



## INDEX

<b>1. RESUMEN EJECUTIVO.....</b>	<b>3</b>
<b>2. CONTEXTO Y MOTIVACIÓN .....</b>	<b>4</b>
<b>3. OBJETIVOS DEL PROYECTO .....</b>	<b>5</b>
Objetivo general.....	5
Objetivos específicos .....	5
<b>4. METODOLOGÍA DEL TRABAJO .....</b>	<b>6</b>
Roles de equipo .....	6
Adaptación de la metodología ágil Scrum.....	7
<b>5. FLUJO DE PROCESAMIENTO Y ARQUITECTURA TÉCNICA .....</b>	<b>8</b>
Flujo general del sistema .....	8
Tecnologías utilizadas .....	8
Infraestructura y arquitectura del sistema.....	9
Escalabilidad y Reproducibilidad .....	10
<b>6. FUNCIONALIDADES DEL SISTEMA.....</b>	<b>11</b>
Funcionalidades generales del frontend.....	11
Funcionalidades generales del backend.....	12
Funcionalidades de accesibilidad.....	13
<b>7. RESULTADOS Y DEMOSTRACIÓN .....</b>	<b>13</b>
Página de inicio .....	13
Herramientas desarrolladas .....	13
<b>8. IMPACTO Y RELEVANCIA .....</b>	<b>13</b>
Impacto social y global.....	13
<b>9. FUTURO Y PRÓXIMOS PASOS.....</b>	<b>13</b>





## 1. Resumen Ejecutivo

El proyecto **Insight**, desarrollado por el equipo *Team Bingo* para abordar el reto de análisis inteligente de datos clínicos, propone una nueva forma de explorar y comprender la información médica relacionada con pacientes con enfermedades mentales.

A lo largo de los años, se han generado miles de registros y bases de datos con información clínica y demográfica, pero su volumen, complejidad y dispersión dificultan su acceso y análisis eficiente. Nuestro objetivo es transformar este conjunto de datos en una base de conocimiento dinámica, interactiva y accesible,.

Para lograrlo, **Insight** combina técnicas avanzadas de análisis de datos con herramientas visuales que permiten **filtrar información relevante y representarla gráficamente** de manera intuitiva. El sistema permite seleccionar atributos clínicos o demográficos —como comunidad autónoma, edad, sexo o categoría diagnóstica— y analizarlos mediante gráficos y filtros personalizados que facilitan la extracción de patrones, tendencias y correlaciones.

La aplicación web integra dos funcionalidades principales:

- **Filtrado de datos inteligente**, que permite a los usuarios aplicar criterios específicos y obtener resultados de forma rápida y visual.
- **Representación gráfica dinámica**, que traduce los datos filtrados en visualizaciones comprensibles y comparativas.

Estas herramientas están diseñadas para distintos perfiles de usuario dentro del ámbito sanitario:

- **Profesionales clínicos**: pueden identificar patrones de distribución de diagnósticos y características de pacientes sin necesidad de conocimientos técnicos avanzados.
- **Investigadores**: pueden explorar relaciones entre variables clínicas y sociodemográficas para apoyar estudios y publicaciones.
- **Gestores sanitarios**: pueden detectar tendencias y áreas prioritarias para la planificación y optimización de recursos en salud mental.

Con esta plataforma, aspiramos a mejorar la comprensión de los datos clínicos, favorecer la investigación en salud mental y democratizar el acceso a la información médica, promoviendo un enfoque transparente, ético y basado en evidencia.

De este modo, **Insight combina el rigor científico con la innovación tecnológica**, contribuyendo a una nueva etapa de investigación interdisciplinar guiada por la inteligencia artificial.

Finalmente, cabe destacar que **Insight no es una demostración conceptual ni un conjunto de datos sintéticos**, sino una herramienta funcional que trabaja con **datos reales**. Este enfoque representa un logro técnico significativo y un paso adelante hacia el análisis inteligente y responsable de datos clínicos.



## 2. Contexto y Motivación

El reto que aborda **Insight** nace de una necesidad urgente dentro del ámbito sanitario: **aprovechar de forma inteligente los datos médicos generados a lo largo de años de investigación y atención clínica en salud mental.**

Cada hospital, centro de salud y organismo público produce enormes volúmenes de información sobre pacientes, diagnósticos y tratamientos. Sin embargo, **estos datos suelen estar fragmentados en múltiples fuentes**, almacenados en formatos heterogéneos y rara vez se integran para extraer conocimiento útil. Esta dispersión limita el potencial de los profesionales para identificar patrones, mejorar la toma de decisiones clínicas o desarrollar políticas basadas en evidencia.

El equipo **Bingo Malackaton** reconoce que uno de los mayores desafíos para la comunidad médica, investigadora y de gestión sanitaria es precisamente **el acceso eficiente y significativo a la información clínica sobre salud mental.**

Aunque muchos de estos datos son accesibles o forman parte de registros oficiales, **la falta de herramientas interactivas de análisis y visualización impide su explotación real.** Como resultado, se desaprovechan oportunidades de descubrimiento, se repiten estudios y se dificulta la detección temprana de tendencias o desigualdades entre regiones y grupos de pacientes.

En este contexto surge **Insight**: una **plataforma web interactiva impulsada por inteligencia artificial** que permite organizar, filtrar y visualizar datos médicos de pacientes con enfermedades mentales de forma dinámica e intuitiva.

La herramienta combina **filtrado inteligente y representación gráfica avanzada** para conectar variables clínicas y sociodemográficas, facilitando una comprensión más profunda de la distribución, evolución y características de las patologías mentales en distintas poblaciones.

A través de esta iniciativa, **Insight** busca no solo **mejorar el acceso y la comprensión de los datos clínicos**, sino también **favorecer la toma de decisiones informadas** en ámbitos clave de la salud mental: desde la planificación de recursos sanitarios hasta la orientación de líneas de investigación y estrategias de prevención.

En última instancia, el proyecto aspira a **transformar la manera en que se descubren, interpretan y aplican los conocimientos sobre salud mental**, contribuyendo al objetivo global de **mejorar la calidad de vida y la atención de las personas que viven con trastornos mentales.**



### 3. Objetivos del Proyecto

#### Objetivo general

El objetivo general de **Insight** es desarrollar una **plataforma web inteligente** que permita analizar, filtrar y visualizar datos médicos de pacientes con enfermedades mentales, facilitando la comprensión de patrones clínicos, demográficos y territoriales. A través de herramientas interactivas de **filtrado de datos** y **representación gráfica dinámica**, la plataforma busca **democratizar el acceso a la información médica**, **mejorar la toma de decisiones en salud mental** y **apoyar la investigación científica**, integrando innovación tecnológica e inteligencia artificial para transformar datos dispersos en conocimiento útil y accionable.

#### Objetivos específicos

- **Diseñar y desplegar una base de datos estructurada y anonimizada** en la infraestructura de Oracle Cloud (OCI), que almacene los datos reales de hospitalización por causas de salud mental, garantizando su normalización, seguridad y accesibilidad.
- **Implementar una aplicación web interactiva** que permita conectar con la base de datos, **filtrar información relevante** y **representarla gráficamente** mediante visualizaciones claras y personalizables, facilitando la exploración intuitiva por parte de investigadores y profesionales sanitarios.
- **Realizar un análisis exploratorio de datos (EDA)** que identifique patrones, distribuciones y correlaciones significativas entre variables demográficas, clínicas y territoriales, sirviendo como base para futuras investigaciones en salud mental.
- **Integrar principios de ingeniería de software** en el desarrollo del sistema, garantizando la modularidad, escalabilidad y mantenimiento del código, además de una interfaz accesible y coherente con las buenas prácticas de diseño.
- **Aplicar técnicas de inteligencia artificial y análisis estadístico** para enriquecer la interpretación de los datos y ofrecer una visión más profunda sobre la distribución y evolución de los diagnósticos de salud mental en distintas comunidades.
- **Promover el uso ético y socialmente responsable de los datos sanitarios**, fomentando la transparencia, la protección de la privacidad y el acceso democratizado al conocimiento médico.



## 4. Metodología del Trabajo

### Roles de equipo

#### **Fernando Luis Pinilla Molina - Tech Lead & Arquitecto de Software**

Como Tech Lead y Arquitecto de Software, Fernando se encarga de sentar las bases técnicas del proyecto. Su rol principal consiste en diseñar la arquitectura global del sistema, seleccionando las tecnologías más adecuadas y definiendo los patrones de diseño que seguirá todo el equipo. Se ocupa de resolver los problemas técnicos más complejos y garantiza que el código mantenga unos estándares de calidad elevados. Además, actúa como mentor para el resto del equipo, compartiendo su conocimiento y experiencia para elevar las capacidades técnicas de todos los miembros. Su visión arquitectónica es crucial para construir una aplicación escalable y mantenible a largo plazo.

#### **Carlos Garcia Guzman - Jefe de Desarrollo & DevOps Lead**

Carlos asume el rol de Jefe de Desarrollo y DevOps Lead, coordinando las actividades diarias del equipo de programación. Su responsabilidad principal es asegurar que el flujo de desarrollo sea ágil y eficiente, gestionando la integración continua y los procesos de despliegue. Se encarga de supervisar la calidad del código producido y optimizar los pipelines de desarrollo para que el equipo pueda trabajar de manera productiva. Además, monitoriza el rendimiento de la aplicación en producción y se asegura de que la infraestructura técnica soporte adecuadamente las necesidades del proyecto. Su labor es fundamental para mantener un ritmo de desarrollo constante y de calidad.

#### **Jose Carlos Durán Nuño - Jefe de Experiencia de Usuario (UX/UI Lead)**

José Carlos lidera el área de Experiencia de Usuario como UX/UI Lead, siendo el responsable de que la aplicación sea intuitiva, atractiva y fácil de usar. Su trabajo comienza con la investigación de usuarios y la creación de wireframes y prototipos que definen la interacción persona-máquina. Diseña interfaces coherentes y visualmente atractivas, estableciendo guías de estilo que todo el equipo seguirá. Colabora estrechamente con los desarrolladores para asegurar que la implementación final se ajusta fielmente a los diseños propuestos. Su enfoque en el usuario garantiza que la aplicación no solo sea técnicamente sólida, sino también agradable y eficiente para quienes la utilizan.

#### **Candela Davila Moreno - Directora de Producto (CPO) & Scrum Master**

Candela combina el rol de Directora de Producto (CPO) con las funciones de Scrum Master, actuando como puente entre la visión del producto y la ejecución técnica. Como CPO, define la visión global del producto, prioriza las funcionalidades según su valor para el usuario y se asegura de que cada entrega aporte valor real. Como Scrum Master, facilita las ceremonias ágiles, elimina impedimentos que puedan ralentizar al equipo y



garantiza que se siguen las metodologías ágiles correctamente. Su labor es mantener al equipo enfocado en los objetivos del producto mientras vela por un proceso de desarrollo fluido y bien organizado.

### Colaboración y Trabajo en Equipo

La fortaleza de este equipo reside en su versatilidad como desarrolladores Full Stack, lo que permite una colaboración excepcionalmente fluida. Fernando proporciona la base técnica sólida sobre la que Carlos construye procesos de desarrollo eficientes. José Carlos transforma los requisitos técnicos en experiencias de usuario excepcionales, mientras Candela asegura que todo el trabajo esté alineado con los objetivos del producto.

La superposición de habilidades Full Stack significa que cada miembro comprende los desafíos y necesidades de las demás áreas, facilitando la comunicación y la resolución colaborativa de problemas. Esta estructura permite que el equipo sea ágil y adaptable, capaz de redistribuir tareas según las necesidades del proyecto en cada momento, manteniendo siempre la calidad y coherencia en el desarrollo.

El enfoque colaborativo asegura que las decisiones técnicas, de diseño y de producto se toman de manera consensuada, aprovechando la perspectiva única que cada rol aporta al proyecto. Esta sinergia entre especialización y versatilidad es la clave para entregar un producto final que sea técnicamente robusto, visualmente atractivo y centrado en las necesidades reales del usuario.

### Adaptación de la metodología ágil Scrum

En **Team Bingo**, implementamos una **metodología ágil basada en Scrum**, cuidadosamente adaptada al contexto intensivo del hackathon. Dado el tiempo limitado, optamos por una **versión Express del marco**, estructurando el desarrollo en **tres sprints de dos horas** cada uno. Este formato permitió mantener un ritmo de trabajo dinámico, fomentar la colaboración continua y garantizar la toma de decisiones rápida y coordinada entre los distintos roles del equipo.

Cada sprint comenzó con una **breve reunión de planificación**, en la que se definieron las tareas prioritarias, y concluyó con una **revisión del progreso y los posibles bloqueos**. La gestión de las tareas se realizó mediante un **tablero Kanban físico**, organizado en tres columnas: “*Backlog*”, “*In Progress*” y “*Done*”. Este enfoque visual facilitó la distribución equilibrada del trabajo, la monitorización del avance y la sincronización entre los ámbitos de diseño, desarrollo e inteligencia artificial.

Gracias a esta metodología ágil, el equipo logró **iterar con rapidez, mantener una visión común del producto y asegurar la entrega continua de valor** a lo largo de todas las fases de desarrollo de **Team Bingo**.

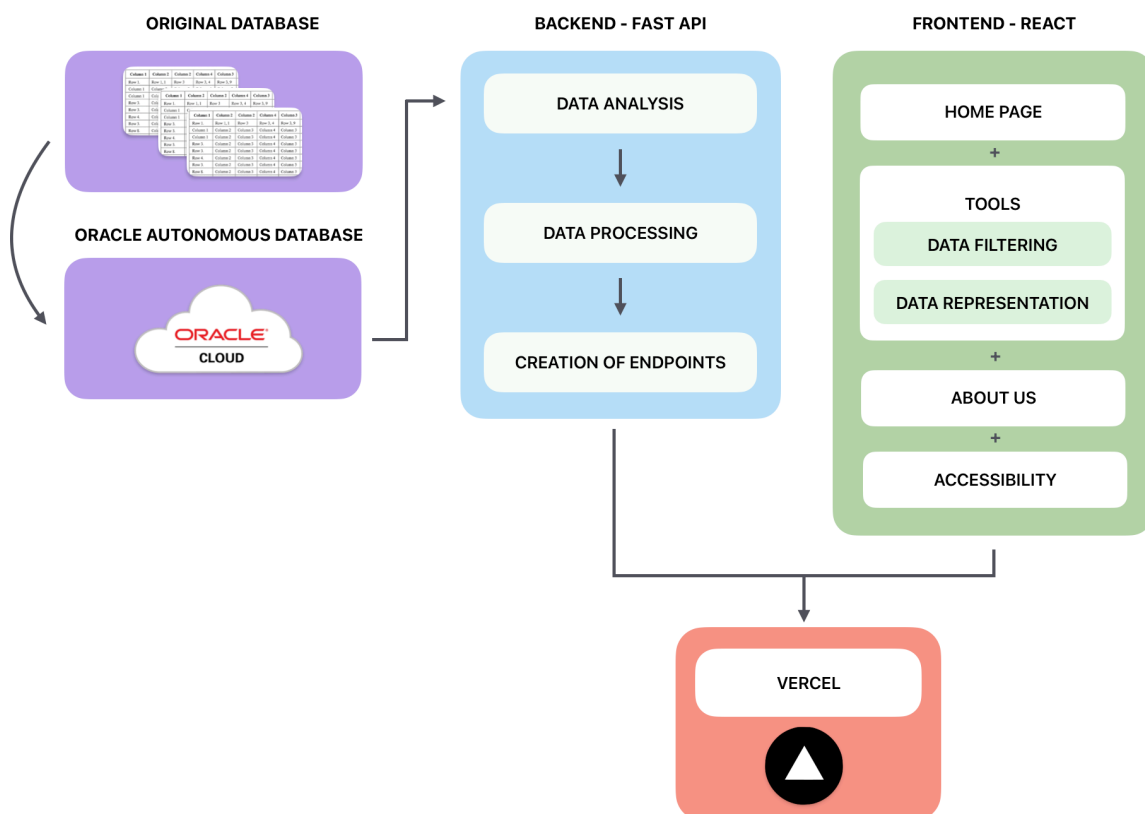


A continuación, se presenta una **secuencia de tres imágenes** que muestra la **evolución progresiva del tablero Kanban físico**, construido en el espacio de trabajo. Cada imagen refleja el estado del proyecto al final de cada sprint de dos horas, evidenciando el flujo de trabajo, la priorización y la coordinación efectiva del equipo.

## 5. Flujo de Procesamiento y Arquitectura Técnica

### Flujo general del sistema

El pipeline de nuestro sistema está dividido en 5 fases



*Vista del diagrama de flujo de trabajo del sistema*

### Tecnologías utilizadas

#### Frontend (Cliente)

- **React 18:** Framework principal para construir una interfaz web dinámica, modular y responsive.





- **Vite:** Herramienta de desarrollo y compilación rápida que mejora el rendimiento y los tiempos de carga.
- **CSS3:** Estilos personalizados con animaciones y degradados que mejoran la experiencia visual del usuario.
- **React Router DOM:** Sistema de navegación entre páginas dentro de la aplicación.
- **JavaScript ES6+:** Lenguaje de programación moderno utilizado para la lógica del cliente y la comunicación con la API.
- **Vercel:** Plataforma utilizada para el despliegue del frontend, con integración continua y alta velocidad de carga.

### Backend (Servidor)

- **Python 3.x:** Lenguaje principal utilizado para la lógica del servidor y el procesamiento de datos.
- **FastAPI:** Framework moderno para crear APIs REST rápidas y eficientes.
- **Uvicorn:** Servidor ASGI que ejecuta la aplicación FastAPI y gestiona las peticiones concurrentes.
- **Pydantic:** Validación y serialización de datos dentro del backend.
- **cx\_Oracle:** Conector entre Python y la base de datos Oracle.
- **python-dotenv:** Gestión segura de variables de entorno.
- **Base de Datos**
- **Oracle Database:** Sistema de gestión de base de datos relacional que almacena los datos anonimizados de pacientes.
- **SQL:** Lenguaje de consultas empleado para el filtrado y la obtención dinámica de información.

### Infraestructura y arquitectura del sistema

El sistema **Insight** se organiza bajo una **arquitectura distribuida cliente-servidor**, donde el **backend (FastAPI)** actúa como punto de acceso a los datos clínicos almacenados en **Oracle Database**, y el **frontend (React)** consume las APIs para mostrar la información de forma dinámica e interactiva.

- El **backend** expone endpoints REST que permiten **filtrar, consultar y procesar datos** médicos, garantizando un acceso eficiente y seguro a la base de datos.
- El **frontend**, desplegado en **Vercel**, se comunica con el servidor para enviar las solicitudes de los usuarios y **representar los resultados visualmente** mediante gráficos y tablas interactivas.

A partir de la base de datos inicial, se han identificado dependencias funcionales. Mediante la aplicación sistemática de las formas normales, se ha obtenido un conjunto de tablas normalizadas que garantizan la consistencia, minimizan la duplicidad de la información y facilitan la integridad referencial. El modelo que se expone a continuación refleja la organización definitiva de las entidades y sus relaciones, constituyendo la base para un diseño lógico más eficiente y escalable:

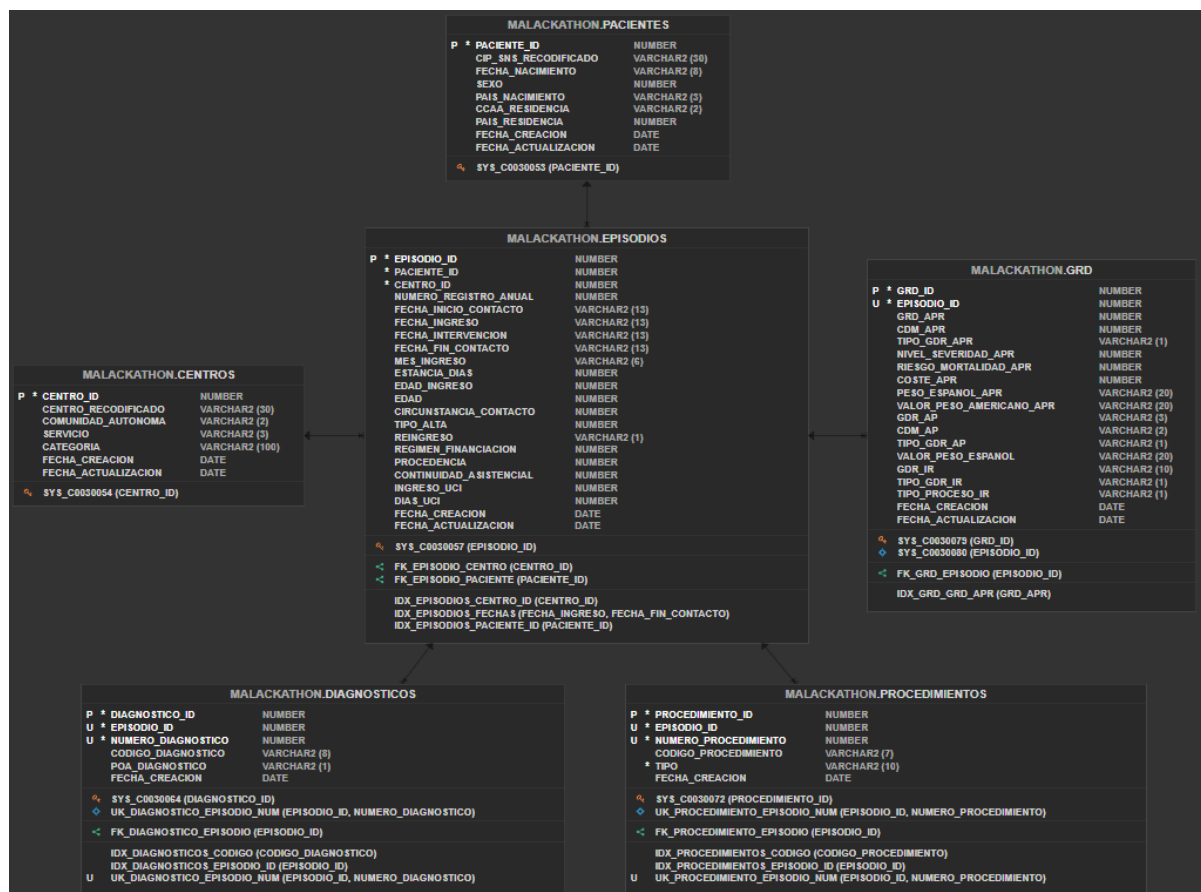


Diagrama de tablas normalizadas

## Escalabilidad y Reproducibilidad

El sistema **Insight** ha sido diseñado bajo **principios de escalabilidad horizontal**, permitiendo ampliar el volumen de datos médicos procesados sin comprometer el rendimiento ni la estabilidad del servicio.

El uso de **formatos y tecnologías estándar** —como **JSON, SQL, FastAPI y React**— garantiza la **portabilidad del proyecto** y la **reproducibilidad científica** de los resultados obtenidos.

Además, su **arquitectura modular** facilita la integración futura con otras fuentes de datos sanitarios o sistemas de análisis, asegurando la **evolución continua** de la plataforma y su adaptación a nuevos entornos de investigación clínica.



## 6. Funcionalidades del Sistema

El proyecto **Insight** está diseñado para ofrecer una **experiencia de exploración de datos médica accesible, visual e inteligente**. Su **arquitectura modular** permite que distintos perfiles de usuario —desde **investigadores y profesionales sanitarios hasta gestores de salud pública**— interactúen con la misma base de datos desde **enfoques adaptados a sus necesidades específicas**, accediendo a funcionalidades personalizadas para el análisis, filtrado y representación gráfica de la información.

### Funcionalidades generales del frontend

- **Landing / Launching page**
  - Iconos representativos para cada herramienta (Filtrado, Visualización, Chatbot).
  - Cada icono es clickeable y navega a la ruta correspondiente; al navegar hace scroll al inicio de la página de destino.
  - Visual: rectángulos alrededor de iconos, efectos hover y active, orden configurable (por ejemplo lupa, gráfico de barras, robot).
- **Página de Filtrado (DataFilteringPage.jsx)**
  - **Interfaz de filtros compuesta por:**
    - Comunidades Autónomas (lista de checkboxes, dinámicamente cargada si el backend lo proporciona).
    - Año de nacimiento (slider de rango; rango actual 1926–2025).
    - Sexo (checkable, valores dinámicos por backend o fallback: Hombre, Mujer, Otros).
    - Diagnósticos (lista de checkboxes con descripciones completas).
    - Centros (lista desplegable/grid con scroll, muestra los primeros 20 y un aviso si hay más).
  - **Aplicar filtros:**
    - Botón "Filtrar" que envía los filtros seleccionados al backend y recarga la tabla.
    - Soporta combinaciones múltiples de filtros (todos excepto el intervalo de año pueden ser múltiples selecciones).
  - **Tabla de resultados:**
    - Columnas: ID, Nombre completo, Comunidad Autónoma, Año de Nacimiento, Sexo, Centro Médico, Fecha de Ingreso, Fecha de Fin de Contacto, Estancia (días), Diagnóstico completo.
    - Las fechas se muestran en formato DD/MM/YYYY (función `formatDate` que maneja distintos formatos entrantes).
    - Tooltips (title) en celdas con texto largo (nombre, centro, diagnóstico).
    - Datos completos (sin truncado forzado en la vista).
  - **Paginación:**
    - Navegación anterior/siguiente; filas por página configuradas (ej. 20).



- El backend devuelve total\_records, total\_pages y current\_page para sincronizar la UI.
- **Estados UI:**
  - Loading (indicador cuando se consultan datos).
  - Error (mensaje visual con detalle y fallback a datos mock).
  - Fallback: si la consulta al backend falla, se generan datos mock para desarrollo y pruebas.
- **Lógica cliente:**
  - loadFilterOptions() obtiene opciones dinámicas del backend (/api/filter-options).
  - loadPatients() hace POST a /api/filter-patients con filtro serializado (comunidades, año\_nacimiento\_min/max, sexo, diagnosticos, centros, page, rows\_per\_page).
  - Formateo y normalización de datos antes de mostrar.
  - Estado gestionado con useState / useEffect.

## Funcionalidades generales del backend

- **API REST (FastAPI)**
  - **Endpoint GET /api/filter-options**
    - Devuelve valores únicos / rangos para poblacion UI: comunidades, sexos, diagnósticos, centros y rango de año de nacimiento.
    - Permite que la UI cargue listas dinámicas y el rango de años.
  - **Endpoint POST /api/filter-patients**
    - Acepta payload con filtros:
      - comunidades: array
      - año\_nacimiento\_min: int
      - año\_nacimiento\_max: int
      - sexo: array
      - diagnosticos: array
      - centros: array
      - page, rows\_per\_page
    - Construye consulta dinámica sobre la tabla DATOS\_ORIGINALES (solo esa tabla).
    - Realiza paginación en la consulta (offset/limit) y devuelve:
      - data: lista de registros normalizados
      - total\_records, total\_pages, current\_page
    - Manejo de errores y respuestas HTTP adecuadas (status codes, mensajes).
- **Conexión a Oracle**
  - Uso de cx\_Oracle (o similar) para ejecutar consultas SQL contra la base Oracle.
  - Escapa/parametriza valores para evitar inyección SQL.
  - Transformación de tipos de fecha y strings a formatos que el frontend espera.
- **Modelado y validación**



- Pydantic para definir esquemas de entrada y salida (validación del payload).
- **Configuración**
  - Variables de entorno para credenciales de DB, host, puerto y CORS.
  - CORS habilitado para permitir peticiones desde el cliente (ej. puerto 3000).

## Funcionalidades de accesibilidad

Texto normal

## 7. Resultados y Demostración

### Página de inicio

Texto normal

### Herramientas desarrolladas

Texto normal

## 8. Impacto y Relevancia

### Impacto social y global

Texto normal

## 9. Futuro y Próximos Pasos

Texto normal

## 10. Descripción de los hitos

### Hito 1

En el marco del primer hito del proyecto, se ha llevado a cabo la **creación de un repositorio en la infraestructura Oracle Cloud Infrastructure (OCI) y el despliegue de una base de datos Autonomous Database 23ai**, destinada a la gestión y análisis de los datos proporcionados en el reto.

Dentro de la base de datos se han **creado las tablas principales** que conforman la estructura de información del sistema, definiendo sus relaciones y claves correspondientes. Aunque en esta fase inicial las tablas no contienen registros cargados, su diseño permite soportar de forma coherente el conjunto de datos del proyecto y facilitar su posterior integración con los módulos del backend y el visualizador.



Para garantizar la **protección de la información sensible**, se ha realizado un proceso de **anonimización de los datos mediante sustitución basada en iniciales**, preservando la coherencia y la capacidad de análisis del conjunto sin comprometer la identidad de las personas representadas. Este enfoque asegura que los datos mantengan utilidad estadística y trazabilidad parcial, cumpliendo a la vez los principios de privacidad.

Se ha creado un **usuario específico de acceso a la base de datos**, con las siguientes credenciales:

Usuario: malackathon

Contraseña: Oci.2025\_v4m0ssss

Este usuario dispone de los **permisos necesarios para consultar y explorar la estructura completa** de la base de datos, incluyendo la visualización de las tablas, sus relaciones y el contenido de las vistas.

El **modelado relacional de las tablas** se ha documentado mediante un **diagrama relacional**, en el que se reflejan las entidades definidas y las relaciones entre ellas. Este diagrama facilita la comprensión de la arquitectura de la base de datos y su correspondencia con los datos del reto.

Además, se ha desarrollado una vista denominada **VISTA\_MUY\_INTERESANTE**, que muestra información analítica relevante derivada del modelo de datos. Esta vista resume aspectos significativos del conjunto de datos, como distribuciones y patrones observables, constituyendo un primer punto de acceso a resultados de valor en el contexto del proyecto.

## Hito 2

En el segundo hito se ha desarrollado una **aplicación web interactiva** que se conecta con la base de datos desplegada en el Hito 1, permitiendo la **consulta, filtrado y visualización de los datos anonimizados**.

El objetivo principal de esta fase ha sido ofrecer una interfaz intuitiva que facilite la exploración visual de la información y la identificación de patrones relevantes en el conjunto de datos.

## Tecnologías utilizadas

La aplicación se ha construido bajo una **arquitectura cliente-servidor distribuida**, empleando las siguientes tecnologías principales:

**Frontend:** desarrollado con **React**, proporcionando una interfaz dinámica e interactiva.

**Backend:** implementado con **FastAPI**, encargado de gestionar las peticiones y el acceso a la base de datos.

**Base de datos:** **Oracle Autonomous Database 23ai**, desplegada en la infraestructura **OCI**.

**Comunicación cliente-servidor:** mediante **endpoints RESTful**, que permiten el intercambio de datos filtrados y agregados entre ambos componentes.



**Control de versiones y despliegue:** el proyecto se gestiona mediante un **repositorio en GitHub**, desde el cual la aplicación ha sido **desplegada utilizando la plataforma Vercel**, garantizando una integración continua y un despliegue automatizado del frontend.

### Funcionalidades principales

La aplicación web integra dos herramientas principales orientadas al análisis visual de los datos:

#### Filtrado de datos

Permite al usuario aplicar filtros personalizados sobre los registros de la base de datos en función de los siguientes campos:

#### Diagnóstico

**Año de nacimiento** (representado con las dos últimas cifras, conforme al formato original del dataset)

#### Sexo

#### Categoría diagnóstica

Una vez seleccionados los filtros, los resultados se actualizan dinámicamente en una tabla que muestra los datos filtrados, facilitando su exploración directa.

#### Representación gráfica de los datos

Se han implementado dos tipos de visualizaciones complementarias:

**Pirámide poblacional:** representa la distribución de pacientes por intervalos de años de ingreso, diferenciando por sexo.

**Diagrama de sectores:** muestra el porcentaje de hombres y mujeres asociados a cada diagnóstico, permitiendo una comparación visual rápida de las proporciones por categoría clínica.

### Accesibilidad y usabilidad

Siguiendo buenas prácticas de diseño inclusivo, la aplicación incorpora varias **funcionalidades de accesibilidad:**

**Modo oscuro**, para mejorar la visualización en distintos entornos de iluminación.

**Ajuste del tamaño de la tipografía**, adaptable a las necesidades del usuario.

**Selección de idioma**, con versiones en **español e inglés**.

Estas características garantizan una experiencia de uso más flexible, accesible y centrada en el usuario.

### Estructura adicional de la aplicación

Además de las herramientas principales, la web incluye una **página “About Us”**, en la que se presentan los integrantes del equipo de desarrollo, sus roles y la contribución de cada uno dentro del proyecto.

Despliegue



La aplicación ha sido **desplegada en la plataforma Vercel**, vinculada directamente con el **repositorio de GitHub** del proyecto.

Este flujo de trabajo permite mantener una **integración continua (CI/CD)**, de modo que cualquier modificación en el código fuente se refleja automáticamente en la versión publicada.

Gracias a este despliegue, la aplicación se encuentra **operativa y accesible en línea**, lo que facilita su evaluación y uso por parte de terceros.

### Hito 3