas_completas: une alertas con su usuario, familia y el operador que atendió.
- v_estadisticas_familias: métricas por familia (miembros y admins).

Relaciones (cardinalidad)

- roles (1) —< (N) usuarios
- familias (1) —< (N) usuarios
- usuarios (1) —< (N) alertas (al eliminar usuario, se borran sus alertas).
- usuarios (1) —< (N) logs_sistema (opcional: eventos del sistema sin usuario).
- usuarios (1) —< (N) alertas.atendido_por (self-FK opcional).
- catalogos: independiente (apoya consistencia referencial lógica vía app).
- microservicios: independiente (monitoreo/health).

Diagrama E-R



Stored Procedures y Consultas Complejas

A continuación se documentan algunos de los procedimientos almacenados (SPs) que cumplen los criterios de la primera entrega. Para consultar el listado completo y su detalle técnico, revisa el archivo backend/init.sql del repositorio HeartGuard.

sp_dashboard_ejecutivo() → KPIs globales

Qué hace: regresa totales de usuarios (excluye superadmin), familias, alertas pendientes/ críticas y un fijo de microservicios activos (4).

Ejemplo:

SQL

```
SELECT * FROM sp_dashboard_ejecutivo();
```

sp_get_usuarios(p_límite, p_offset, p_rol_id, p_familia_id) → Listado de usuarios con rol y familia

Qué hace: devuelve usuarios con su rol y, si aplica, el nombre de la familia; permite filtrar por rol y/o familia; excluye superadmin.

Ejemplo:

SQL

```
SELECT * FROM sp_get_usuarios(50, 0, NULL, NULL);
```

sp_get_alertas() → Alertas con datos del usuario y familia

Qué hace: lista alertas uniendo usuario y familia, ordenadas por fecha desc; incluye nivel, estatus de atención y geolocalización.

Ejemplo:

SQL

```
SELECT * FROM sp_get_alertas();
```

sp_get_logs_sistema(p_límite, p_offset, p_usuario_id, p_accion) → Bitácora con filtros básicos

Qué hace: consulta logs con JOIN a usuarios; filtra por usuario y patrón en “acción” (ILIKE); ordena por fecha desc con paginación.

Ejemplo:

SQL

```
SELECT * FROM sp_get_familias(50, 0);
```

sp_get_dashboard_stats() → Resumen ejecutivo rápido

Qué hace: devuelve totales de usuarios (excluye rol_id=1), familias, alertas y pendientes; replica conteos simples para “activos”.

Ejemplo:

SQL

```
SELECT * FROM sp_get_dashboard_stats();
```

sp_get_familias(p_límite, p_offset) → Familias con métricas de membresía

Qué hace: retorna familias con totales de miembros y de admins (conteos agregados), ordenadas por fecha de creación desc.

Ejemplo:

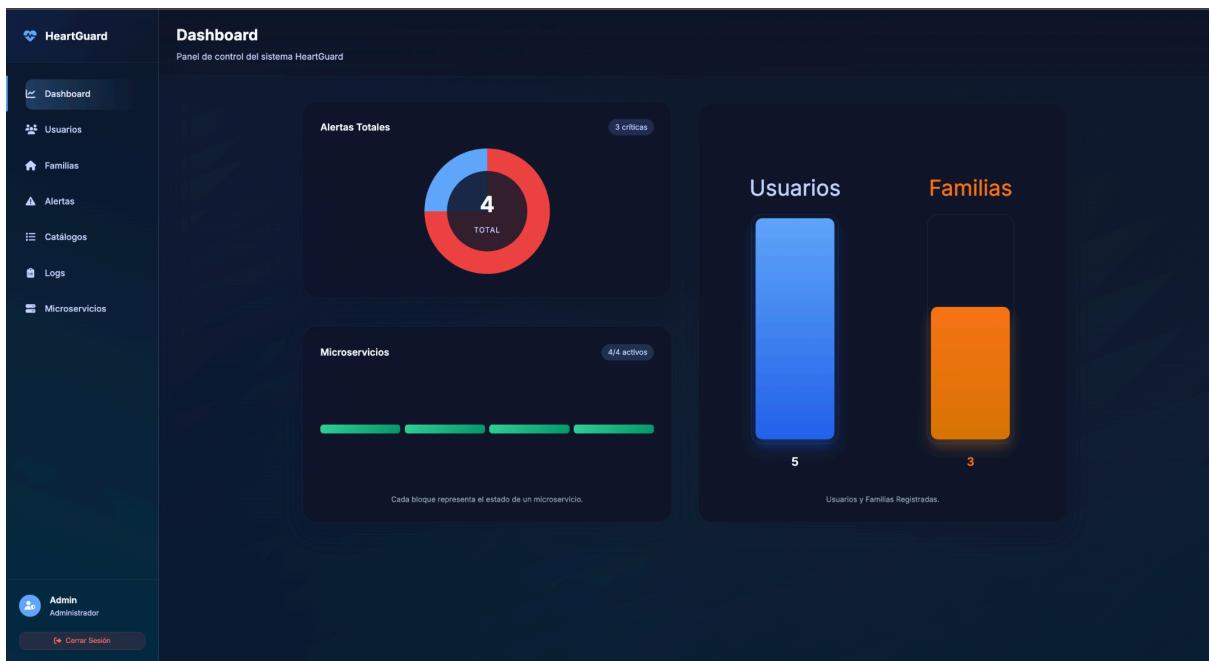
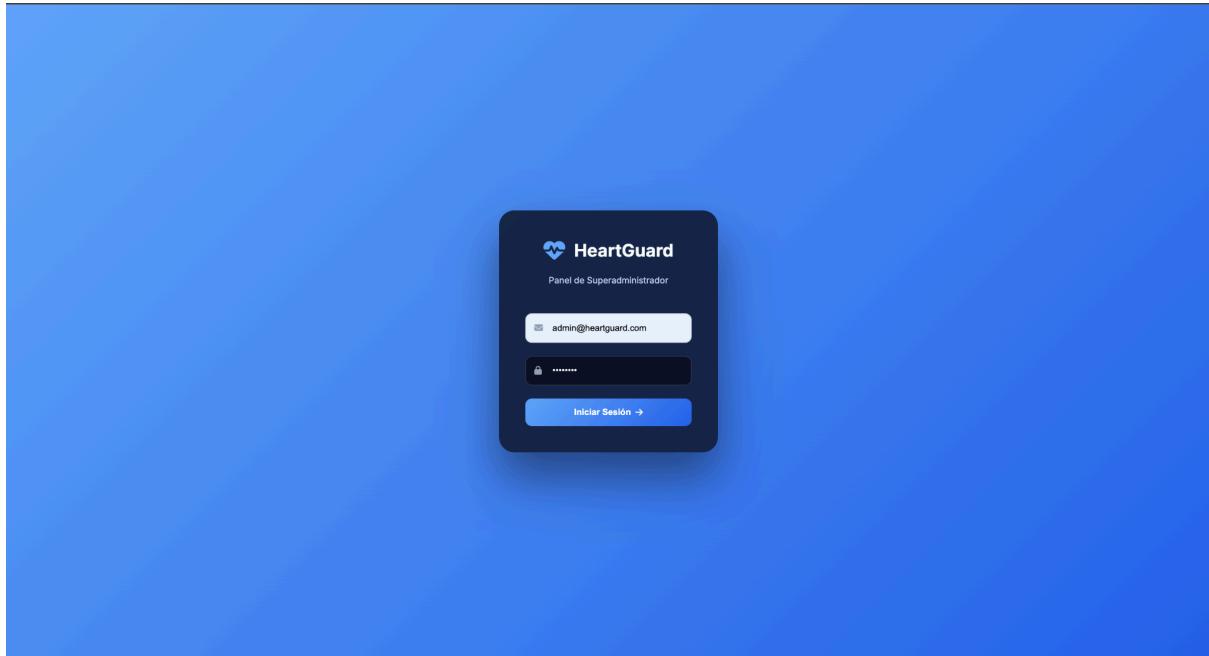
SQL

```
SELECT * FROM sp_get_familias(50, 0);
```

6) GUI: mockups y justificación de diseño

Admin de admin (superadmin):

Dashboard ejecutivo: conteos (usuarios, familias, alertas), estado de microservicios, logs, gestión de familias/usuarios y catálogos.



HeartGuard

Dashboard

Panel de control del sistema HeartGuard

Dashboard Usuarios Familias Alertas Catálogos Logs Microservicios Admin Administrador Cerrar Sesión

Gestión de Usuarios

Buscar usuario... Nuevo Usuario

ID	NOMBRE	EMAIL	ROL	FAMILIA	ACCIONES
4	Carlos Rodríguez	carlos.rodriguez@email.com	admin_familia	Familia López	Editar Eliminar
2	Juan García	juan.garcia@email.com	admin_familia	Familia Rodríguez	Editar Eliminar
5	Ana López	ana.lopez@email.com	miembro	Familia López	Editar Eliminar
3	Maria García	maria.garcia@email.com	miembro	Familia Rodríguez	Editar Eliminar

HeartGuard

Dashboard

Panel de control del sistema HeartGuard

Dashboard Usuarios Familias Alertas Catálogos Logs Microservicios Admin Administrador Cerrar Sesión

Gestión de Familias

Buscar familia... Nuevo Familia

ID	NOMBRE	MIEMBROS	FECHA CREACIÓN	ACCIONES
2	Familia Rodríguez	2	9/18/2025	Editar Eliminar
1	Familia García	0	9/18/2025	Editar Eliminar
3	Familia López	2	9/18/2025	Editar Eliminar

HeartGuard

Panel de control del sistema HeartGuard

Gestión de Alertas

Buscar alerta...

ID	USUARIO	TIPO	DESCRIPCIÓN	NIVEL	FECHA	LATITUD	LONGITUD
1	Maria García	frecuencia_cardiaca	Frecuencia cardíaca elevada detectada: 120 bpm	alto	9/18/2025	19.4326	-99.1332
4	Juan García	frecuencia_cardiaca	Frecuencia cardíaca irregular detectada	alto	9/18/2025	25.6866	-100.3161
2	Carlos Rodríguez	presion_arterial	Presión arterial alta: 150/95 mmHg	critico	9/18/2025	25.6866	-100.3161
3	Ana López	oxigenacion	Saturación de oxígeno baja: 88%	medio	9/18/2025	19.4326	-99.1332

Admin
Administrador

Cerrar Sesión

HeartGuard

Panel de control del sistema HeartGuard

Catálogos del Sistema

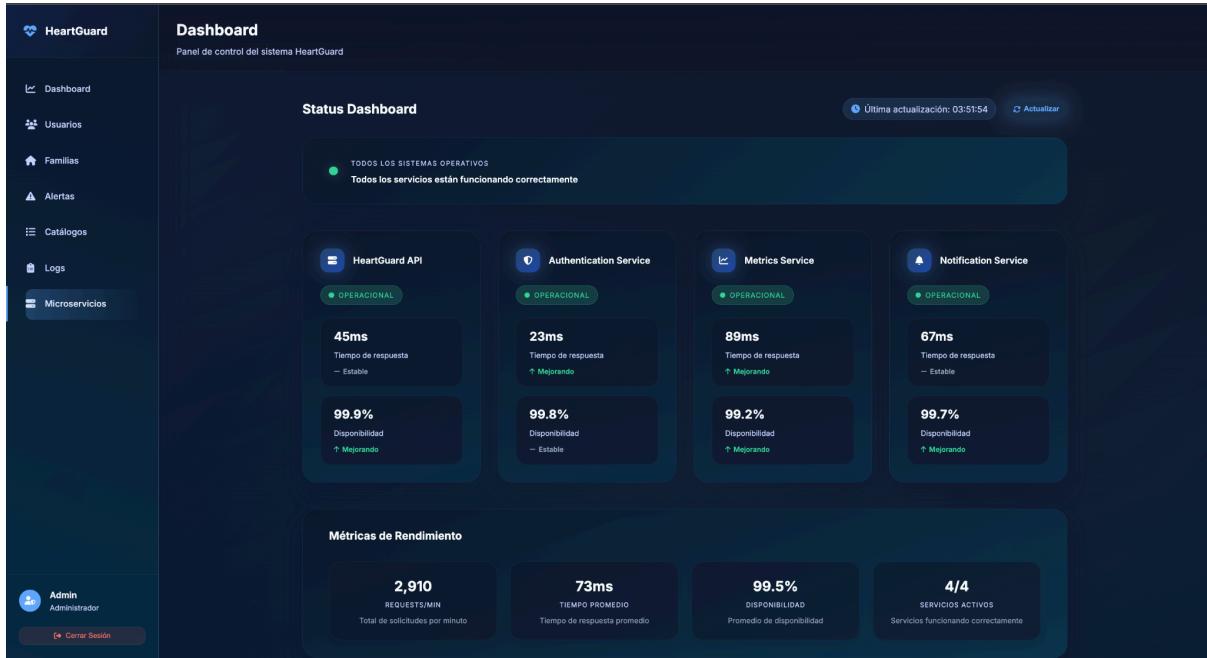
+ Nuevo Catálogo

ID	TIPO	CLAVE	VALOR	ACCIONES
12	estado_sistema	activo	Activo	<button>Editar</button> <button>Eliminar</button>
14	estado_sistema	inactivo	Inactivo	<button>Editar</button> <button>Eliminar</button>
13	estado_sistema	mantenimiento	Mantenimiento	<button>Editar</button> <button>Eliminar</button>
7	nivel_alerta	alto	Alto	<button>Editar</button> <button>Eliminar</button>
9	nivel_alerta	bajo	Bajo	<button>Editar</button> <button>Eliminar</button>
6	nivel_alerta	critico	Criticó	<button>Editar</button> <button>Eliminar</button>
8	nivel_alerta	medio	Medio	<button>Editar</button> <button>Eliminar</button>
4	tipo_alerta	frecuencia_alta	Frecuencia Alta	<button>Editar</button> <button>Eliminar</button>
5	tipo_alerta	frecuencia_baja	Frecuencia Baja	<button>Editar</button> <button>Eliminar</button>
2	tipo_alerta	presion_alta	Presión Alta	<button>Editar</button> <button>Eliminar</button>
3	tipo_alerta	presion_baja	Presión Baja	<button>Editar</button> <button>Eliminar</button>
1	tipo_alerta	ritmo_irregular	Ritmo Irregular	<button>Editar</button> <button>Eliminar</button>
11	tipo_usuario	admin	Administrador	<button>Editar</button> <button>Eliminar</button>
10	tipo_usuario	superadmin	Super Administrador	<button>Editar</button> <button>Eliminar</button>

Admin
Administrador

Cerrar Sesión

localhost:8080/#



Justificación del diseño de la interfaz de superadmin:

El panel de superadministrador está concebido como un centro de comando centralizado, diseñado para ofrecer al personal técnico y administrativo una herramienta eficiente, clara y potente para la gestión y monitorización integral del ecosistema HeartGuard.

El objetivo principal no es replicar la experiencia del usuario final, sino proporcionar una visibilidad completa sobre la salud del sistema, la actividad de los usuarios y la configuración de la plataforma. Cada decisión de diseño se ha tomado con tres principios en mente:

Claridad Inmediata: Presentar los datos más críticos de un vistazo.

Gestión Enfocada: Facilitar las tareas administrativas esenciales sin distracciones.

Diagnóstico Rápido: Permitir la identificación y el análisis de problemas de forma intuitiva.

La pantalla de inicio o Dashboard principal está diseñada para responder a tres preguntas fundamentales de un administrador en menos de 10 segundos:

¿Hay una emergencia? (Alertas Totales): Este es el componente más crítico. Muestra de forma prominente el número total de alertas y, más importante aún, cuántas son críticas. Su diseño circular y llamativo lo convierte en el principal foco de atención, permitiendo una reacción inmediata ante problemas de los usuarios.

¿Cómo está creciendo la plataforma? (Usuarios y Familias): Estas dos barras representan las métricas clave de crecimiento y adopción. Un administrador puede ver de un vistazo el volumen de usuarios y la estructura de las familias, entendiendo así la escala actual de la plataforma.

¿Está la plataforma funcionando correctamente? (Microservicios): Este componente muestra la salud técnica del sistema. Informa de manera visual e inmediata si la infraestructura subyacente está operativa, permitiendo un diagnóstico rápido de problemas técnicos.

Las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) se han limitado intencionadamente a las entidades maestras del sistema: Usuarios y Familias.

Justificación: El rol del superadministrador es gestionar la estructura de la plataforma, no generar los datos transaccionales. Los administradores necesitan poder crear cuentas, asignar roles, modificar datos de usuarios y gestionar los grupos familiares. Estas son las bases sobre las que operan los usuarios finales.

Exclusión de CRUD para Alertas y Logs: Las alertas y los logs son registros generados por la actividad del sistema y de los usuarios. No tendría sentido que un administrador los "creara" manualmente. Para estas secciones, el rol del administrador es de supervisión, consulta y auditoría, no de creación. El diseño refleja esto al presentar los datos en tablas de solo lectura.

Por ejemplo, la sección de microservicios va más allá del resumen del dashboard. Proporciona una vista detallada del estado de cada componente de la arquitectura. El objetivo es permitir un diagnóstico proactivo, donde el administrador pueda identificar qué servicio específico está fallando o presentando degradación del rendimiento, antes de que se convierta en un problema mayor.

Logs por otro lado es una herramienta de trazabilidad y auditoría por excelencia. Permite responder a preguntas críticas como: "¿Quién hizo qué y cuándo?". Es

fundamental para la seguridad, para depurar errores complejos y para tener un registro histórico de todas las acciones importantes que ocurren en el sistema.

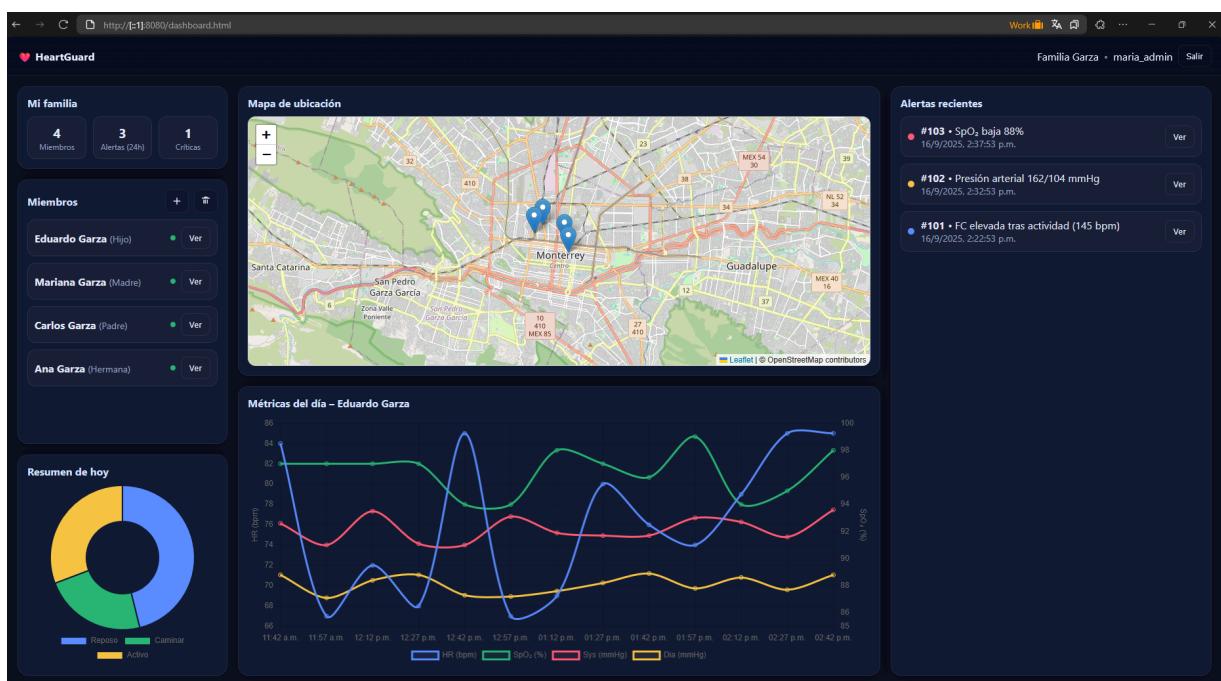
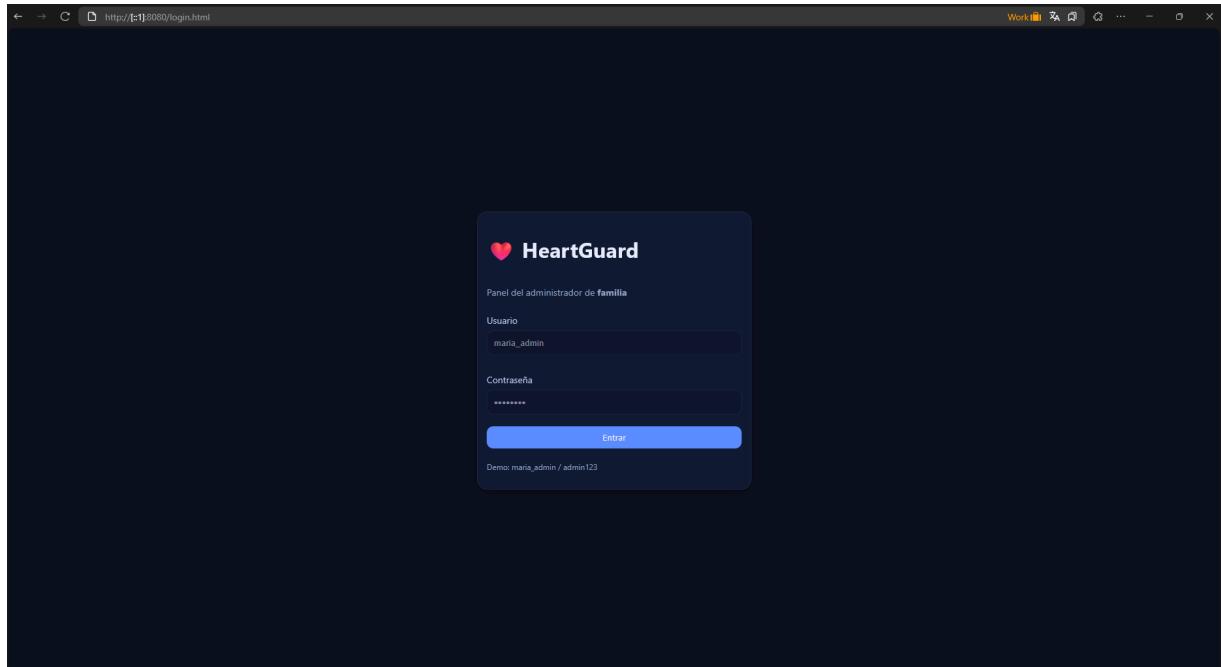
Aunque no es una sección de CRUD tradicional, la gestión de Catálogos es una de las funciones más potentes para un superadministrador.

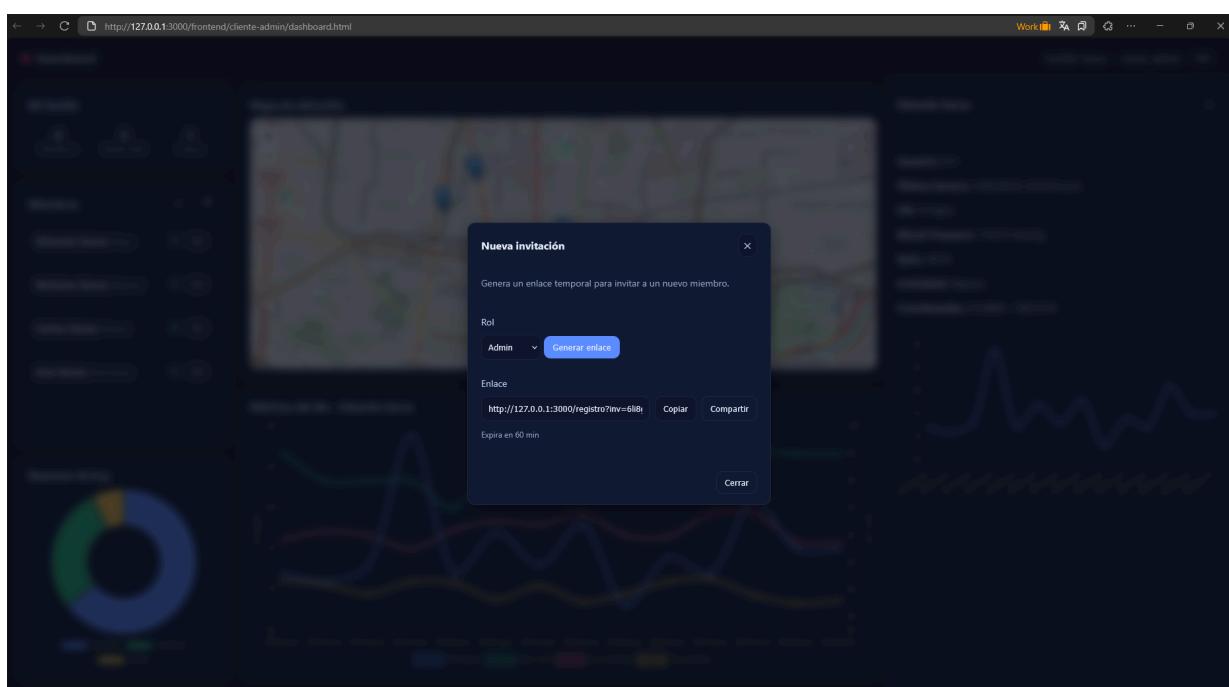
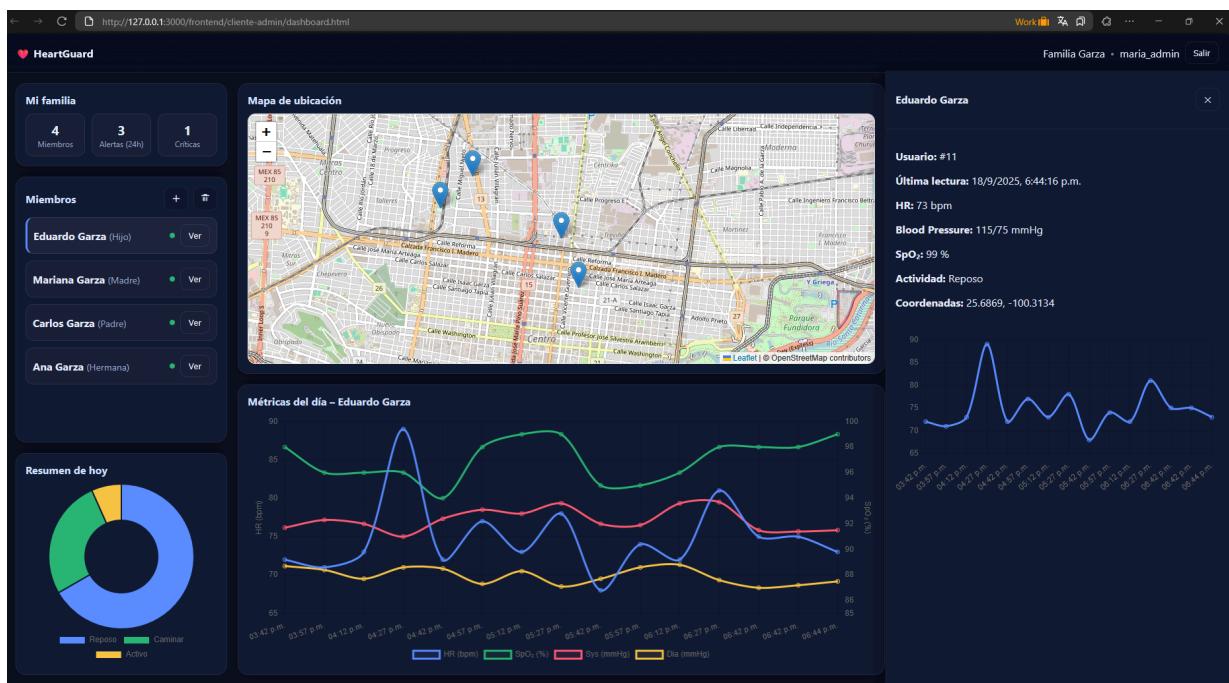
Justificación: Esta sección permite modificar el comportamiento del sistema sin necesidad de cambiar el código fuente. Por ejemplo, se pueden añadir nuevos niveles de alerta, definir nuevos roles de usuario o cambiar tipos de notificaciones. Esto dota a la plataforma de una enorme flexibilidad y escalabilidad, permitiendo que se adapte a futuras necesidades de negocio de forma rápida y segura.

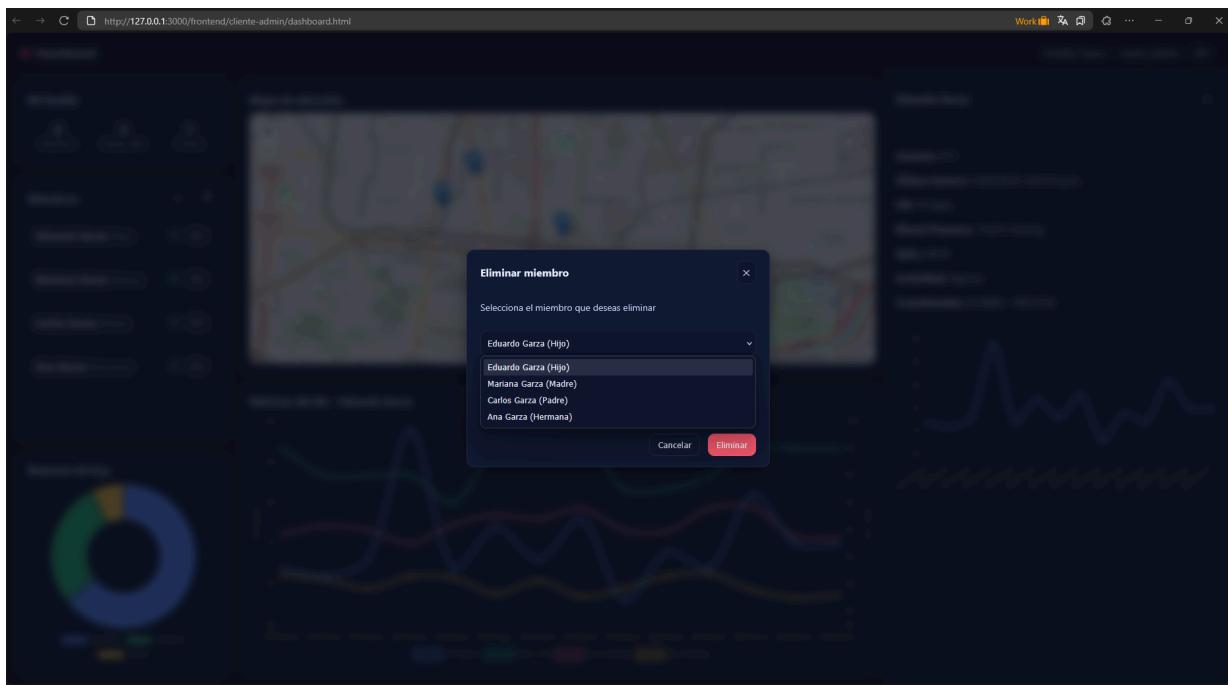
En resumen, el diseño del panel está justificado por una clara separación de responsabilidades: el administrador gestiona la estructura (Usuarios, Familias, Catálogos) y monitoriza la actividad y la salud del sistema (Dashboard, Alertas, Logs, Microservicios). Este enfoque garantiza que el panel sea una herramienta de control eficiente y no una simple réplica de las funciones del usuario final.

Admin de familia (cabeza de familia):

Panel de su familia: agregar y eliminar miembros, alertas recientes, métricas resumidas del día, contactos de emergencia y localización en tiempo real.







Justificación del diseño de la interfaz

La interfaz de HeartGuard busca que el administrador de familia pueda supervisar la salud de sus seres queridos de manera sencilla y rápida. El diseño parte de una pantalla de acceso clara y directa, que garantiza privacidad y transmite confianza antes de entrar al panel principal.

Una vez dentro, el dashboard está organizado para que la información más importante aparezca primero. En un vistazo se pueden ver cuántos miembros hay en la familia, cuántas alertas se han registrado y cuáles son críticas. La lista de miembros y el mapa refuerzan el enfoque familiar, mostrando tanto a las personas como su ubicación en tiempo real.

Las gráficas y los indicadores están pensados para que cualquier usuario, sin necesidad de conocimientos médicos o técnicos, pueda identificar cambios relevantes en signos vitales como la frecuencia cardíaca, la presión arterial o la oxigenación en sangre. Los colores y etiquetas permiten distinguir rápidamente entre estados normales y situaciones de riesgo.

El estilo visual se mantuvo oscuro y sobrio, con acentos de color para resaltar lo urgente. Esto no solo mejora la legibilidad, sino que también transmite seriedad y foco en la seguridad. Además, la interfaz se adapta a distintos tamaños de pantalla,

de modo que pueda usarse cómodamente en computadora o en un dispositivo móvil.

En resumen, el diseño de la interfaz convierte datos complejos en una experiencia clara, accesible y centrada en la familia, cumpliendo con la finalidad del proyecto: facilitar la prevención y reacción temprana ante emergencias cardiovasculares.

Usuario final (móvil):

Última Lectura

HR 110 bpm • SpO₂ 98%
PA 120/80 mmHg • T 36.5°C
2023-10-27T10:00:00Z

Alertas

Heart Rate Alert
Ritmo cardiaco alto

Ver Historial

Última Lectura

HR 90 bpm • SpO₂ 99% • PA 122/81 mmHg • T 36.8°C
2023-10-27T09:45:00Z

Última Lectura

HR 88 bpm • SpO₂ 97% • PA 118/79 mmHg • T 36.6°C
2023-10-27T09:30:00Z

Última Lectura

HR 92 bpm • SpO₂ 98% • PA 125/82 mmHg • T 37.0°C
2023-10-27T09:15:00Z

Última Lectura

HR 86 bpm • SpO₂ 99% • PA 121/80 mmHg • T 36.7°C
2023-10-27T09:00:00Z

Justificación del diseño de la interfaz

Pantalla de Inicio de Sesión:

La pantalla de inicio de sesión está diseñada para ser minimalista y directa, con solo dos campos de texto (Email y Password) y un botón claro (Entrar), la interfaz minimiza las distracciones y reduce la posibilidad de errores, ofreciendo una experiencia de usuario sin fricciones desde el primer momento.

Pantalla de Lectura Actual:

Esta pantalla es el centro de atención de la aplicación, priorizando la información más relevante para el usuario. El diseño jerárquico presenta la "Última Lectura" de forma destacada, mostrando los signos vitales clave de un vistazo. La sección de "Alertas" es crucial; su diseño con un color rojo llamativo y un ícono de advertencia asegura que cualquier anomalía, como un ritmo cardíaco alto, sea notificada al usuario de manera inmediata y efectiva.

Pantalla de Historial:

El historial de lecturas está organizado de manera clara y cronológica para facilitar el seguimiento de la salud a largo plazo. Cada lectura individual se muestra en una tarjeta separada que contiene toda la información vital (HR, SpO2, PA) junto con su fecha y hora exactas. Este formato de tarjetas es fácil de escanear y permite a los usuarios revisar su progreso o identificar patrones a lo largo del tiempo sin esfuerzo.

Referencias Bibliográficas

1. InfluxData. (2024). *InfluxDB Documentation*. Disponible en:
<https://docs.influxdata.com/influxdb>
2. Redis Labs. (2024). *Redis Documentation*. Disponible en:
<https://redis.io/documentation>
3. Android Developers. (2024). *Developer Guides*. Disponible en:
<https://developer.android.com/docs>
4. Leap Motion. (2020). *Leap Motion Developer Documentation*. Disponible en:
<https://developer.leapmotion.com/>