UNIVERSIDAD DE MONTERREY

Reporte - Avance 2 - Proyecto Final

MATERIA: Integración de Aplicaciones Computacionales

Eduardo Garza Briceño - 611441

Jorge Enrique Serangelli Andrade - 596711

Juan Carlos Mendoza Castillo - 598701

Maestro:

Raul Morales Salcedo

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

1. Objetivo del Módulo 1 (Back End CMS)

El objetivo principal de este módulo es establecer la fundación completa del backend y la base de datos para la aplicación HeartGard. Esto incluye la creación de un sistema de gestión de contenido (CMS) robusto y una interfaz de administración que permita la gestión integral de todos los aspectos de la plataforma. El módulo busca centralizar la lógica de negocio en la base de datos a través de Stored Procedures para garantizar la integridad, seguridad y consistencia de los datos. Además, se enfoca en proveer las herramientas necesarias para el monitoreo operativo, la gestión de usuarios multi-tenant, el dominio clínico (pacientes y alertas), y la administración de contenido editorial, sentando las bases para la escalabilidad futura de la aplicación.

2. Descripción Detallada de las Funcionalidades

El CMS de HeartGuard cuenta con una interfaz web robusta que permite a los administradores gestionar todas las facetas del ecosistema. A continuación, se describen las funcionalidades implementadas, organizadas por módulo:

Gestión de Entidades Principales

- **Dashboard:** Pantalla principal que ofrece una vista general del estado del sistema, incluyendo métricas clave y accesos directos a las funciones más importantes.
- **Organizaciones:** Permite crear, visualizar, actualizar y eliminar organizaciones (hospitales, clínicas) que utilizan el sistema.
- **Pacientes:** Gestión completa del ciclo de vida de los pacientes, incluyendo su registro, asignación a organizaciones, consulta de datos demográficos y estado actual.
- **Cuidadores (Caregivers):** Administración de los profesionales de la salud o cuidadores, incluyendo su creación, asignación a pacientes y gestión de roles.
- **Usuarios (Users):** Gestión de todas las cuentas de usuario del sistema, incluyendo administradores y cuidadores, con control sobre sus permisos y acceso.

Gestión de Dispositivos y Datos

- **Dispositivos (Devices):** Módulo para registrar y gestionar los dispositivos de monitoreo (wearables), asociarlos a pacientes y verificar su estado (activo, inactivo).
- **Flujos de Señales (Signal Streams):** Permite supervisar los flujos de datos biométricos que envían los dispositivos en tiempo real.
- **Alertas (Alerts):** Visualización y gestión de las alertas generadas por el sistema cuando se detectan anomalías en los datos de un paciente.
- **Inferencias (Inferencias):** Consulta de las predicciones o inferencias generadas por los modelos de machine learning a partir de los datos de los pacientes.

- **Ground Truth:** Módulo para etiquetar datos y validar las inferencias del sistema, crucial para el reentrenamiento y la mejora de los modelos de IA.

Configuración y Administración del Sistema

- **Roles:** Creación y asignación de roles de usuario con permisos específicos para garantizar la seguridad y el principio de mínimo privilegio.
- **Tipos de Eventos (Event Types):** Catálogo para definir y gestionar los tipos de eventos que el sistema puede registrar (ej. "caída detectada", "ritmo cardíaco anómalo").
- **Claves de API (API Keys):** Gestión de claves de API para la integración segura con servicios de terceros o aplicaciones móviles.
- **Modelos (Models):** Administración de las versiones de los modelos de machine learning que se utilizan en la plataforma.
- **Auditoría (Audit):** Registro detallado de todas las acciones importantes realizadas en el sistema (quién, qué, cuándo), fundamental para la seguridad y el cumplimiento normativo.
- **Exportaciones por Lote (Batch Exports):** Funcionalidad para generar y descargar grandes volúmenes de datos para análisis externos o copias de seguridad.
- **Gestión de Contenido (Content Blocks):** CMS interno para administrar el contenido estático o informativo que se muestra en las aplicaciones cliente (móvil/web).

4. RDBMS Nuevo vs RBDMS Antiguo

4.1 El antiguo RBDMS

RBMS estaba pensado para un alcance familiar donde un supervisor (papá, mamá, etc…) podía gestionar a los miembros familiares para ver sus síntomas en todo momento.

Diagrama del antiguo RDBMS:

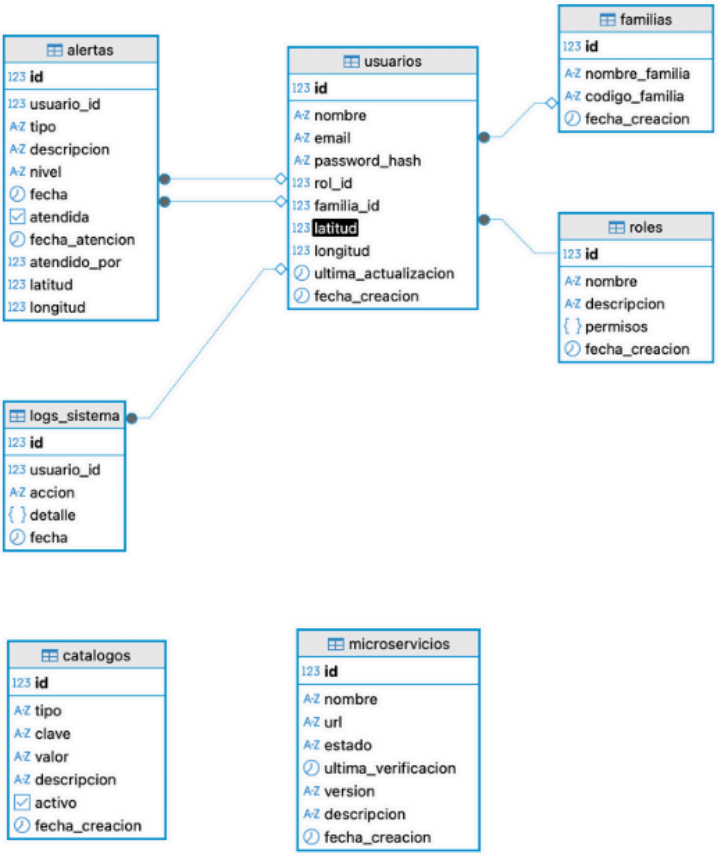


Figura 1: Diagrama RDBMS nuevo de Heartguard.

4.2 El nuevo RBDMS

El nuevo RDBMS ahora está pensado para cualquier tipo de uso donde definimos el alcance de manera abstracta para que pueda ser implementado ya sea para hospitales, centros de organizaciones, escuelas, familias, etc. Y implementamos un modelo normalizado 3FN debido a que minimiza la redundancia de datos, lo que es crucial para garantizar la integridad y consistencia de las transacciones. Este proceso asegura que cada dato se almacena una sola vez y que, al actualizarse, los cambios se propaguen correctamente

Diagrama del nuevo RDBMS:

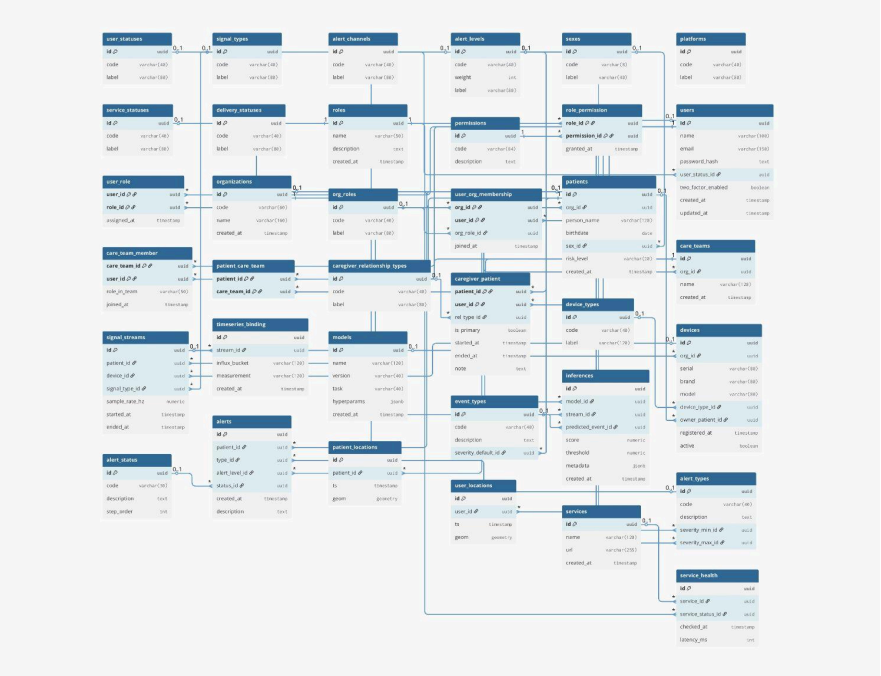


Figura 2: Diagrama RDBMS nuevo de Heartguard.

4.3 Razonamiento de Diseño y Normalización del Nuevo RDBMS (HeartGuard v2.5)

4.3.1 Visión General del Modelo

El nuevo modelo relacional de HeartGuard v2.5 surge de la necesidad de generalizar y escalar la solución más allá del uso doméstico.
Mientras que la primera versión (RBDMS familiar) estaba centrada en una sola entidad supervisora (“papá/mamá”) y miembros asociados, el nuevo modelo busca representar entidades de salud, pacientes, cuidadores, organizaciones, dispositivos e inferencias de IA dentro de una arquitectura modular y extensible.
Su diseño sigue la Tercera Forma Normal (3FN):

- Cada tabla representa una sola entidad conceptual.
- Se eliminan dependencias transitivas o repetitivas.
- Las relaciones se establecen mediante claves foráneas (FK) con reglas explícitas de borrado y restricción.

Esto garantiza integridad referencial, coherencia transaccional y escalabilidad para un sistema multi-tenant (múltiples organizaciones en un mismo despliegue).

4.3.2 Principales Entidades y su Propósito

Grupo	Entidades	Propósito / Justificación
Seguridad y multi-tenant	users, roles, permissions, user_role, organizations, org_roles, user_org_membership	Implementan un modelo de **Control de Acceso Basado en Roles (RBAC)** y **multi-organización**. Cada usuario puede pertenecer a múltiples organizaciones con distintos roles. Se evita duplicación de usuarios y se centraliza la autenticación.
Catálogos	user_statuses, signal_types, alert_levels, alert_channels, sexes, etc.	Reemplazan los antiguos ENUMs. Permiten agregar o modificar códigos sin alterar el esquema, ofreciendo flexibilidad. Esta decisión responde a la experiencia previa donde los ENUMs limitaban despliegues en entornos distintos.
Dominio clínico	patients, care_teams, care_team_member, caregiver_patient, caregiver_relationship_types	Normalizan la relación entre pacientes, equipos médicos y cuidadores. Antes existía una tabla única “usuarios” con campo de tipo. Ahora se separan roles y relaciones, permitiendo múltiples cuidadores o equipos por paciente.
Dispositivos y streams	device_types, devices, signal_streams, timeseries_binding	Introducen el concepto de **IoT médico**. Cada dispositivo está ligado a un paciente y transmite señales fisiológicas registradas como flujos (signal streams). Esta capa es nueva y soporta la integración con sistemas como InfluxDB.

Modelos e inferencias (IA)	models, event_types, inferences	Representan la capa de Machine Learning. Permite registrar versiones de modelos, sus parámetros y los resultados de inferencias sobre señales. En la versión inicial no existía este componente predictivo.
Alertas y ciclo de vida	alert_types, alert_status, alerts, alert_assignment, alert_resolution, alert_delivery	Gestionan el ciclo de vida de una alerta clínica o técnica. Permiten trazabilidad, asignación y resolución. Antes las alertas eran simples flags sin persistencia ni jerarquía de severidad.
Ubicación y operación	patient_locations, user_locations, services, service_health	Permiten rastreo geográfico y monitoreo de servicios internos. Aportan robustez operacional para entornos hospitalarios distribuidos.

4.3.3 Relaciones y Cardinalidades

Relación	Cardinalidad	Explicación
users ↔ roles	N:M vía userrole	Un usuario puede tener múltiples roles; un rol puede pertenecer a varios usuarios.
organizations ↔ users	N:M vía user_org_membership	Soporta multi-organización (hospitales, clínicas, escuelas).
patients ↔ organizations	N:1	Cada paciente pertenece a una organización.
care_teams ↔ users	N:M vía care_team_member	Un equipo de atención tiene varios usuarios, y un usuario puede estar en varios equipos.
patients ↔ care_teams	N:M vía patient_care_team	Permite asignar pacientes a equipos distintos (ej. cardiología, nutrición).
patients ↔ users (cuidadores)	N:M vía caregiver_patient	Modela relaciones de cuidado personal o familiar.
devices ↔ patients	N:1	Cada dispositivo puede tener un paciente asignado; un paciente puede tener varios dispositivos.
signal_streams ↔ devices y patients	N:1 + N:1	Registra los flujos de señal por dispositivo y paciente.
models ↔ inferences	1:N	Un modelo genera múltiples inferencias.
inferences ↔ signal_streams	N:1	Cada inferencia se asocia a un flujo específico.
alerts ↔ patients	N:1	Cada alerta pertenece a un paciente.
alert_status ↔ alerts	1:N	Define el estado actual del ciclo de vida de una alerta.
service_health ↔ services	1:N	Cada registro de salud corresponde a un chequeo del servicio.

4.3.4 Decisiones Clave Frente a la Versión Inicial

Aspecto	Versión Inicial	Nueva Versión (v2.5)
Alcance	Familiar cerrado	Multi-organización y multi-tenant
Normalización	1FN con redundancias (usuarios, síntomas, relaciones directas)	3FN completa con catálogos, relaciones N:M y claves externas
Escalabilidad	Limitada a un hogar	Escalable a cientos de pacientes y dispositivos IoT
Alertas	Flags o booleans	Sistema completo de gestión de alertas con niveles, tipos y estatus
Roles y permisos	Fijos (supervisor / usuario)	RBAC con granularidad y auditoría
Datos de salud	Tablas planas de síntomas	Streams + series temporales + inferencias
Integración con IA	No existía	Soporte a modelos, entrenamiento y resultados predictivos
Ubicación geográfica	No contemplada	Implementación PostGIS para trazabilidad espacial
Mantenibilidad	Cambios manuales en código	Catálogos parametrizables sin alterar el esquema

4.3.5 Beneficios Técnicos

1. **Integridad referencial total**
   Cada entidad se conecta mediante claves foráneas con acciones de ON DELETE RESTRICT o CASCADE, según el impacto deseado.
   Ejemplo: si un paciente se elimina, sus inferencias y alertas se eliminan automáticamente.
2. **Reducción de redundancia y anomalías**
   Un solo registro por paciente, organización o tipo de señal evita inconsistencias al actualizar.
3. **Extensibilidad y analítica avanzada**
   El modelo ahora puede conectarse con sistemas externos (InfluxDB, ML pipelines) mediante las tablas timeseries_binding y inferences.
4. **Acoplamiento débil entre módulos**
   El diseño de HeartGuard v2.5 implementa un **acoplamiento débil** entre los distintos dominios del sistema —clínico, técnico, analítico y operativo—, lo que significa que cada módulo puede operar, mantenerse y escalarse de manera independiente sin comprometer la integridad global de la base de datos.

3. Queries SQL Principales (Ejecutadas desde el Repositorio Go)

A continuación se presenta un listado exhaustivo de las consultas encapsuladas en las funciones del repositorio de Go (`repo.go`), agrupadas por módulo:

Gestión de Organizaciones

- **`GetOrganizations()`:** Obtiene un listado de todas las organizaciones registradas en el sistema.
- **`CreateOrganization()`:** Inserta un nuevo registro de organización en la tabla `organizations`.
- **`GetOrganizationByID()`:** Recupera los detalles de una organización específica a través de su ID.
- **`GetOrganizationDetail()`:** Obtiene una vista detallada de una organización, incluyendo la lista de usuarios y pacientes asociados a ella.
- **`UpdateOrganization()`:** Actualiza los datos de una organización existente (nombre, información de contacto, etc.).

Gestión de Usuarios y Roles

- **`GetUsers(filter)`:** Obtiene una lista paginada de usuarios, con la capacidad de aplicar filtros por organización, rol o estado.
- **`GetUserByID()`:** Recupera los detalles completos de un usuario específico, incluyendo su rol y organización.
- **`GetUserByEmail()`:** Busca y recupera un usuario a partir de su dirección de correo electrónico, usado principalmente para el proceso de login.
- **`CreateUser()`:** Inserta un nuevo registro de usuario en la tabla `users`.
- **`UpdateUser()`:** Modifica la información de un usuario, como su nombre, rol o estado.
- **`GetRoles()`:** Obtiene el catálogo completo de roles disponibles en el sistema desde la tabla `roles`.

Gestión de Pacientes

- **`GetPatients(filter)`:** Obtiene una lista paginada de pacientes, permitiendo filtrar por organización.
- **`CreatePatient()`:** Inserta un nuevo registro de paciente en la tabla `patients`.
- **`GetPatientByID()`:** Recupera los detalles completos de un paciente específico a través de su ID.
- **`UpdatePatient()`:** Actualiza la información demográfica o de estado de un paciente.

Gestión de Equipos de Cuidado (Care Teams) y Cuidadores (Caregivers)

- **`GetCaregivers(filter)`:** Obtiene una lista paginada de cuidadores (usuarios con rol de cuidador).
- **`GetCareTeams(filter)`:** Recupera una lista de equipos de cuidado, permitiendo filtrar por organización.
- **`CreateCareTeam()`:** Crea un nuevo equipo de cuidado asociado a una organización.
- **`UpdateCareTeam()`:** Actualiza el nombre o la información de un equipo de cuidado.
- **`AddPatientToCareTeam()`:** Asocia un paciente a un equipo de cuidado, insertando un registro en `care_team_patients`.
- **`RemovePatientFromCareTeam()`:** Desvincula un paciente de un equipo de cuidado.
- **`AddCaregiverToCareTeam()`:** Asocia un cuidador a un equipo de cuidado, insertando un registro en `care_team_caregivers`.
- **`RemoveCaregiverFromCareTeam()`:** Desvincula a un cuidador de un equipo de cuidado.

Gestión de Dispositivos

- **`GetDevices(filter)`:** Obtiene una lista paginada de dispositivos de monitoreo, con filtros por paciente u organización.
- **`CreateDevice()`:** Registra un nuevo dispositivo en el sistema.
- **`GetDeviceByID()`:** Recupera la información de un dispositivo específico.
- **`UpdateDevice()`:** Actualiza el estado o la asignación de un dispositivo a un paciente.

Alertas y Eventos

- **`GetAlerts(filter)`:** Consulta las alertas del sistema, permitiendo filtrar por rango de fechas, paciente, tipo de alerta o estado.
- **`GetEventTypes()`:** Obtiene el catálogo completo de tipos de eventos desde la tabla `event_types`.

Seguridad y Auditoría

- **`GetAPIKeys(userID)`:** Obtiene todas las claves de API asociadas a un usuario específico.
- **`CreateAPIKey()`:** Genera y almacena una nueva clave de API para un usuario o servicio.
- **`DeleteAPIKey()`:** Elimina una clave de API específica.
- **`GetAPIKeyByKeyValue()`:** Valida una clave de API buscándola por su valor.
- **`GetAuditLogs(page, pageSize)`:** Obtiene los registros de auditoría de forma paginada para supervisar la actividad del sistema.
- **`CreateAuditLog()`:** Inserta una nueva entrada en el registro de auditoría.

Gestión de Contenido

- **`GetContentBlocks()`:** Obtiene todos los bloques de contenido administrable.
- **`CreateContentBlock()`:** Crea un nuevo bloque de contenido.
- **`GetContentBlockByID()`:** Recupera un bloque de contenido específico por su ID.
- **`UpdateContentBlock()`:** Actualiza el contenido de un bloque existente.

Stored Procedures

Catálogos

- **`sp_catalog_resolve`**: Resuelve el nombre de la tabla de un catálogo específico.
- **`sp_catalog_list`**: Lista los registros de un catálogo.
- **`sp_catalog_create`**: Crea un nuevo registro en un catálogo.
- **`sp_catalog_update`**: Actualiza un registro en un catálogo.
- **`sp_catalog_delete`**: Elimina un registro de un catálogo.

Pacientes

- **`sp_patient_create`**: Crea un nuevo paciente.
- **`sp_patient_update`**: Actualiza la información de un paciente.
- **`sp_patient_list`**: Lista pacientes con filtros y paginación.
- **`sp_patient_get`**: Obtiene un paciente por su ID.
- **`sp_patient_locations_list`**: Lista las ubicaciones de un paciente.
- **`sp_patient_location_create`**: Crea una nueva ubicación para un paciente.
- **`sp_patient_caregivers_list`**: Lista los cuidadores de un paciente.
- **`sp_patient_caregiver_add`**: Añade un cuidador a un paciente.
- **`sp_patient_caregiver_remove`**: Remueve un cuidador de un paciente.

Invitaciones

- **`sp_invitation_create`**: Crea una nueva invitación.
- **`sp_invitation_get`**: Obtiene una invitación por su token.
- **`sp_invitation_list`**: Lista invitaciones con filtros.
- **`sp_invitation_accept`**: Acepta una invitación.
- **`sp_invitation_reject`**: Rechaza una invitación.
- **`sp_invitation_cancel`**: Cancela una invitación.

Señales y Streams

- **`sp_signal_stream_create`**: Crea un nuevo stream de señales.
- **`sp_signal_stream_list`**: Lista streams de señales.
- **`sp_signal_stream_get`**: Obtiene un stream por su ID.

- `sp_timeseries_binding_create`: Crea un binding de series de tiempo.
- `sp_timeseries_binding_list`: Lista bindings de series de tiempo.
- `sp_timeseries_binding_get`: Obtiene un binding por su ID.

Dispositivos

- `sp_device_create`: Crea un nuevo dispositivo.
- `sp_device_update`: Actualiza la información de un dispositivo.
- `sp_device_list`: Lista dispositivos con filtros.
- `sp_device_get`: Obtiene un dispositivo por su ID.
- `sp_push_device_create`: Crea un nuevo dispositivo push.
- `sp_push_device_update`: Actualiza un dispositivo push.
- `sp_push_device_list`: Lista dispositivos push.
- `sp_push_device_get`: Obtiene un dispositivo push por su ID.

Modelos e Inferencias

- `sp_model_create`: Crea un nuevo modelo de ML.
- `sp_model_update`: Actualiza un modelo de ML.
- `sp_model_list`: Lista modelos de ML.
- `sp_model_get`: Obtiene un modelo de ML por su ID.
- `sp_inference_create`: Crea una nueva inferencia.
- `sp_inference_list`: Lista inferencias.
- `sp_inference_get`: Obtiene una inferencia por su ID.
- `sp_ground_truth_create`: Crea un nuevo registro de ground truth.
- `sp_ground_truth_update`: Actualiza un registro de ground truth.
- `sp_ground_truth_list`: Lista registros de ground truth.
- `sp_ground_truth_get`: Obtiene un registro de ground truth por su ID.

Tipos de Eventos y Alertas

- `sp_event_type_create`: Crea un nuevo tipo de evento.
- `sp_event_type_update`: Actualiza un tipo de evento.
- `sp_event_type_list`: Lista tipos de evento.
- `sp_event_type_get`: Obtiene un tipo de evento por su ID.
- `sp_alert_create`: Crea una nueva alerta.
- `sp_alert_update`: Actualiza una alerta.
- `sp_alert_list`: Lista alertas con filtros.
- `sp_alert_get`: Obtiene una alerta por su ID.

Contenido Editorial

- `sp_content_create`: Crea un nuevo contenido.
- `sp_content_update`: Actualiza un contenido.

- `sp_content_delete`: Elimina un contenido.
- `sp_content_get`: Obtiene un contenido por su ID.
- `sp_content_list`: Lista contenidos con filtros.

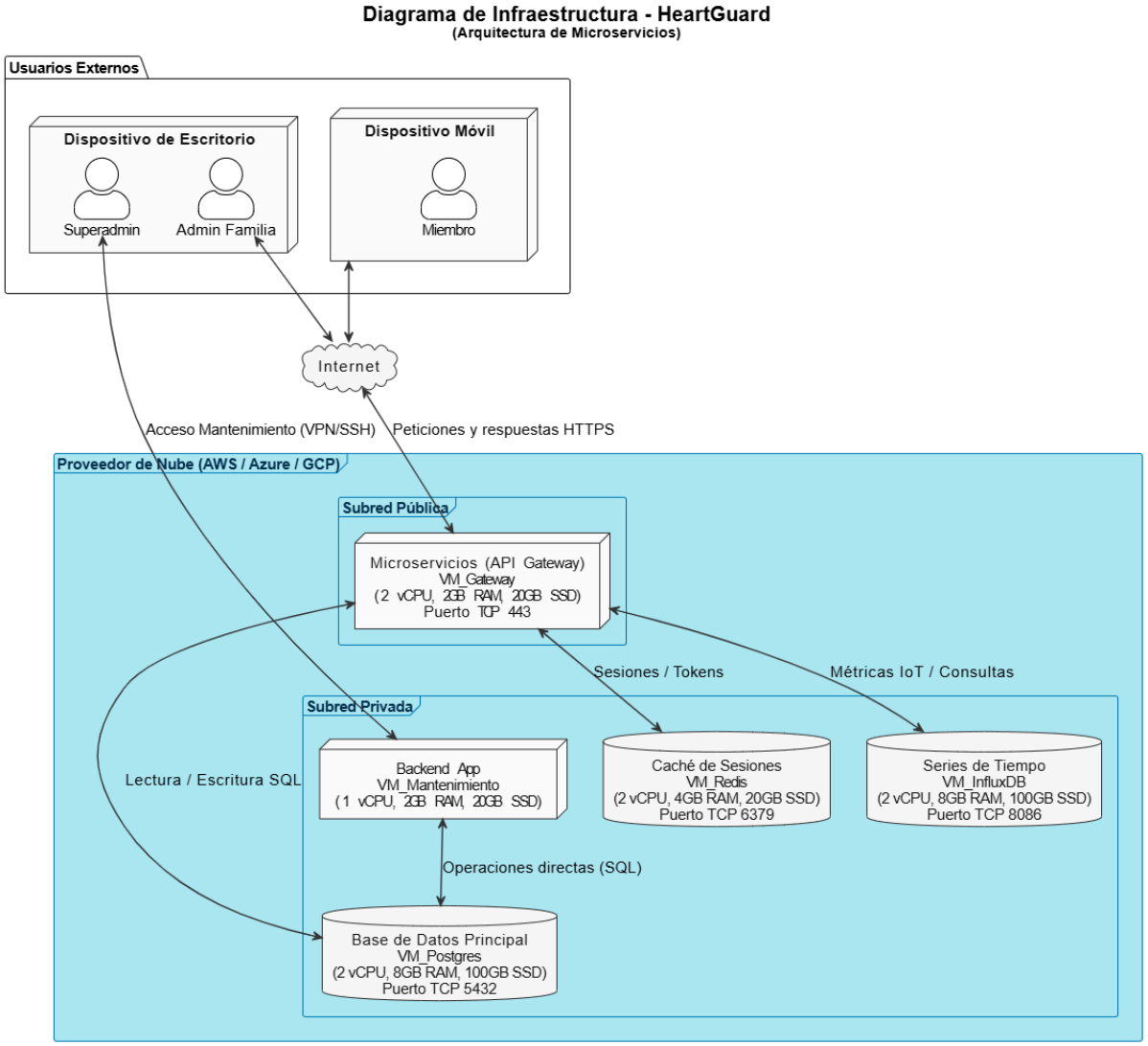
Métricas y Reportes

- `sp_metrics_dashboard`: Obtiene métricas para el dashboard.
- `sp_metrics_recent_activity`: Obtiene actividad reciente del sistema.
- `sp_metrics_user_status_breakdown`: Obtiene estadísticas de estados de usuarios.
- `sp_metrics_invitation_breakdown`: Obtiene estadísticas de invitaciones.
- `sp_metrics_content_snapshot`: Obtiene un snapshot de métricas de contenido.
- `sp_metrics_content_report`: Genera un reporte detallado de contenido.
- `sp_metrics_operations_report`: Genera un reporte de operaciones.
- `sp_metrics_users_report`: Genera un reporte de usuarios.

Vistas

- `v_patient_active_alerts`: Vista que muestra las alertas activas de los pacientes para una consulta rápida.

4. Diagrama hardware



La Arquitectura de Hardware de HeartGuard fue diseñada con un enfoque estratégico que prioriza rendimiento, seguridad, escalabilidad y mantenibilidad, pilares esenciales para una plataforma moderna orientada a entornos institucionales como hospitales, centros médicos, escuelas u organizaciones corporativas. Su diseño modular permite adaptar la infraestructura a distintos contextos sin modificar la lógica central del sistema.

- **Enfoque modular y desacoplado:**
  La arquitectura sigue el principio de *desacoplamiento semántico*, donde cada servicio opera de manera independiente pero coordinada. Este modelo garantiza que la falla o actualización de un módulo no afecte la operatividad

del resto, lo que es clave en entornos donde la disponibilidad continua es crítica, como hospitales o centros de monitoreo.

- **Infraestructura en la nube (AWS / Azure / GCP):**
  Se optó por una infraestructura basada en la nube para aprovechar su elasticidad, disponibilidad global y escalabilidad bajo demanda. Esto permite a HeartGuard adaptarse fácilmente a distintos tamaños de implementación, desde pequeños centros médicos hasta redes institucionales más amplias, sin necesidad de infraestructura física local. Además, las plataformas de nube ofrecen herramientas nativas de respaldo, monitoreo y cumplimiento normativo (HIPAA, ISO, etc.), esenciales para sistemas que manejan información sensible.

- **Segmentación de redes (seguridad por capas):**
  La arquitectura se divide en subred pública y subred privada con el objetivo de reforzar la seguridad perimetral.

    - La subred pública aloja el API Gateway, encargado de recibir y canalizar todas las solicitudes externas mediante HTTPS, validando credenciales, tokens JWT y políticas de acceso antes de redirigirlas a los microservicios correspondientes. Esto protege el núcleo del sistema y centraliza el control de entrada.

    - La subred privada alberga los componentes críticos como los servicios internos y las bases de datos, aislados del tráfico externo para garantizar confidencialidad, estabilidad y protección ante ataques.

- **Gestión segura y control administrativo:**
  El acceso administrativo se restringe al Superadmin mediante canales seguros como VPN o SSH, evitando la exposición directa del backend a Internet. Esto asegura un entorno controlado para tareas de mantenimiento, auditoría y despliegue, fundamental para entornos institucionales donde la trazabilidad y el control de acceso son indispensables.

- **Distribución especializada de bases de datos:**
  La arquitectura incorpora tres motores de datos, cada uno optimizado para su función específica, aplicando el principio de responsabilidad única:

    - **PostgreSQL:** Base relacional encargada de almacenar información estructurada, transaccional y sensible, garantizando integridad y consistencia en los registros médicos, administrativos o de usuario.

    - **Redis:** Sistema de caché en memoria de alta velocidad utilizado para gestión de sesiones, autenticación, tokens y almacenamiento temporal, mejorando la capacidad de respuesta y reduciendo la carga de los microservicios.

    - **InfluxDB:** Base orientada a series de tiempo para almacenar métricas continuas o datos biométricos provenientes de sensores IoT, ideal para monitorear variables clínicas o estadísticas operativas en tiempo real.

- **Asignación eficiente de recursos virtuales:**
  Cada máquina virtual (VM) fue configurada según el perfil de carga esperado:

    - Los microservicios y el API Gateway priorizan procesamiento concurrente y comunicación segura, utilizando instancias ligeras de 2 vCPU y 2 GB de RAM para balancear eficiencia y rendimiento.

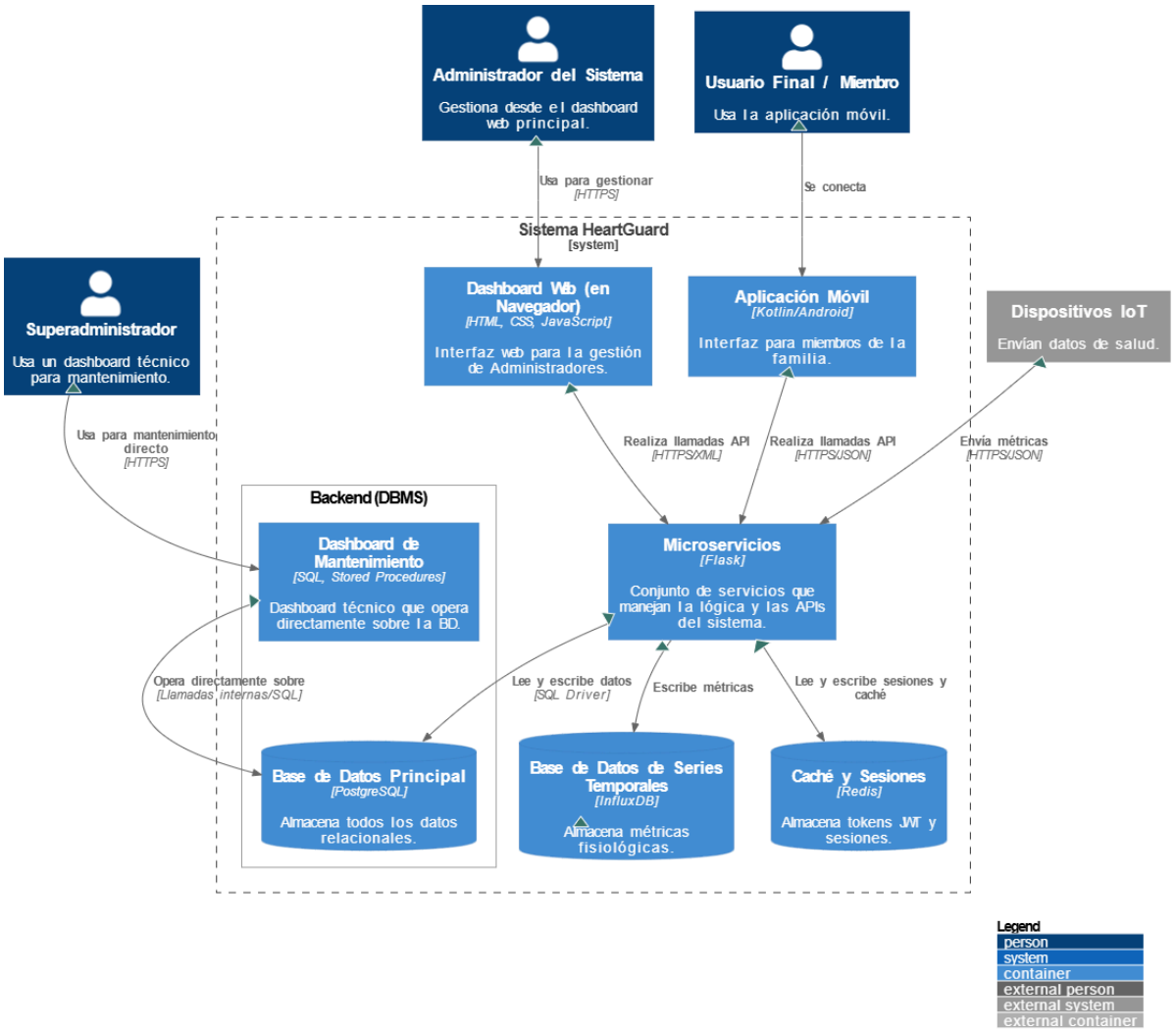
    - Las bases de datos y sistemas de métricas priorizan memoria y almacenamiento, empleando configuraciones con 8 GB de RAM y discos SSD de alta velocidad para mantener la integridad de grandes volúmenes de datos.
      Esta planificación permite optimizar costos sin sacrificar rendimiento, manteniendo un equilibrio entre consumo de recursos y velocidad de procesamiento.

- **Seguridad, redundancia y resiliencia operativa:**
  Todas las comunicaciones internas utilizan puertos cifrados y protocolos

seguros, y la arquitectura está preparada para respaldos automáticos, replicación de datos y recuperación ante desastres. Esto asegura alta disponibilidad, incluso en caso de fallos en una instancia o subred.

- **Finalidad y visión de expansión:**
  El objetivo de esta arquitectura no es solo soportar la operación actual, sino preparar el sistema para su crecimiento futuro. Su diseño modular facilita la incorporación de nuevas capacidades como analítica avanzada, machine learning, paneles de control en tiempo real o integraciones con sistemas hospitalarios (HIS/EMR). En esencia, se construyó una infraestructura adaptable, segura y escalable, capaz de sostener la evolución de HeartGuard como una plataforma integral de gestión y monitoreo institucional.

5. Diagrama software



Arquitectura general de HeartGuard

Estructura en tres niveles:
El sistema está organizado en capas claramente separadas según su función y nivel de exposición.

Capa interna – Backend privado (en VM)

- Ubicación: Dentro de una máquina virtual segura (VM).
- Uso exclusivo: Solo para el superadministrador.
- Acceso: Local y privado — sin APIs REST ni endpoints externos.
- Componentes:
    - Backend interno: ejecuta tareas de gestión, mantenimiento y control.
    - Dashboard técnico: interfaz Web para supervisión directa.
    - Base de datos PostgreSQL: almacena toda la información relacional (usuarios, roles, organizaciones, registros, etc.).
- Propósito: Administración técnica y mantenimiento de la infraestructura del sistema.

Capa intermedia – Microservicios operativos

- Tecnología: Microservicios desarrollados con Flask (Python).
- Rol: Manejan la lógica de negocio y el procesamiento de datos.
- Funciones clave:
    - Gestión de usuarios, pacientes, dispositivos y roles.
    - Procesamiento de métricas de salud recibidas desde dispositivos IoT.
    - Comunicación con bases de datos especializadas:

        - Redis: sesiones y caché.
        - InfluxDB: almacenamiento de series temporales (ritmo cardiaco, presión, etc.).
    - Enrutamiento de solicitudes hacia los clientes autorizados (dashboard web y app móvil).
- Seguridad: Conexiones internas cifradas (HTTPS, JWT, control de roles).

Capa externa – Interfaz de usuarios

1. Dashboard Web (HTML/CSS/JS)

- Usuarios: Administradores de organizaciones.
- Función: Gestionan personal, dispositivos, pacientes y alertas.
- Conexión: HTTPS hacia los microservicios (nunca directo a la BD).

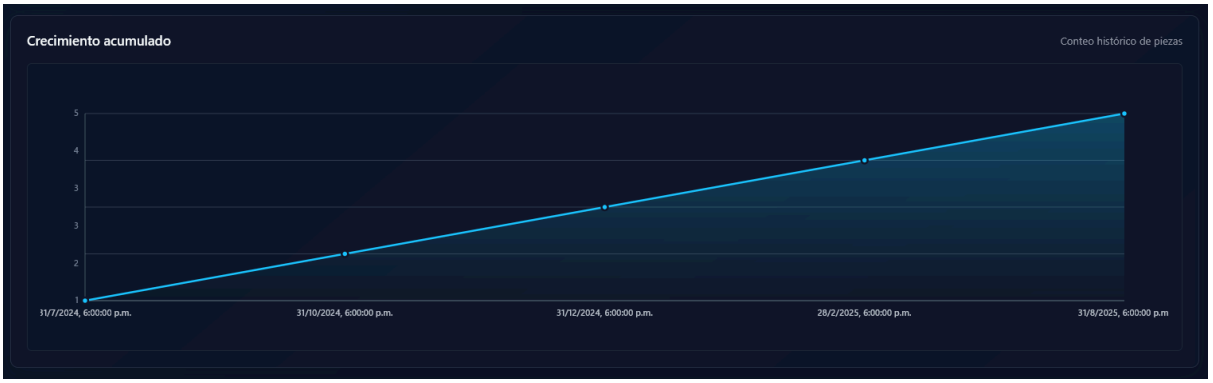
2. Aplicación Móvil (Kotlin/Android)

- Usuarios: Miembros o usuarios finales.
- Función: Visualizan métricas, historial de salud y notificaciones.
- Comunicación: HTTPS/JSON con los microservicios.

6. 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
IA	Creación y entretenimiento de un modelo de IA	Permite analizar que tipo de alerta se le enviará al usuario	Configurar con app
Android	Aplicación móvil	Amplia compatibilidad con usuarios	Configurar Android Studio y librerías

7. Presentación de KPIs, métricas y al menos 6 gráficas generadas



1. Crecimiento Histórico de Usuarios

- **Descripción:** Un gráfico de líneas que muestra el número acumulado de usuarios (pacientes, cuidadores, administradores) registrados en la plataforma a lo largo del tiempo (mes a mes).
- **Interpretación:** Permite visualizar la velocidad de adopción y escalabilidad del sistema. Un crecimiento constante indica una buena aceptación en el mercado.

2. Estado y Flujo de Invitaciones

- **Descripción:** Un gráfico de pastel o de dona que desglosa el estado actual de todas las invitaciones enviadas a través del sistema (Pendientes, Aceptadas, Rechazadas, Canceladas).
- **Interpretación:** Ofrece una visión clara de la efectividad del proceso de onboarding. Un alto número de invitaciones pendientes o rechazadas podría indicar problemas en la comunicación o en el proceso de registro.

3. Resumen de Operaciones del Sistema

- **Descripción:** Un gráfico de barras que muestra el volumen de las operaciones más importantes realizadas en un período determinado (ej. últimos 30 días), como "pacientes creados", "alertas generadas", "dispositivos registrados".
- **Interpretación:** Ayuda a entender qué partes del sistema son las más utilizadas y a dimensionar la carga operativa.

4. Actividad Reciente del Sistema

- **Descripción:** Una línea de tiempo o una lista que muestra las últimas acciones significativas realizadas en el sistema, como "El Dr. Smith se conectó" o "Nueva alerta para el paciente John Doe".
- **Interpretación:** Proporciona una visión en tiempo real de lo que está sucediendo en la plataforma, útil para la supervisión y la detección de actividad inusual.

5. Estado del Contenido Editorial

- **Descripción:** Un gráfico de dona que muestra la proporción del contenido editorial según su estado: "Publicado", "Borrador", "Archivado".

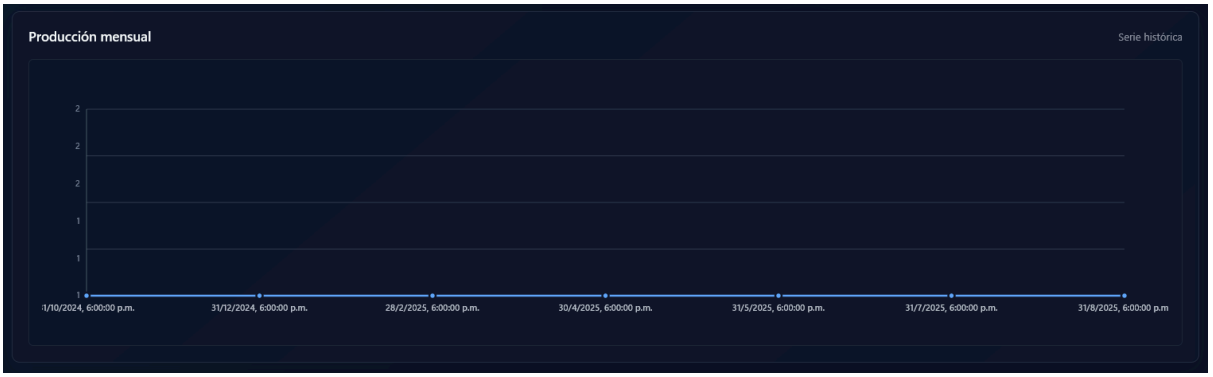
- **Interpretación:** Permite a los administradores de contenido ver de un vistazo cuánto material está disponible para los usuarios y cuánto está en proceso de creación o revisión.

6. Distribución por Categorías Principales

- **Descripción:** Un gráfico de barras que muestra el número de elementos (como contenidos o tipos de eventos) registrados en las categorías más relevantes del sistema.
- **Interpretación:** Ayuda a identificar qué temas o tipos de eventos son los más comunes, lo que puede guiar la creación de nuevo contenido o la optimización de procesos.

7. Actividad por Rol de Usuario

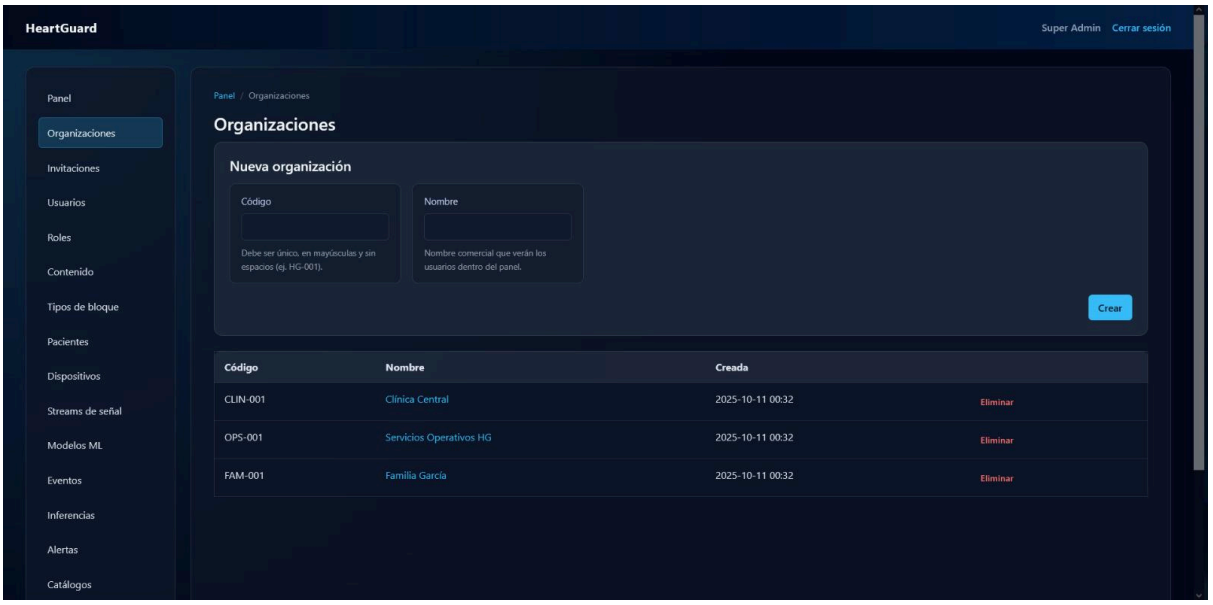
- **Descripción:** Un gráfico de barras apiladas que muestra el volumen de actividad (acciones realizadas) desglosado por cada rol de usuario (Administrador, Cuidador).
- **Interpretación:** Permite comparar el nivel de interacción de los diferentes tipos de usuario con la plataforma, ayudando a entender cómo cada rol utiliza el sistema.

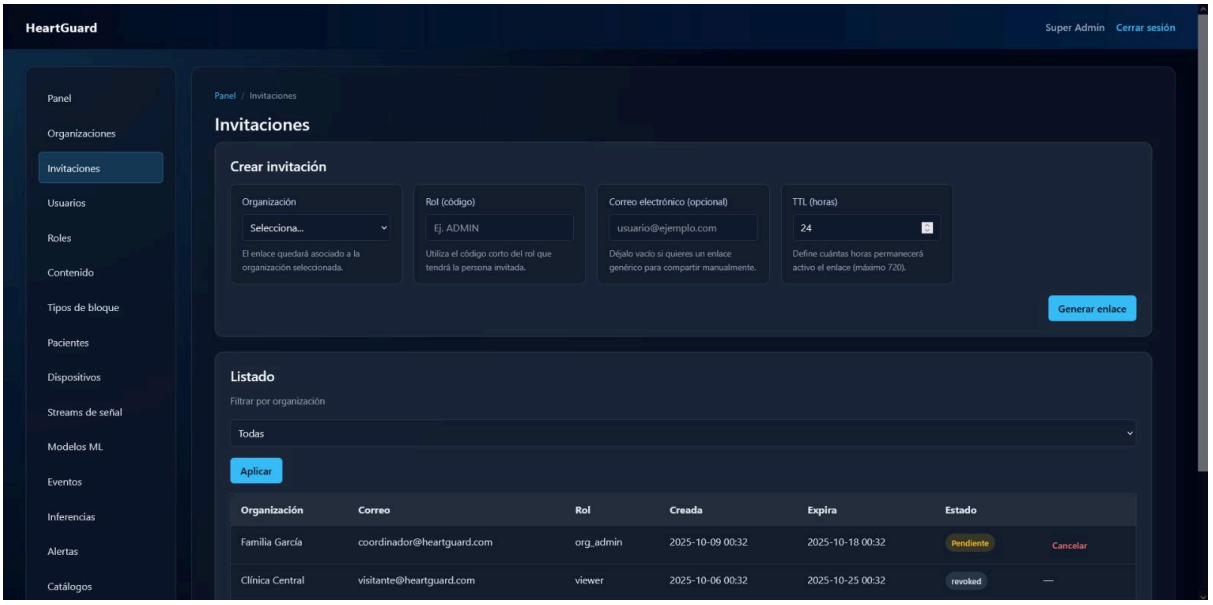


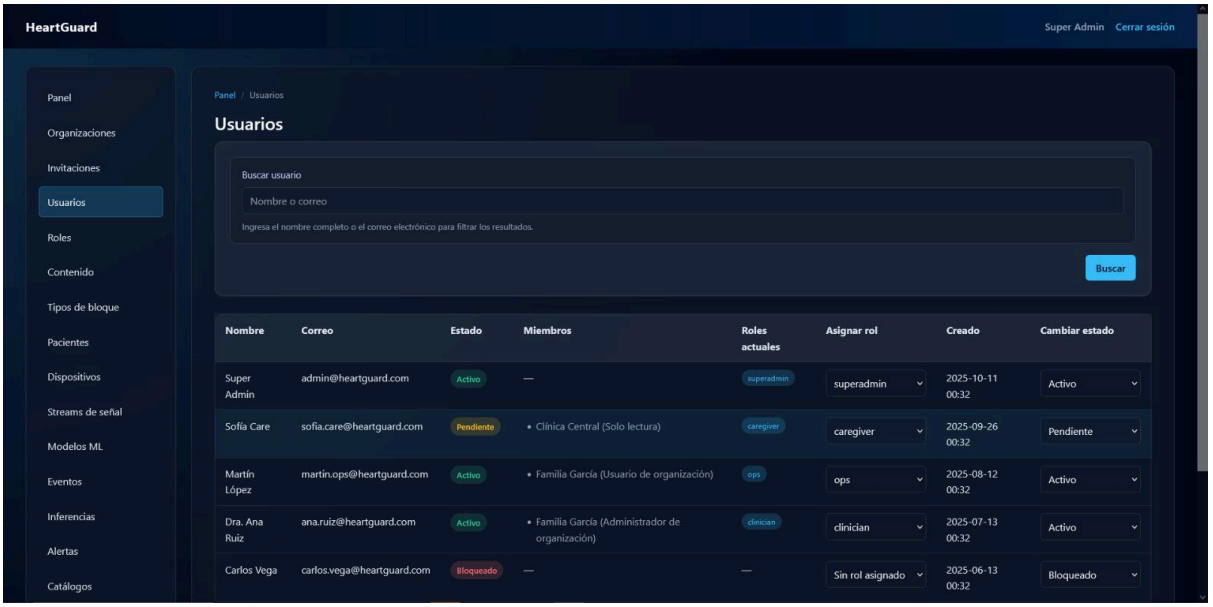
8. Productividad Mensual por Organización

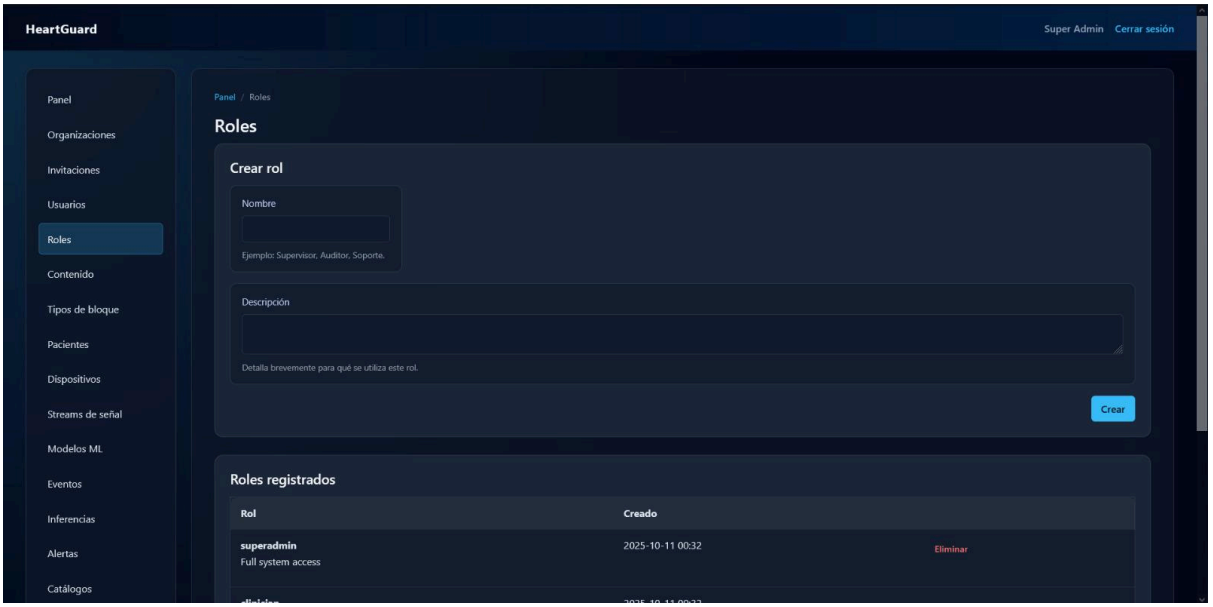
- **Descripción:** Un gráfico de barras agrupadas que compara métricas clave (ej. nuevos pacientes, alertas gestionadas) entre las diferentes organizaciones que utilizan la plataforma mes a mes.
- **Interpretación:** Facilita la identificación de las organizaciones más activas y permite comparar su rendimiento, lo que puede ser útil para la gestión de cuentas y el soporte.

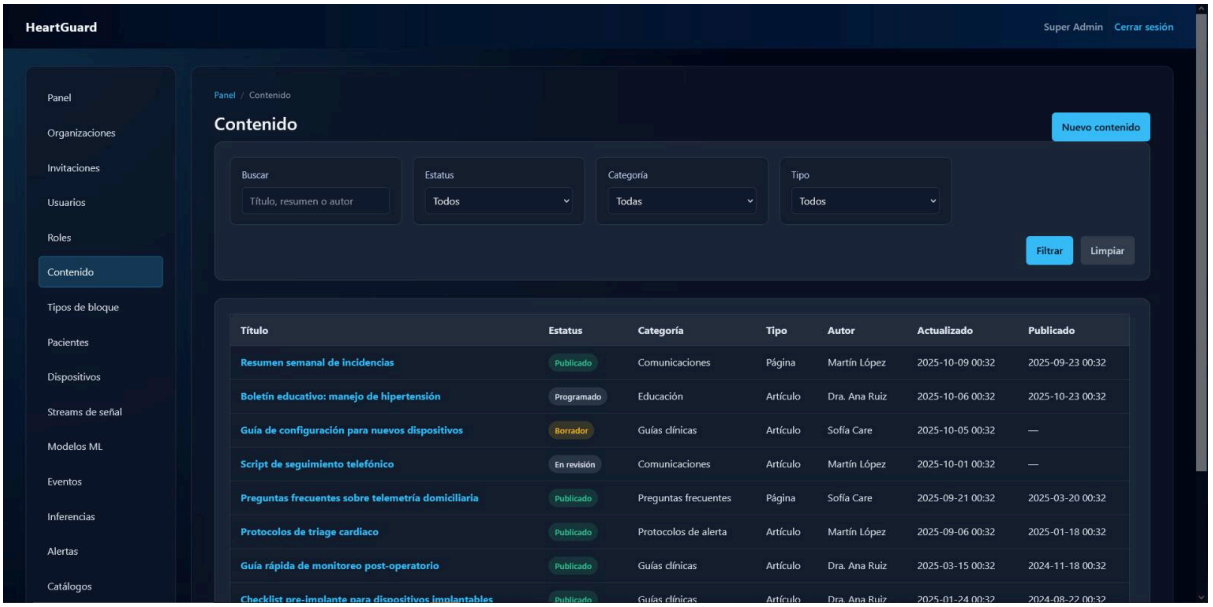
8. Capturas de pantalla del sistema en funcionamiento

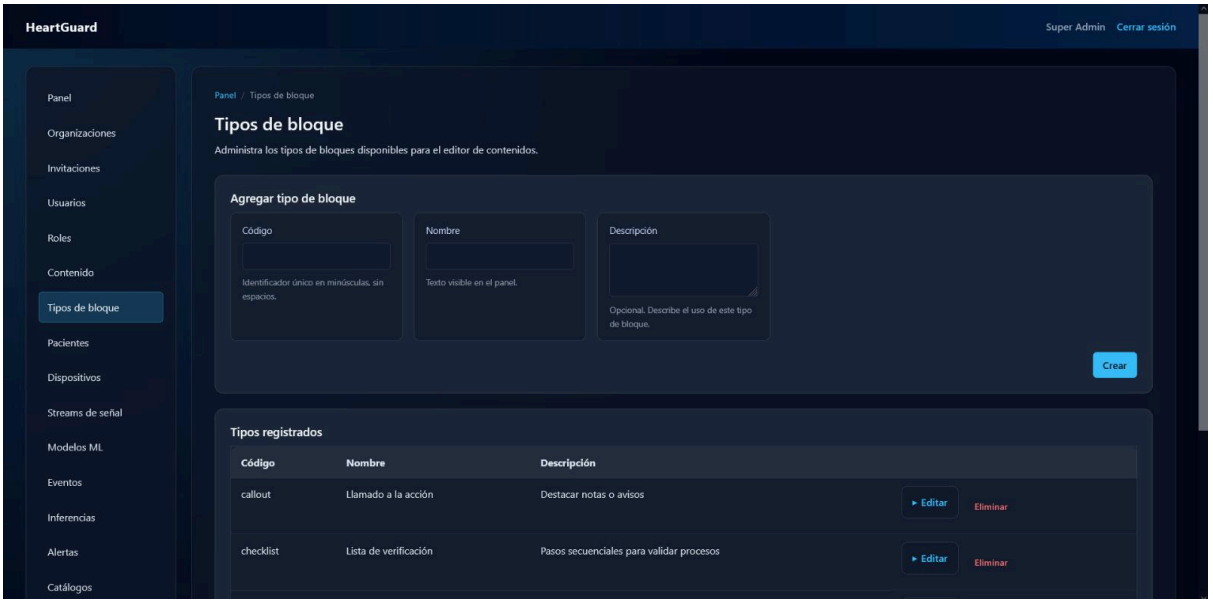


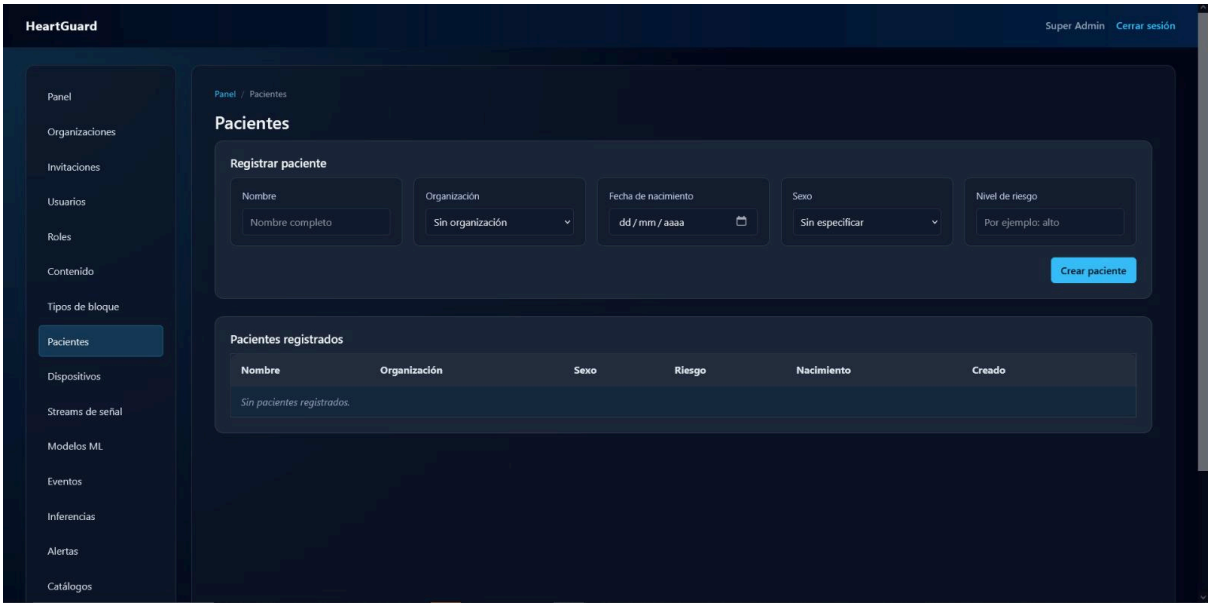


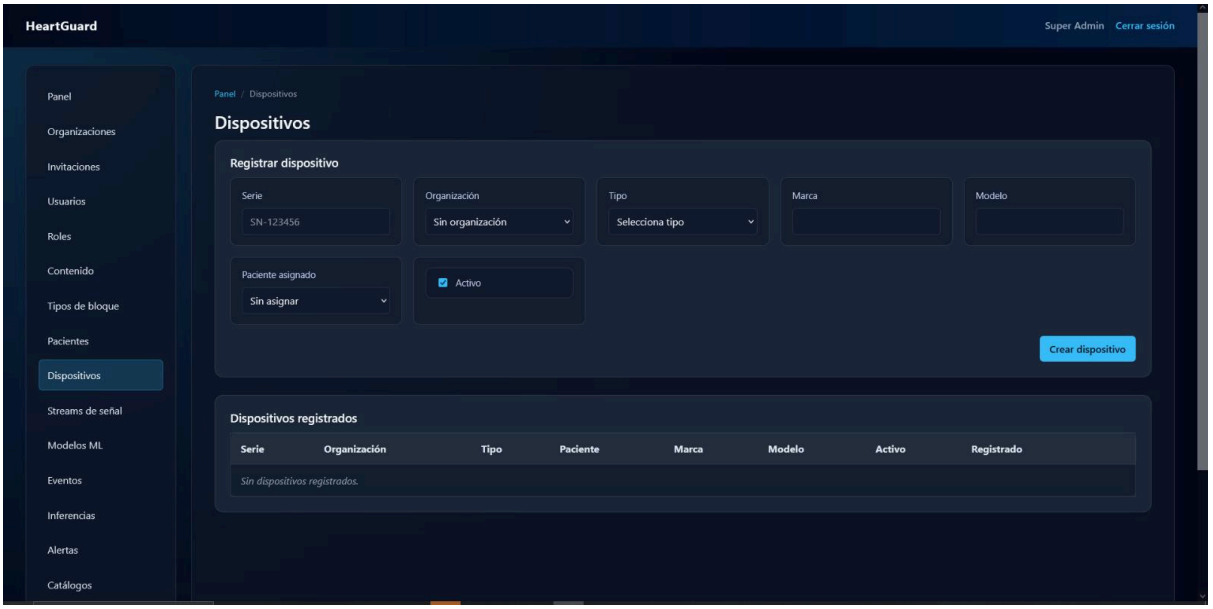

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Usuarios
Usuarios
Buscar usuario
Nombre o correo
Ingresa el nombre completo o el correo electrónico para filtrar los resultados.
Buscar
Nombre
Correo
Estado
Miembros
Roles actuales
Asignar rol
Creado
Cambiar estado
Super Admin
admin@heartguard.com
Activo
—
superadmin
superadmin
2025-10-11 00:32
Activo
Sofía Care
sofia.care@heartguard.com
Pendiente
• Clínica Central (Solo lectura)
caregiver
caregiver
2025-09-26 00:32
Pendiente
Martín López
martin.ops@heartguard.com
Activo
• Familia García (Usuario de organización)
ops
ops
2025-08-12 00:32
Activo
Dra. Ana Ruiz
ana.ruiz@heartguard.com
Activo
• Familia García (Administrador de organización)
clinician
clinician
2025-07-13 00:32
Activo
Carlos Vega
carlos.vega@heartguard.com
Bloqueado
—
—
Sin rol asignado
2025-06-13 00:32
Bloqueado

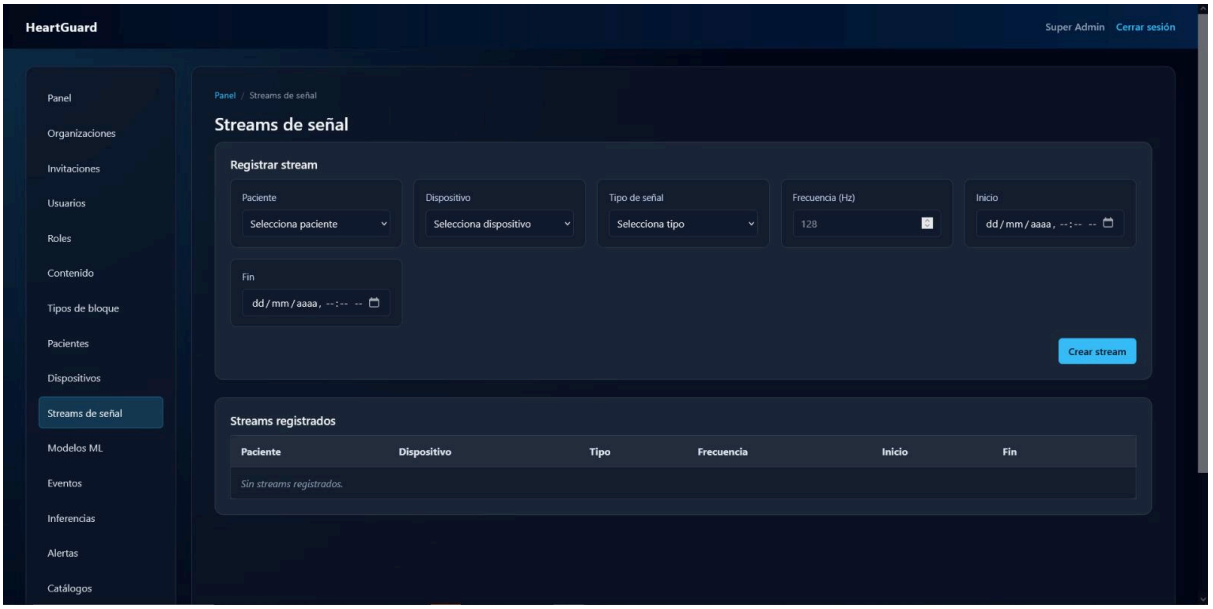

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Roles
Roles
Crear rol
Nombre
Ejemplo: Supervisor, Auditor, Soporte.
Descripción
Detalla brevemente para qué se utiliza este rol.
Crear
Roles registrados
Rol
Creado
superadmin
Full system access
2025-10-11 00:32
Eliminar

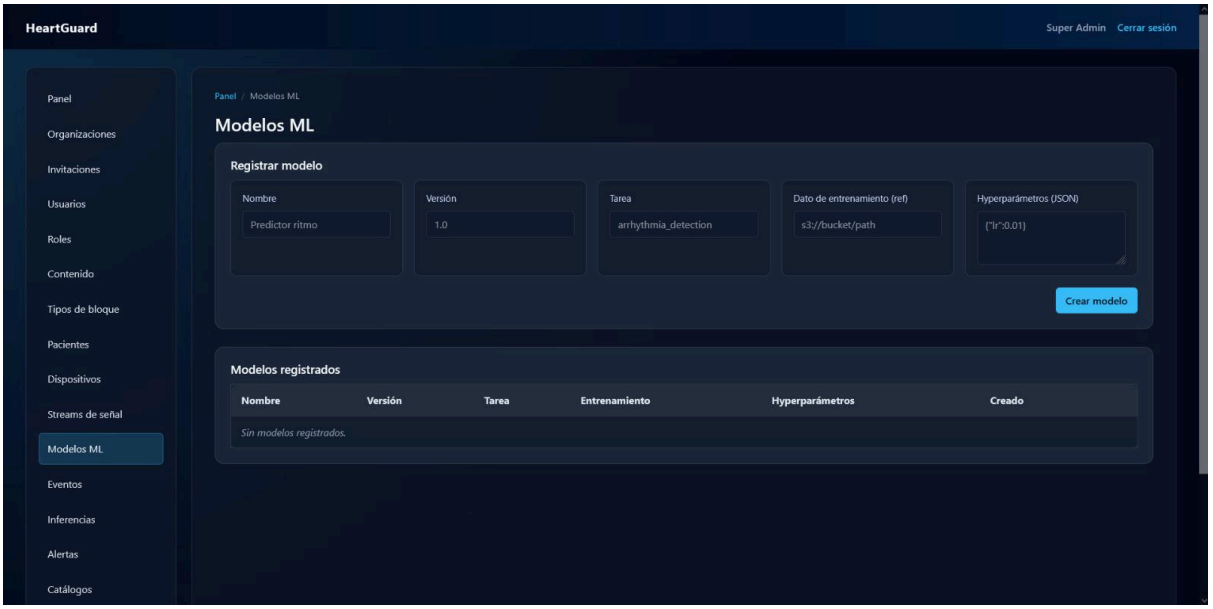

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Contenido
Contenido
Nuevo contenido
Buscar
Título, resumen o autor
Estatus
Todos
Categoría
Todas
Tipo
Todos
Filtrar
Limpiar
Título
Estatus
Categoría
Tipo
Autor
Actualizado
Publicado
Resumen semanal de incidencias
Publicado
Comunicaciones
Página
Martín López
2025-10-09 00:32
2025-09-23 00:32
Boletín educativo: manejo de hipertensión
Programado
Educación
Artículo
Dra. Ana Ruiz
2025-10-06 00:32
2025-10-23 00:32
Guía de configuración para nuevos dispositivos
Borrador
Guías clínicas
Artículo
Sofía Care
2025-10-05 00:32
—
Script de seguimiento telefónico
En revisión
Comunicaciones
Artículo
Martín López
2025-10-01 00:32
—
Preguntas frecuentes sobre telemetría domiciliaria
Publicado
Preguntas frecuentes
Página
Sofía Care
2025-09-21 00:32
2025-03-20 00:32
Protocolos de triage cardiaco
Publicado
Protocolos de alerta
Artículo
Martín López
2025-09-06 00:32
2025-01-18 00:32
Guía rápida de monitoreo post-operatorio
Publicado
Guías clínicas
Artículo
Dra. Ana Ruiz
2025-03-15 00:32
2024-11-18 00:32
Checklist pre-implante para dispositivos implantables
Publicado
Guías clínicas
Artículo
Dra. Ana Ruiz
2025-01-24 00:32
2024-08-22 00:32

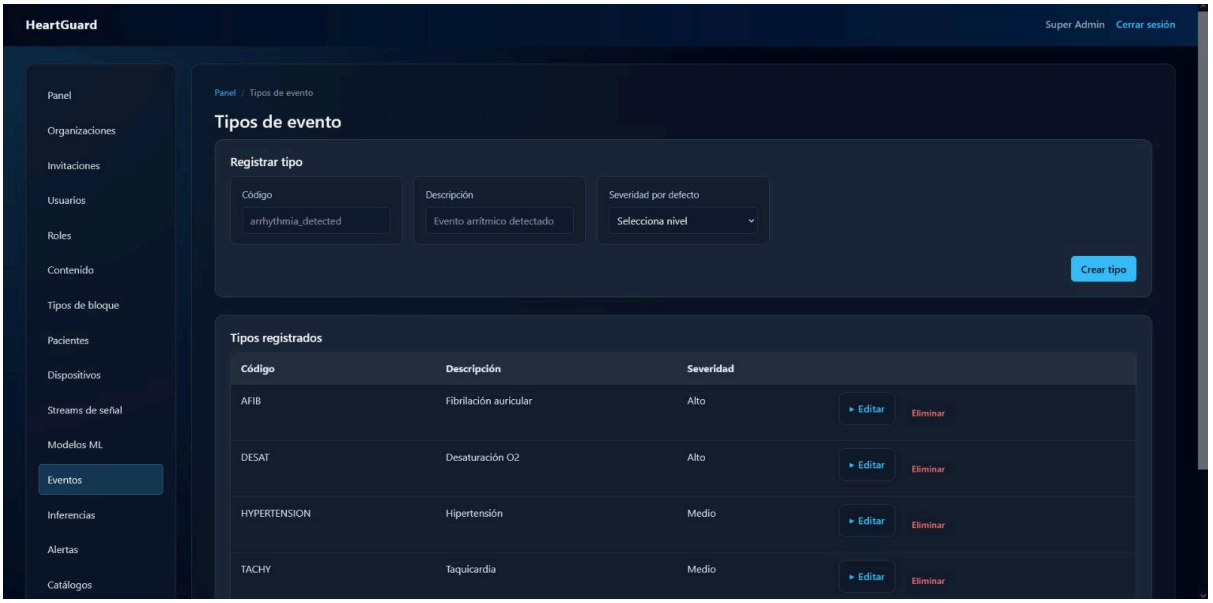

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Tipos de bloque
Tipos de bloque
Administra los tipos de bloques disponibles para el editor de contenidos.
Agregar tipo de bloque
Código
Identificador único en minúsculas, sin espacios.
Nombre
Texto visible en el panel.
Descripción
Opcional. Describe el uso de este tipo de bloque.
Crear
Tipos registrados
Código
Nombre
Descripción
callout
Llamado a la acción
Destacar notas o avisos
Editar
Eliminar
checklist
Lista de verificación
Pasos secuenciales para validar procesos
Editar
Eliminar

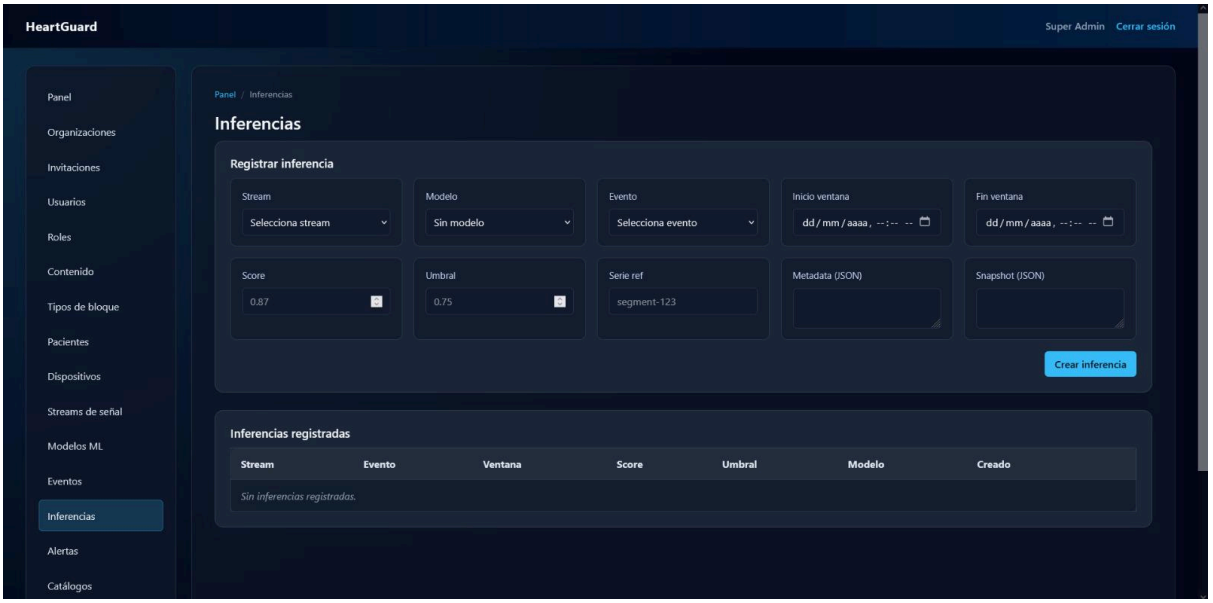

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Pacientes
Pacientes
Registrar paciente
Nombre
Nombre completo
Organización
Sin organización
Fecha de nacimiento
dd / mm / aaaa
Sexo
Sin especificar
Nivel de riesgo
Por ejemplo: alto
Crear paciente
Pacientes registrados
Nombre
Organización
Sexo
Riesgo
Nacimiento
Creado
Sin pacientes registrados.

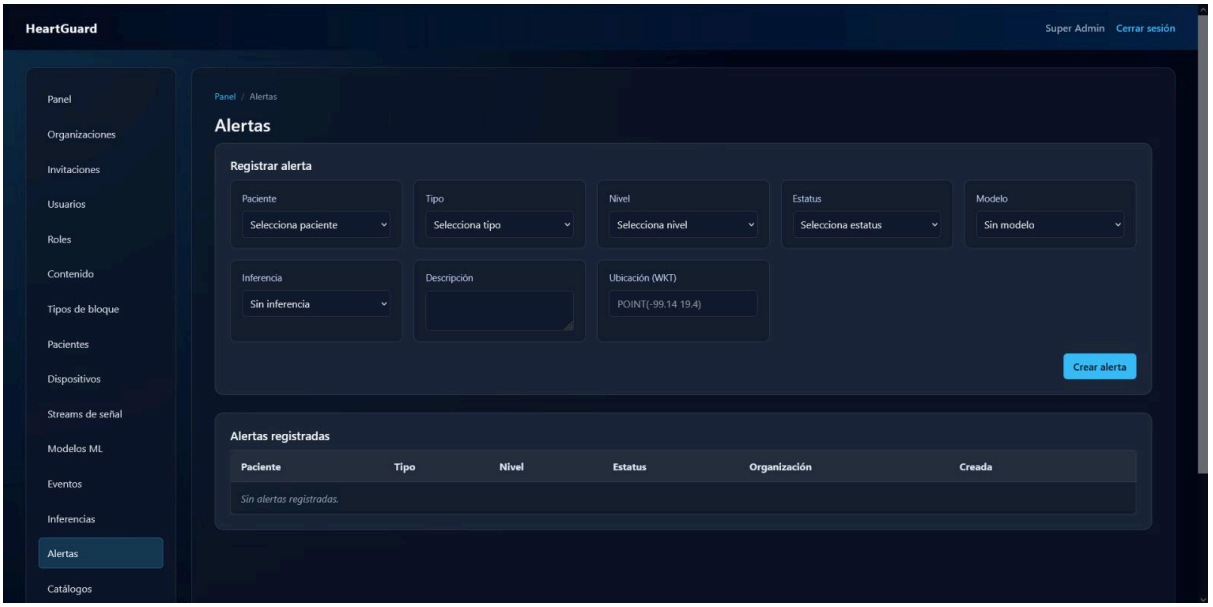

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Dispositivos
Dispositivos
Registrar dispositivo
Serie
SN-123456
Organización
Sin organización
Tipo
Selecciona tipo
Marca
Modelo
Paciente asignado
Sin asignar
Activo
Crear dispositivo
Dispositivos registrados
Serie
Organización
Tipo
Paciente
Marca
Modelo
Activo
Registrado
Sin dispositivos registrados.

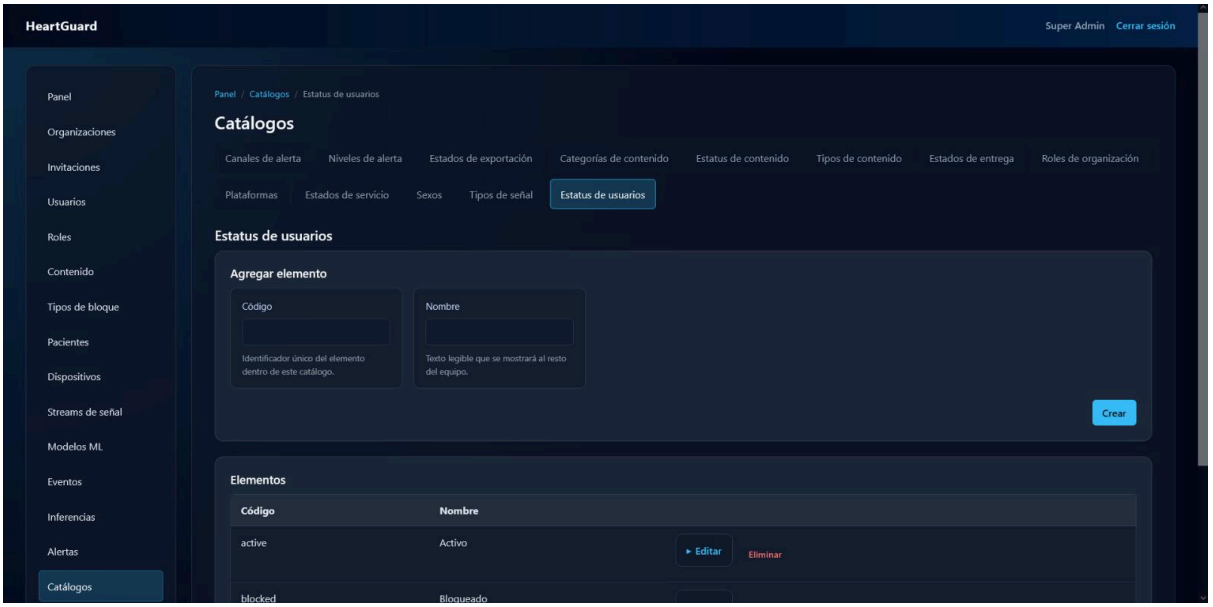

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Streams de señal
Streams de señal
Registrar stream
Paciente
Selecciona paciente
Dispositivo
Selecciona dispositivo
Tipo de señal
Selecciona tipo
Frecuencia (Hz)
128
Inicio
dd / mm / aaaa, --:-- --
Fin
dd / mm / aaaa, --:-- --
Crear stream
Streams registrados
Paciente
Dispositivo
Tipo
Frecuencia
Inicio
Fin
Sin streams registrados.

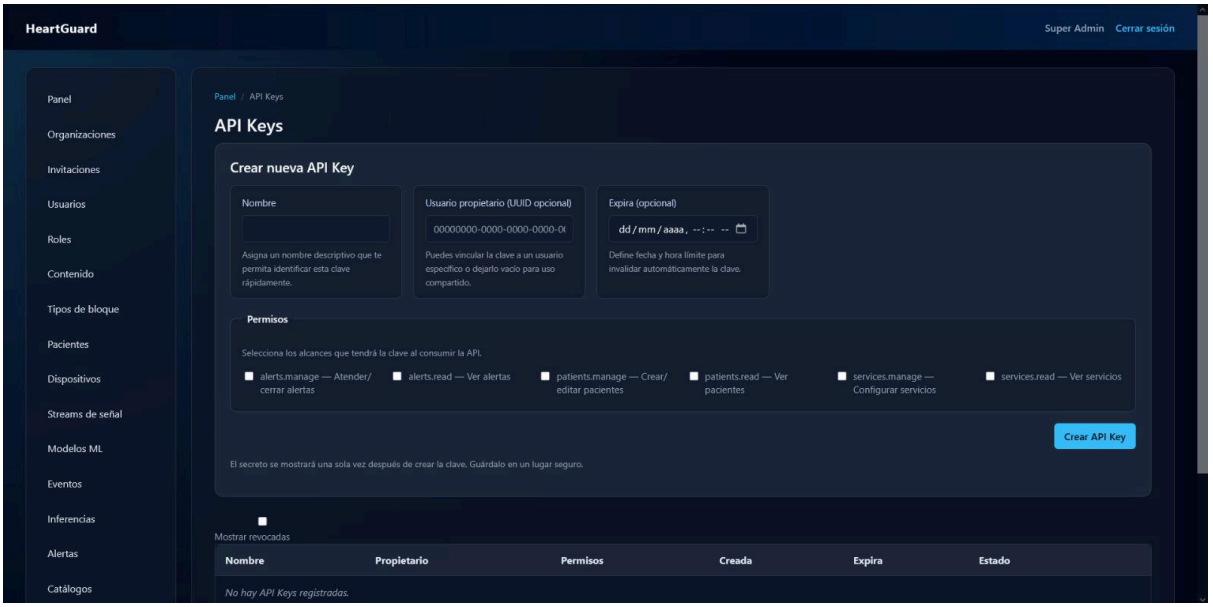

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Modelos ML
Modelos ML
Registrar modelo
Nombre
Predictor ritmo
Versión
1.0
Tarea
arrhythmia_detection
Dato de entrenamiento (ref)
s3://bucket/path
Hyperparámetros (JSON)
{"lr":0.01}
Crear modelo
Modelos registrados
Nombre
Versión
Tarea
Entrenamiento
Hyperparámetros
Creado
Sin modelos registrados.

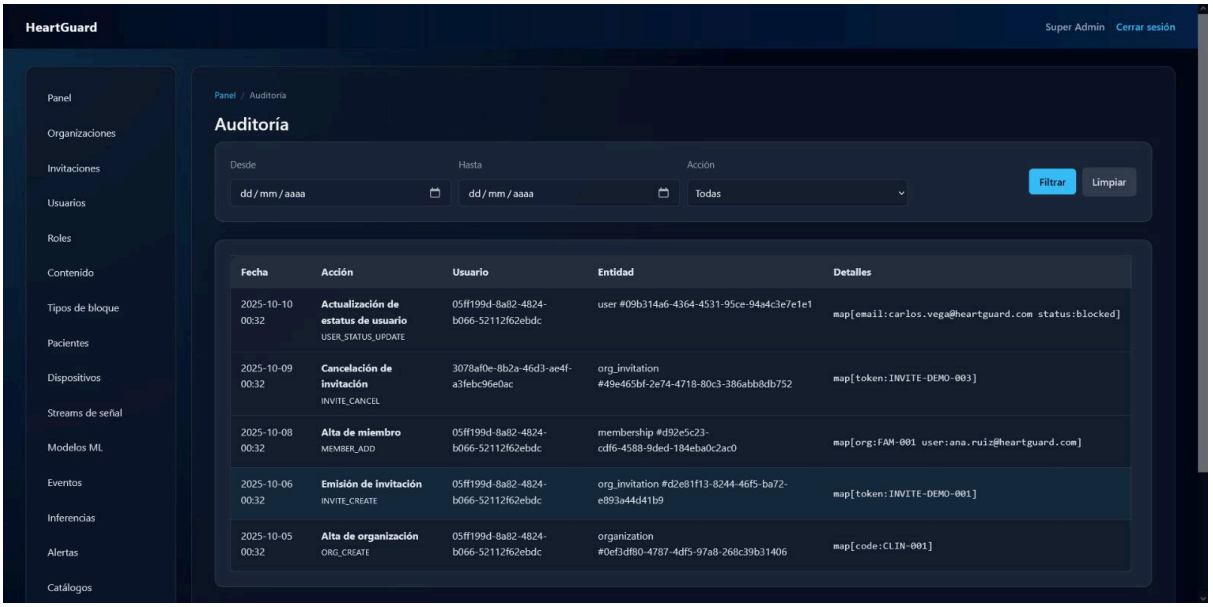

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Tipos de evento
Tipos de evento
Registrar tipo
Código
arrhythmia_detected
Descripción
Evento arrítmico detectado
Severidad por defecto
Selecciona nivel
Crear tipo
Tipos registrados
Código
Descripción
Severidad
AFIB
Fibrilación auricular
Alto
DESAT
Desaturación O2
Alto
HYPERTENSION
Hipertensión
Medio
TACHY
Taquicardia
Medio
Editar
Eliminar

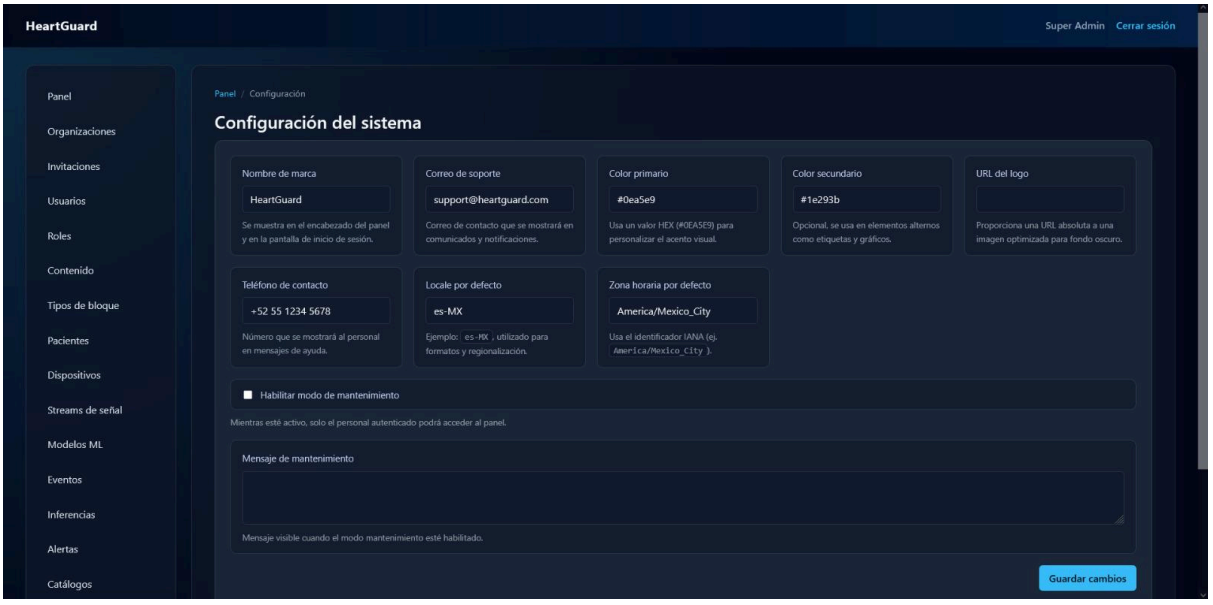

HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Inferencias
Inferencias
Registrar inferencia
Stream
Selecciona stream
Modelo
Sin modelo
Evento
Selecciona evento
Inicio ventana
dd / mm / aaaa, --:-- --
Fin ventana
dd / mm / aaaa, --:-- --
Score
0.87
Umbral
0.75
Serie ref
segment-123
Metadata (JSON)
Snapshot (JSON)
Crear inferencia
Inferencias registradas
Stream
Evento
Ventana
Score
Umbral
Modelo
Creado
Sin inferencias registradas.


HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Alertas
Alertas
Registrar alerta
Paciente
Selecciona paciente
Tipo
Selecciona tipo
Nivel
Selecciona nivel
Estatus
Selecciona estatus
Modelo
Sin modelo
Inferencia
Sin inferencia
Descripción
Ubicación (WKT)
POINT(-99.14 19.4)
Crear alerta
Alertas registradas
Paciente
Tipo
Nivel
Estatus
Organización
Creada
Sin alertas registradas.


HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Catálogos / Estatus de usuarios
Catálogos
Canales de alerta
Niveles de alerta
Estados de exportación
Categorías de contenido
Estatus de contenido
Tipos de contenido
Estados de entrega
Roles de organización
Plataformas
Estados de servicio
Sexos
Tipos de señal
Estatus de usuarios
Estatus de usuarios
Agregar elemento
Código
Identificador único del elemento dentro de este catálogo.
Nombre
Texto legible que se mostrará al resto del equipo.
Crear
Elementos
Código
Nombre
active
Activo
Editar
Eliminar
blocked
Bloqueado


HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / API Keys
API Keys
Crear nueva API Key
Nombre
Asigna un nombre descriptivo que te permita identificar esta clave rápidamente.
Usuario propietario (UUID opcional)
00000000-0000-0000-0000-0(
Puedes vincular la clave a un usuario específico o déjarlo vacío para uso compartido.
Expira (opcional)
dd / mm / aaaa, --:-- --
Define fecha y hora límite para invalidar automáticamente la clave.
Permisos
Selecciona los alcances que tendrá la clave al consumir la API.
alerts.manage — Atender/ cerrar alertas
alerts.read — Ver alertas
patients.manage — Crear/ editar pacientes
patients.read — Ver pacientes
services.manage — Configurar servicios
services.read — Ver servicios
Crear API Key
El secreto se mostrará una sola vez después de crear la clave. Guárdalo en un lugar seguro.
Mostrar revocadas
Nombre
Propietario
Permisos
Creada
Expira
Estado
No hay API Keys registradas.


HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Auditoría
Auditoría
Desde
dd / mm / aaaa
Hasta
dd / mm / aaaa
Acción
Todas
Filtrar
Limpiar
Fecha
Acción
Usuario
Entidad
Detalles
2025-10-10 00:32
Actualización de estatus de usuario
USER_STATUS_UPDATE
05ff199d-8a82-4824-b066-52112f62ebdc
user #09b314a6-4364-4531-95ce-94a4c3e7e1e1
map[email:carlos.vega@heartguard.com status:blocked]
2025-10-09 00:32
Cancelación de invitación
INVITE_CANCEL
3078af0e-8b2a-46d3-ae4f-a3febc96e0ac
org_invitation #49e465bf-2e74-4718-80c3-386abb8db752
map[token:INVITE-DEMO-003]
2025-10-08 00:32
Alta de miembro
MEMBER_ADD
05ff199d-8a82-4824-b066-52112f62ebdc
membership #d92e5c23-cdf6-4588-9ded-184eba0c2ac0
map[org:FAM-001 user:ana.ruiz@heartguard.com]
2025-10-06 00:32
Emisión de invitación
INVITE_CREATE
05ff199d-8a82-4824-b066-52112f62ebdc
org_invitation #d2e81f13-8244-46f5-ba72-e893a44d41b9
map[token:INVITE-DEMO-001]
2025-10-05 00:32
Alta de organización
ORG_CREATE
05ff199d-8a82-4824-b066-52112f62ebdc
organization #0ef3df80-4787-4df5-97a8-268c39b31406
map[code:CLIN-001]


HeartGuard
Super Admin
Cerrar sesión
Panel
Organizaciones
Invitaciones
Usuarios
Roles
Contenido
Tipos de bloque
Pacientes
Dispositivos
Streams de señal
Modelos ML
Eventos
Inferencias
Alertas
Catálogos
Panel / Configuración
Configuración del sistema
Nombre de marca
HeartGuard
Se muestra en el encabezado del panel y en la pantalla de inicio de sesión.
Correo de soporte
support@heartguard.com
Correo de contacto que se mostrará en comunicados y notificaciones.
Color primario
#0ea5e9
Usa un valor HEX (#0EA5E9) para personalizar el acento visual.
Color secundario
#1e293b
Opcional, se usa en elementos alternos como etiquetas y gráficos.
URL del logo
Proporciona una URL absoluta a una imagen optimizada para fondo oscuro.
Teléfono de contacto
+52 55 1234 5678
Número que se mostrará al personal en mensajes de ayuda.
Locale por defecto
es-MX
Ejemplo: es-MX , utilizado para formatos y regionalización.
Zona horaria por defecto
America/Mexico_City
Usa el identificador IANA (ej. America/Mexico_City).
Habilitar modo de mantenimiento
Mientras esté activo, solo el personal autenticado podrá acceder al panel.
Mensaje de mantenimiento
Mensaje visible cuando el modo mantenimiento esté habilitado.
Guardar cambios

HEARTGUARD - REPORTE EMPRESARIAL

HG

REPORTE EJECUTIVO

Panel de Administración SuperAdmin

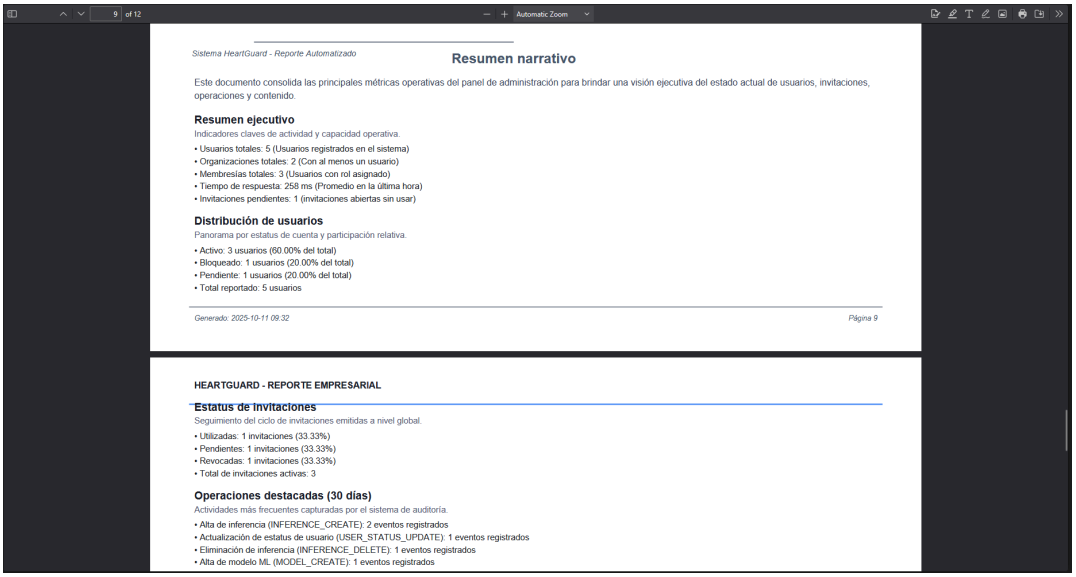
INFORMACIÓN DEL REPORTE

Fecha de generación: 2025-10-11 09:32

Sistema: HeartGuard SuperAdmin v1.0

Documento confidencial - Uso interno

Generado: 2025-10-11 09:32 *Página 1*



Sistema HeartGuard - Reporte Automatizado

Resumen narrativo

Este documento consolida las principales métricas operativas del panel de administración para brindar una visión ejecutiva del estado actual de usuarios, invitaciones, operaciones y contenido.

Resumen ejecutivo

Indicadores claves de actividad y capacidad operativa.

- Usuarios totales: 5 (Usuarios registrados en el sistema)
- Organizaciones totales: 2 (Con al menos un usuario)
- Membresías totales: 3 (Usuarios con rol asignado)
- Tiempo de respuesta: 258 ms (Promedio en la última hora)
- Invitaciones pendientes: 1 (invitaciones abiertas sin usar)

Distribución de usuarios

Panorama por estatus de cuenta y participación relativa.

- Activo: 3 usuarios (60.00% del total)
- Bloqueado: 1 usuarios (20.00% del total)
- Pendiente: 1 usuarios (20.00% del total)
- Total reportado: 5 usuarios

Generado: 2025-10-11 09:32 *Página 9*

HEARTGUARD - REPORTE EMPRESARIAL

Estatus de invitaciones

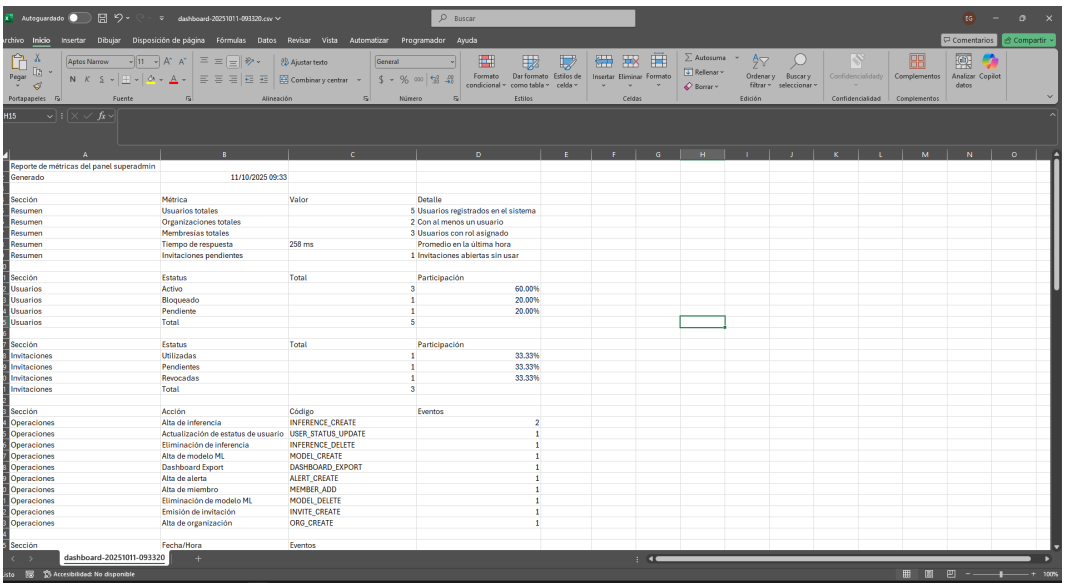
Seguimiento del ciclo de invitaciones emitidas a nivel global.

- Utilizadas: 1 invitaciones (33.33%)
- Pendientes: 1 invitaciones (33.33%)
- Revocadas: 1 invitaciones (33.33%)
- Total de invitaciones activas: 3

Operaciones destacadas (30 días)

Actividades más frecuentes capturadas por el sistema de auditoría.

- Alta de inferencia (INFERENCE_CREATE): 2 eventos registrados
- Actualización de estatus de usuario (USER_STATUS_UPDATE): 1 eventos registrados
- Eliminación de inferencia (INFERENCE_DELETE): 1 eventos registrados
- Alta de modelo ML (MODEL_CREATE): 1 eventos registrados



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
Reporte de métricas del panel superadmin															
Generado	11/10/2025 09:33														
Sección	Métrica	Valor	Detalle												
Resumen	Usuarios totales		5 Usuarios registrados en el sistema												
Resumen	Organizaciones totales		2 Con al menos un usuario												
Resumen	Membresías totales		3 Usuarios con rol asignado												
Resumen	Tiempo de respuesta	258 ms	Promedio en la última hora												
Resumen	Invitaciones pendientes		1 Invitaciones abiertas sin usar												
Sección	Estatus	Total	Participación												
Usuarios	Activo		3 60.00%												
Usuarios	Bloqueado		1 20.00%												
Usuarios	Pendiente		1 20.00%												
Usuarios	Total		5												
Sección	Estatus	Total	Participación												
Invitaciones	Utilizadas		1 33.33%												
Invitaciones	Pendientes		1 33.33%												
Invitaciones	Revocadas		1 33.33%												
Invitaciones	Total		3												
Sección	Acción	Código	Eventos												
Operaciones	Alta de inferencia	INFERENCE_CREATE	2												
Operaciones	Actualización de estatus de usuario	USER_STATUS_UPDATE	1												
Operaciones	Eliminación de inferencia	INFERENCE_DELETE	1												
Operaciones	Alta de modelo ML	MODEL_CREATE	1												
Operaciones	Dashboard Export	DASHBOARD_EXPORT	1												
Operaciones	Alta de alerta	ALERT_CREATE	1												
Operaciones	Alta de miembro	MEMBER_ADD	1												
Operaciones	Eliminación de modelo ML	MODEL_DELETE	1												
Operaciones	Emisión de invitación	INVITE_CREATE	1												
Operaciones	Alta de organización	ORG_CREATE	1												
Sección	Fecha/Hora	Eventos													

9. Conclusiones sobre el desarrollo, retos enfrentados y lecciones aprendidas

Cumplimiento de Objetivos y Arquitectura Robusta

El proyecto cumplió exitosamente con el objetivo principal de desarrollar un sistema de gestión de contenidos (CMS) backend completamente funcional. Se implementaron todas las características críticas especificadas, incluyendo operaciones CRUD para las entidades principales, un sistema de autenticación seguro basado en roles mediante tokens JWT, y la capacidad de generar reportes y visualizaciones de datos analíticos. La entrega de estas funcionalidades demuestra la capacidad del equipo para traducir requerimientos complejos en una solución técnica coherente y operativa.

La arquitectura seleccionada, fundamentada en una base de datos normalizada hasta la tercera forma normal (3FN), demostró ser una decisión acertada que garantiza la integridad de los datos y la escalabilidad del sistema. El uso intensivo de Stored Procedures para encapsular la lógica de negocio directamente en la base de datos no solo optimizó el rendimiento de las consultas complejas, sino que también simplificó la capa de aplicación. Esta estrategia permitió que el backend se centrará en la gestión de rutas y la comunicación, delegando las operaciones de datos intensivas al motor de la base de datos.

Como resultado, el sistema posee una base sólida que facilita futuras expansiones. La clara separación entre la lógica de la aplicación y la lógica de datos permite que nuevos módulos o funcionalidades se integren con un mínimo de fricción. Esta modularidad es una de las lecciones más importantes aprendidas, ya que asegura que el CMS pueda evolucionar para satisfacer nuevas necesidades del negocio sin requerir una reestructuración fundamental.

Retos Técnicos y Superación de Obstáculos

Uno de los desafíos técnicos más significativos fue la implementación del control de acceso basado en roles (RBAC). Asegurar que cada endpoint estuviera protegido adecuadamente y que los distintos roles de usuario (administrador, editor,

etc.) tuvieran los permisos correctos representó una tarea compleja. Esto requirió un diseño cuidadoso de middlewares de autorización y pruebas exhaustivas para prevenir cualquier posible brecha de seguridad que permitiera el acceso no autorizado a datos o funcionalidades críticas.

El desarrollo y la optimización de las consultas SQL para el dashboard de analíticas también presentaron un reto considerable. Agregar grandes volúmenes de datos en tiempo real para generar las gráficas de KPIs exigió un análisis profundo de los planes de ejecución de las consultas y la implementación de estrategias de indexación adecuadas. El equipo dedicó tiempo a depurar Stored Procedures para asegurar que los cálculos fueran no solo correctos, sino también eficientes y que no degradará el rendimiento general de la plataforma.

La superación de estos obstáculos reforzó la importancia de las pruebas unitarias y de integración, especialmente en los módulos de seguridad. La lección aprendida clave fue la necesidad de anticipar los cuellos de botella de rendimiento y diseñar las consultas de la base de datos pensando en la escalabilidad desde el principio. La experiencia también subrayó el valor de las revisiones de código en equipo para identificar errores lógicos y de seguridad de manera temprana.

Colaboración Efectiva y Lecciones de Gestión de Proyectos

La colaboración fue un pilar fundamental para el éxito del módulo. El uso de un sistema de control de versiones como Git fue indispensable para gestionar las contribuciones del equipo, coordinar cambios y resolver conflictos de manera ordenada. Establecer un flujo de trabajo claro, con ramas de desarrollo por funcionalidad y revisiones de pull requests, aseguró la calidad y la coherencia del código base a lo largo del proyecto.

Las herramientas seleccionadas jugaron un papel crucial en la productividad del equipo. La combinación de un framework de backend moderno con un sistema de gestión de bases de datos robusto proporcionó un entorno de desarrollo eficiente. Asimismo, el uso de herramientas como Postman para la documentación y prueba de la API permitió a los miembros del equipo trabajar en paralelo, validando

el comportamiento de los endpoints de forma independiente antes de la integración final.

La principal lección aprendida desde la perspectiva de la gestión fue la importancia de una planificación inicial detallada, especialmente en lo que respecta al diseño del esquema de la base de datos y la definición de los contratos de la API. Mantener una comunicación constante y realizar ciclos de retroalimentación cortos permitió al equipo adaptarse a los desafíos imprevistos y encontrar soluciones de manera colectiva. Este enfoque ágil no solo mejoró el producto final, sino que también fortaleció las habilidades de colaboración y resolución de problemas del equipo.