

Proyecto Final Bootcamp Ingeniería de Datos

La clínica Star Platinum es una institución en el ámbito de la odontología, contando con cerca de 30 años de experiencia, destacándose por la excelencia de sus especialistas. A lo largo de los años, la clínica ha mantenido su compromiso con ofrecer servicios de alta calidad a precios accesibles, en un esfuerzo por permanecer fieles a la filosofía de "si funciona, no lo cambies". Sin embargo, esta resistencia al cambio ha comenzado a impactar negativamente en la clínica, especialmente en su sistema de programación de citas, que ha permanecido inalterado desde su apertura.

En la actualidad, la Clínica Star Platinum enfrenta desafíos significativos debido a la evolución del mercado y las expectativas de los clientes. Entre estos desafíos se incluyen:

- **Presión Económica y Competencia:** Con el paso de los años, la inflación ha aumentado el costo de los tratamientos y consultas en general. Mientras que la clínica ha hecho esfuerzos por mantener bajos sus precios, esto ha llevado a una disminución en la calidad del servicio. Paralelamente, otras clínicas dentales han surgido, ofreciendo sistemas más modernos de gestión de citas y mejores servicios o tratamientos mas avanzados, atrayendo así a una parte de la clientela de Clínica Star Platinum.
- **Sistema de Citas Obsoleto:** El método tradicional de programación de citas con el que cuenta la Clínica Star Platinum —únicamente llamada telefónica o presencial o mensajes de whatsapp donde la recepcionista anota todo manualmente en una libreta y se va llevando todo el registro y/o seguimiento manualmente— se ha vuelto ineficiente y obsoleto. Comparado con las soluciones digitales implementadas por los competidores que incluyen programación de citas en línea, pago en línea, recordatorios automáticos y una gestión/seguimiento más eficiente del flujo de clientes por lo que el sistema actual de la clínica resulta anticuado y menos atractivo para los clientes mas jovenes.

Frente a estos desafíos, la Clínica Star Platinum reconoce la necesidad urgente de modernizar su sistema de gestión de citas junto a sus instalaciones para mejorar la eficiencia, atraer y retener a los clientes, y mantener su competitividad en el mercado.

El proyecto propuesto, un Sistema de reservas online, busca revitalizar la operación de la clínica mediante la implementación de una solución digital que permita a los clientes programar, modificar y cancelar citas fácilmente a través de una interfaz web. Esta solución no solo apunta a mejorar la experiencia del cliente sino también a optimizar la gestión de recursos y el flujo de pacientes, asegurando que la clínica pueda manejar un mayor volumen de citas de manera efectiva, manteniendo al mismo tiempo la calidad del servicio por la que ha sido conocida durante décadas.

Objetivo del Proyecto

El objetivo principal de este proyecto es desarrollar e implementar un sistema de reservas online para la Clínica Star Platinum, que modernice y mejore la gestión de citas. Este sistema deberá ser intuitivo, fácil de usar tanto para clientes como para el personal de la clínica, y capaz de integrarse con las operaciones existentes de la clínica sin alterar la calidad del servicio. Además, buscará proporcionar funcionalidades como recordatorios automáticos de citas, una visión clara de la disponibilidad de especialistas y tratamientos, y una mejor gestión del tiempo y recursos de la clínica.

Justificación del Proyecto

La necesidad de innovar el sistema de reservas de la Clínica Star Platinum se hace imperativa ante los crecientes desafíos que enfrenta el sistema tradicional, los cuales comprometen tanto la eficiencia operativa como la satisfacción del cliente. A continuación, se detallan las principales deficiencias que justifican la implementación de un nuevo sistema de reservas online:

- **Fallas en el Manejo de Citas:** El sistema actual, basado en llamadas telefónicas y anotaciones manuales, ha demostrado ser propenso a errores humanos significativos. Incidentes como citas empalmadas y la omisión de datos cruciales de pacientes son frecuentes, afectando la calidad del servicio y la confianza del cliente.
- **Dependencia del Personal Clave:** La eficacia del sistema actual depende en gran medida del personal administrativo. Por ejemplo, los especialistas tenían quejas sobre las omisiones o errores de empleados clave, como ocurrió cuando "Chachita" olvidó dejar preparadas las hojas de citas, estos errores pueden paralizar todo el proceso de programación, resultando en retrasos y confusión.
- **Legibilidad de la Información:** La información escrita manualmente en libretas a menudo resulta ilegible para pacientes y dentistas, lo que puede llevar a malentendidos sobre los tiempos y detalles de las citas, comprometiendo la eficiencia de las operaciones de la clínica.

Este proyecto no solo aborda estas deficiencias de manera efectiva, sino que también facilita y simplifica las tareas de todos los involucrados. Mediante la automatización de tareas como la programación y cancelación de citas, la delegación de procesos de pago y confirmaciones a un sistema digital se espera una mejora sustancial en la calidad del servicio. Adicionalmente, se propone habilitar una red WiFi gratis dentro del consultorio, permitiendo que los pacientes puedan actualizar su información y gestionar sus citas en tiempo real mientras esperan ser atendidos. Esta innovación no solo aumenta la comodidad para el usuario, sino que también optimiza el flujo de información y la comunicación dentro de la clínica.

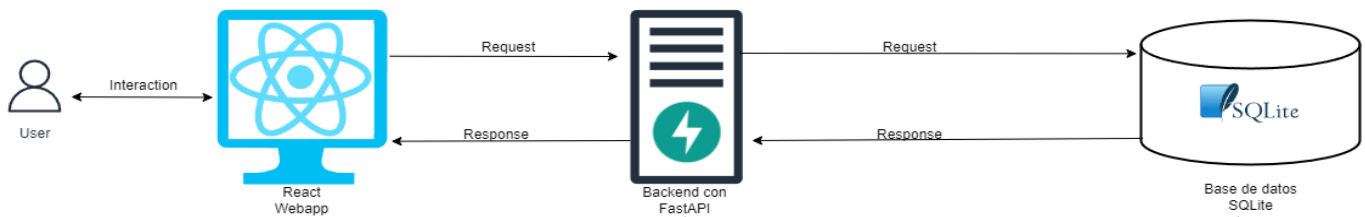
Es importante resaltar que este proyecto no tiene la intención de sustituir empleados, sino de optimizar el trabajo al simplificar las tareas repetitivas y reducir al mínimo posible los errores humanos que el personal enfrenta, lo que les permite dedicarse a otras actividades que demanden su atención.

Diseño de la Solución

Arquitectura del Sistema: El sistema estará basado en una “web app” (software que se ejecuta en el navegador web), accesible tanto para clientes como para el personal administrativo y médico de la clínica.

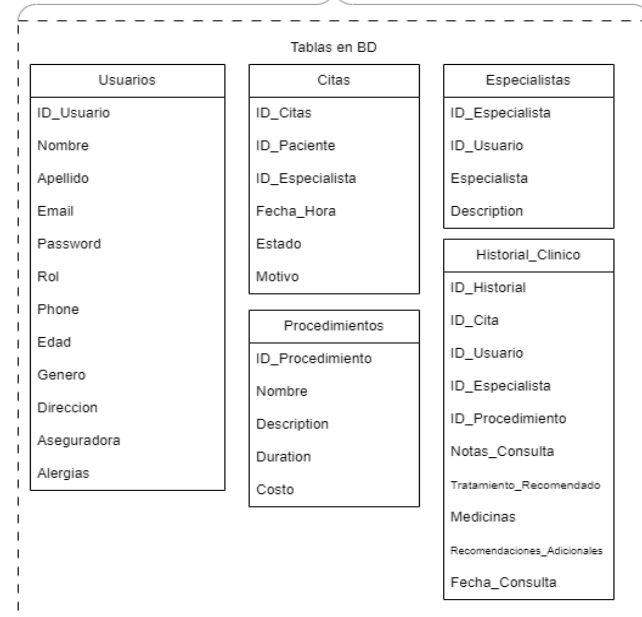
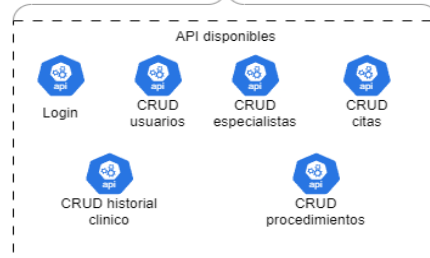
La arquitectura incluirá:

- **Frontend:** Interfaz de usuario intuitiva para la reserva, modificación y cancelación de citas. El frontend tendrá funcionalidades distintas dependiendo si el usuario es “Especialista” o “Paciente”.
- **Backend:** Servidor para procesar solicitudes, gestionar la base de datos de citas, y enviar recordatorios automáticos.
- **Base de Datos:** SQLite para almacenar información de usuarios, especialistas, citas, procedimientos y el historial clínico del paciente.
- **Modelado de Datos:** Se diseñará un esquema de base de datos relacional que capture las entidades principales (clientes, especialistas, citas, procedimientos y el historial clínico) y sus relaciones. Se utilizarán diagramas ER para visualizar la estructura de la base de datos.
- **Flujos de Usuario:** Se definirán flujos detallados para diferentes usuarios (pacientes o especialistas), cubriendo el proceso de reserva de citas, gestión de horarios, y acceso a historiales de tratamiento.
- **Arquitectura:**



Conceptos Clave

API: Conjunto de reglas y herramientas que permiten que diferentes partes del software se comuniquen y trabajen juntas de manera efectiva, simplificando el desarrollo de aplicaciones al proporcionar una forma estandarizada de acceso y manipulación de recursos.

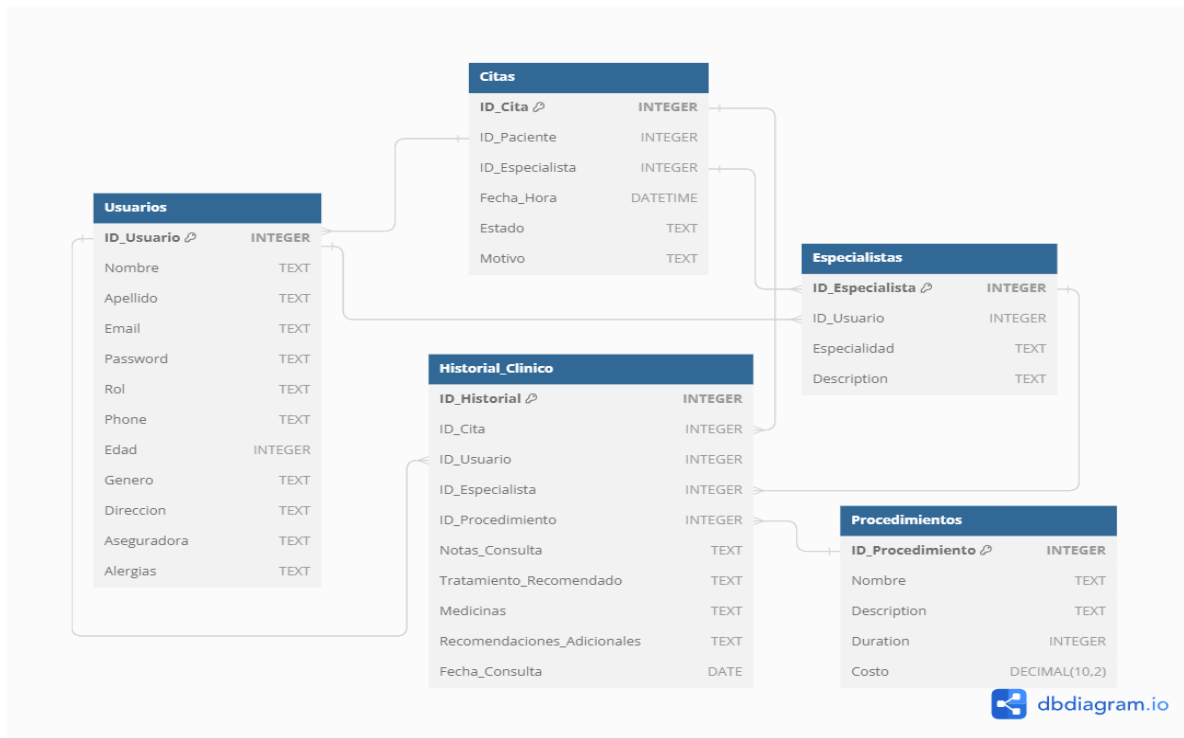


CRUD: Acrónimo que se utiliza para describir las cuatro operaciones básicas que se pueden realizar en la mayoría de los sistemas de gestión de bases de datos. Estas operaciones son:

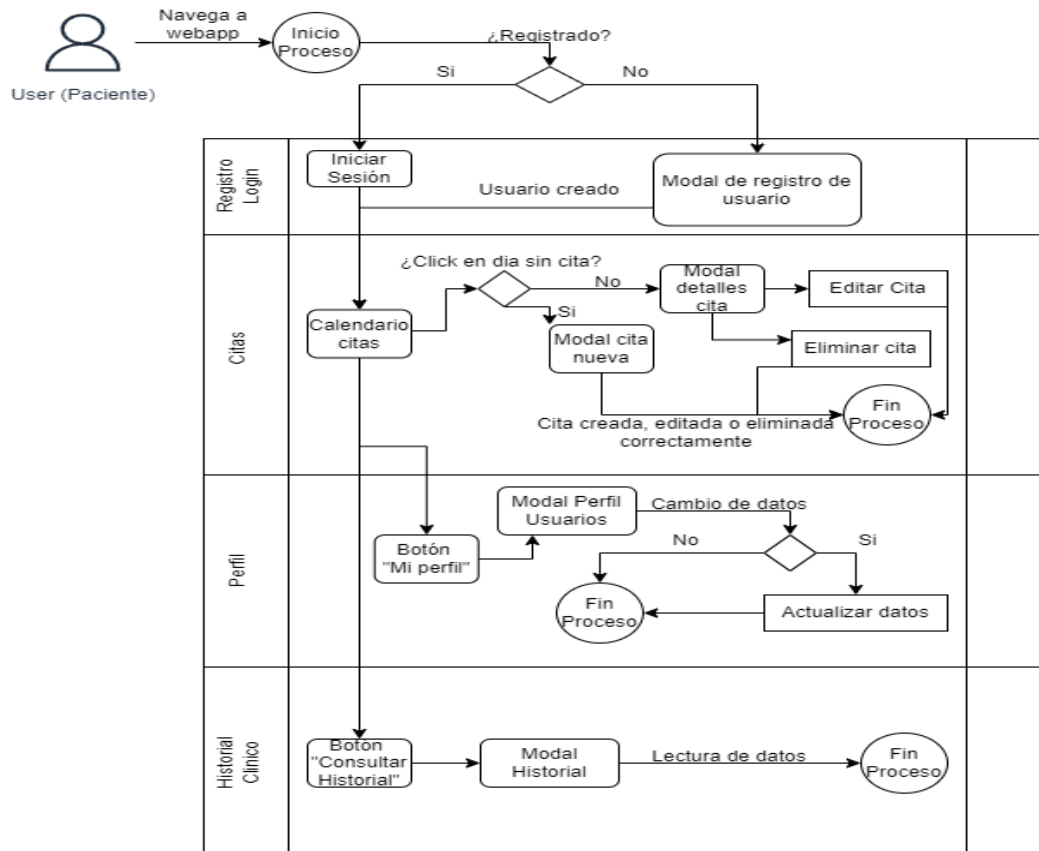
1. **Crear (Create):** Esta operación implica la creación de un nuevo registro o entrada en la base de datos.
2. **Leer (Read):** Esta operación implica la lectura o recuperación de datos de la base de datos. Es utilizada para obtener información existente.
3. **Actualizar (Update):** Esta operación implica la modificación de un registro existente en la base de datos.
4. **Eliminar (Delete):** Esta operación implica la eliminación de un registro existente en la base de datos.

Estas cuatro operaciones básicas son fundamentales en el desarrollo de aplicaciones que interactúan con bases de datos u otros sistemas de almacenamiento de datos, ya que proporcionan una forma estándar y efectiva de crear, leer, actualizar y eliminar información.

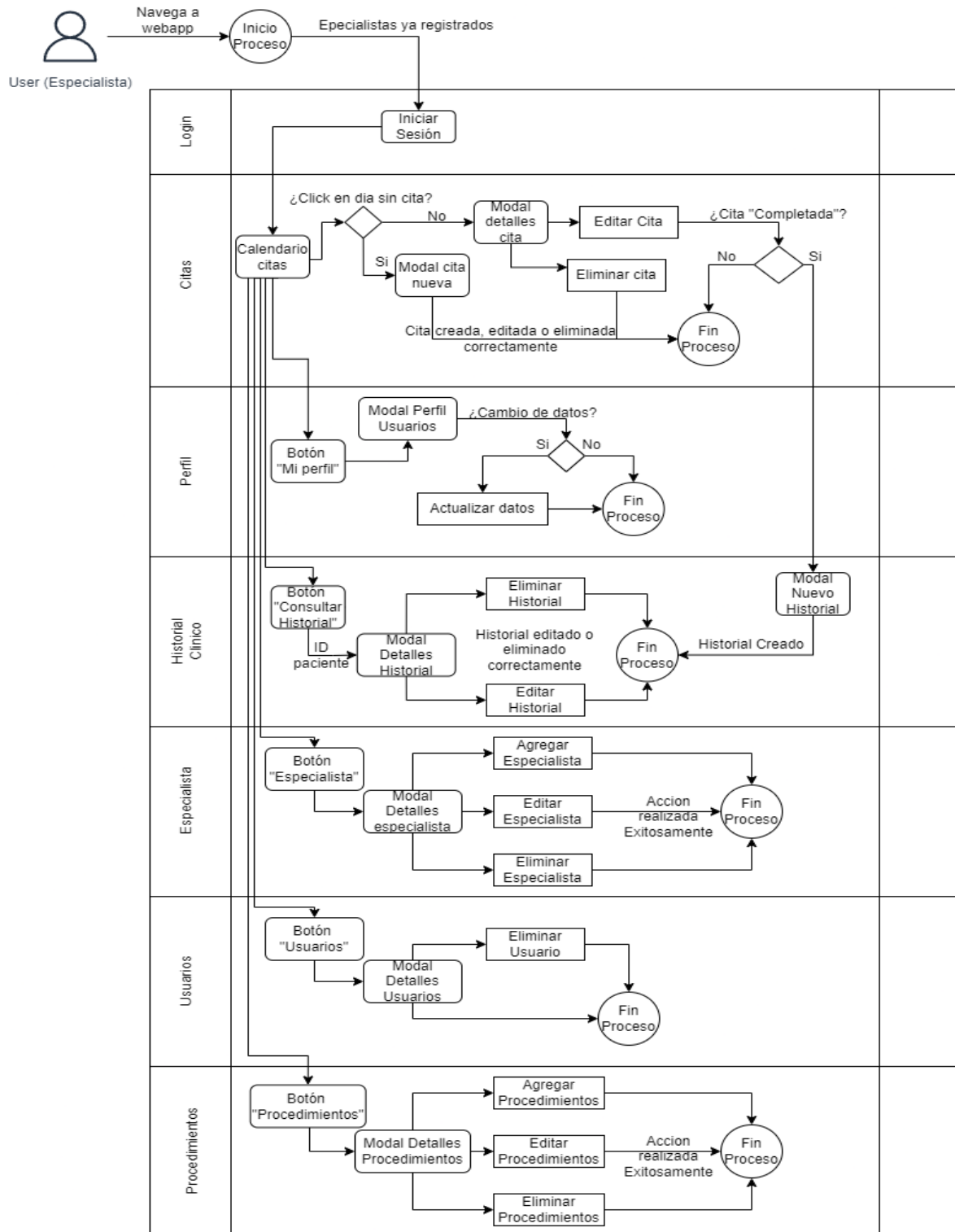
- ERD



- Flujo de usuario: Paciente



- Flujo de usuario: Especialista/Administrador



Desarrollo Técnico

Tecnologías Utilizadas: Este proyecto se construirá utilizando un stack moderno y eficiente para desarrollo web, se utilizarán los frameworks **React** para frontend y **FastAPI** para backend complementado por **SQLite** para la gestión de la base de datos.

Esta combinación de tecnologías ofrece un desarrollo ágil y una integración fluida, optimizando el proceso de creación del sistema de reservas online.

FastAPI: Utilizado para desarrollar la API backend del sistema de reservas. FastAPI facilita la creación de API endpoints de alta performance que son asíncronos y basados en tipado estático, lo que mejora la velocidad de desarrollo y la seguridad del código. FastAPI integrará operaciones de base de datos para cada una de las características que el sistema requiera.

React: Empleado para construir la interfaz de usuario del sistema de reservas. React facilita la creación de aplicaciones web de una sola página donde el contenido se carga dinámicamente sin necesidad de recargar la página completa. La interfaz permitirá a los usuarios pacientes realizar reservas, ver disponibilidades, y gestionar citas existentes de manera eficiente y en tiempo real. Si son especialistas tendrán funcionalidades extras sobre el sistema como la creación de historiales clínicos, modificaciones a los procedimientos entre otras acciones.

SQLite: La base de datos relacional elegida para almacenar datos del sistema debido a su simplicidad, eficiencia y fácil integración con FastAPI y el entorno de desarrollo Python. La estructura de la base de datos está diseñada para soportar eficientemente las operaciones clave del sistema. La integración de SQLite con FastAPI se realizará a través de ORMs (Object-Relational Mapping) como SQLAlchemy, facilitando operaciones de datos asíncronas y seguras. Se pensó mucho en la estructura de tablas y sus relaciones para evitar cuellos de botella, a su vez también se revisaron todos los queries para evitar consultas pesadas que puedan alentar la base de datos.

GitHub Actions como Orquestador (CI/CD):

Integración Continua (CI): Se configuraron workflows de GitHub Actions para automatizar el despliegue a github pages de todo el frontend y la creación de la imagen contenedor de Docker donde se encuentran los endpoints, se ejecutará cada vez que se realice un commit en las carpetas del frontend o backend en el repositorio, se tiene configurado "Eslint" para identificar y corregir errores de sintaxis, estilo y notificar problemas potenciales en el código asegurando que el código nuevo no introduzca errores y mantenga la calidad del proyecto.

Despliegue Continuo (CD): El workflow que se encarga de la compilación y despliegue del frontend lo hace en github pages, el cual ya se encuentra funcionando exitosamente y únicamente se ejecutará cuando se haga un commit en código del frontend y que ese código supere la evaluación del Eslint, garantizando que los cambios se vean reflejados en la página web en cuestión de minutos y sin errores.

El workflow que contruye la imagen del contenedor, únicamente la guarda y actualiza dentro del repositorio, en ese estado se puede tomar la imagen y desplegarla en algún sitio en la nube o desde algún servidor. O mejor aún, agregar esas instrucciones directamente en el workflow para que en automático se despliegue.

Componente de Ingeniería de Datos

Para este módulo, se decidió adelantar el desarrollo de la base de datos del módulo “ventas” de la clínica, creando una nueva tabla llamada “ventas”. En esta tabla se almacenan detalles de las transacciones realizadas después de cada cita completada, junto con otros detalles importantes como el método de pago, los medicamentos recetados y el estado de pago.

id_venta	ID_Usuario	id_cita	tratamiento	costo_total	fecha	ID_Especialista	metodo_pago	estado_pago	medicamentos_recetados
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	132	1001	Limpieza Dental	\$50.00	2024-05-01 09:00:00	10	Tarjeta de Crédito	Pagado	Ibuprofeno 400mg, Paracetamol 500mg
2	133	1002	Extracción de Muelas	\$150.00	2024-05-02 10:30:00	11	Efectivo	Meses sin intereses	Amoxicilina 500mg
3	134	1003	Blanqueamiento Dental	\$200.00	2024-05-03 11:00:00	12	Tarjeta de Débito	Pagado	Diclofenaco 100mg
4	135	1004	Carillas Dentales	\$300.00	2024-05-04 12:00:00	13	Transferencia Bancaria	Pagado	Paracetamol 500mg
5	136	1005	Implantes Dentales	\$1000.00	2024-05-05 13:30:00	14	Efectivo	Pagado	Ibuprofeno 400mg, Amoxicilina 500mg
6	137	1006	Ortodoncia	\$500.00	2024-05-06 14:00:00	15	Tarjeta de Crédito	Meses sin intereses	N/A
7	138	1007	Limpieza Dental	\$50.00	2024-05-07 15:30:00	16	Efectivo	Pagado	Ibuprofeno 400mg
8	139	1008	Extracción de Muelas	\$150.00	2024-05-08 16:00:00	17	Tarjeta de Débito	Meses sin intereses	Amoxicilina 500mg

La tabla "ventas" se diseñó intencionalmente para aceptar datos sin procesar (incluyendo caracteres especiales, signos de puntuación, etc.), con el fin de aplicar un proceso de transformación de datos.

Para este propósito, se eligió usar DBT (Data Build Tool), una herramienta de código abierto que facilita la transformación y modelado de datos de manera controlada, reutilizable y confiable. La versión de DBT utilizada fue "dbt Core", ejecutada localmente desde la consola junto con el conector "dbt-sqlite" para conectarse a la base de datos SQLite.

Se desarrolló el siguiente query que se ejecuta desde DBT para transformar los datos de la tabla "ventas". En la configuración de DBT se indica que durante la primera ejecución del script se generará una tabla nueva llamada "ventas_transformadas" con todos los datos actuales de la tabla "ventas". Posteriormente, en ejecuciones subsecuentes, DBT únicamente actualizará la tabla "ventas_transformadas" con los nuevos registros.

```

select
  id_venta,
  ID_Usuario,
  id_cita,
  lower(trim(treatment)) as tratamiento, -- Convertir tratamiento a minúsculas
  cast(replace(cost_total, '$', '' ) as decimal(10,2)) as costo_total, -- Limpiar y convertir costo_total a decimal
  date(fecha) as fecha, -- Normalizar fecha a solo la parte de la fecha
  strftime('%Y', fecha) as año, -- Crear columna derivada para el año
  strftime('%m', fecha) as mes, -- Crear columna derivada para el mes
  ID_Especialista,
  case
    when lower(trim(metodo_pago)) = 'tarjeta de crédito' then 1
    when lower(trim(metodo_pago)) = 'efectivo' then 2
    when lower(trim(metodo_pago)) = 'tarjeta de débito' then 3
    when lower(trim(metodo_pago)) = 'transferencia bancaria' then 4
    else null
  end as metodo_pago_codificado, -- Codificar metodo_pago
  lower(trim(metodo_pago)) as metodo_pago, -- Limpiar y estandarizar metodo_pago original
  case
    when lower(trim(estado_pago)) = 'meses sin intereses' then 2
    when lower(trim(estado_pago)) = 'pagado' then 3
    when lower(trim(estado_pago)) = 'pendiente' then 1
    else null
  end as estado_pago_codificado, -- Codificar estado_pago
  lower(trim(estado_pago)) as estado_pago, -- Limpiar y estandarizar estado_pago original
  replace(medicamentos_recetados, '\n', ' ') as medicamentos_recetados, -- Limpiar caracteres especiales en medicamentos_recetados
  length(replace(medicamentos_recetados, ' ', '')) - length(replace(replace(medicamentos_recetados, ' ', ''), ',', '')) + 1 as cantidad_medicamentos
from
  ventas

```

Este es el log de DBT indicando que se han transformado correctamente los datos y que la tabla “ventas_transformadas” ha sido creada.

```

02:20:18 Running with dbt=1.5.11
02:20:18 Registered adapter: sqlite=1.5.0
02:20:18 [WARNING]: Configuration paths exist in your dbt_project.yml file which do not apply to any resources.
There are 1 unused configuration paths:
- models.DataDental.ventas_transformadas
02:20:18 Found 1 model, 0 tests, 0 snapshots, 0 analyses, 311 macros, 0 operations, 0 seed files, 0 sources, 0 exposures, 0 metrics, 0 groups
02:20:18
02:20:18 Concurrency: 1 threads (target='dev')
02:20:18
02:20:18 1 of 1 START sql table model main.ventas_transformadas_table ..... [RUN]
02:20:18 1 of 1 OK created sql table model main.ventas_transformadas_table ..... [OK in 0.04s]
02:20:18
02:20:18 Finished running 1 table model in 0 hours 0 minutes and 0.08 seconds (0.08s).
02:20:18
02:20:18 Completed successfully
02:20:18
02:20:18 Done. PASS=1 WARN=0 ERROR=0 SKIP=0 TOTAL=1

```

Con la tabla "ventas_transformadas" es posible realizar diversos tipos de análisis, como:

- Total de ventas por año y mes
- Distribución de métodos de pago
- Costo promedio de tratamiento por método de pago
- Tratamientos más comunes
- Estado de pago de las ventas
- Cantidad de medicamentos recetados por tratamiento
- Frecuencia de medicamentos recetados

Se planea agregar un botón visible solo para los administradores llamado "Ver reporte de ventas" que al hacer clic, se desplegará un dashboard interactivo donde el administrador podrá conocer gráficamente los detalles de ventas.

El objetivo del dashboard es proporcionar una visión integral y accesible de las métricas clave, facilitando la toma de decisiones informadas sobre la gestión y optimización de las operaciones de la clínica.

El diseño se basará en otros dashboards que proporcionan datos médicos, asegurando así que la presentación de la información sea clara, relevante y fácil de interpretar. Para crear el dashboard, se planea usar el software Tableau y la tabla "ventas_transformadas".

Diseño propuesto:

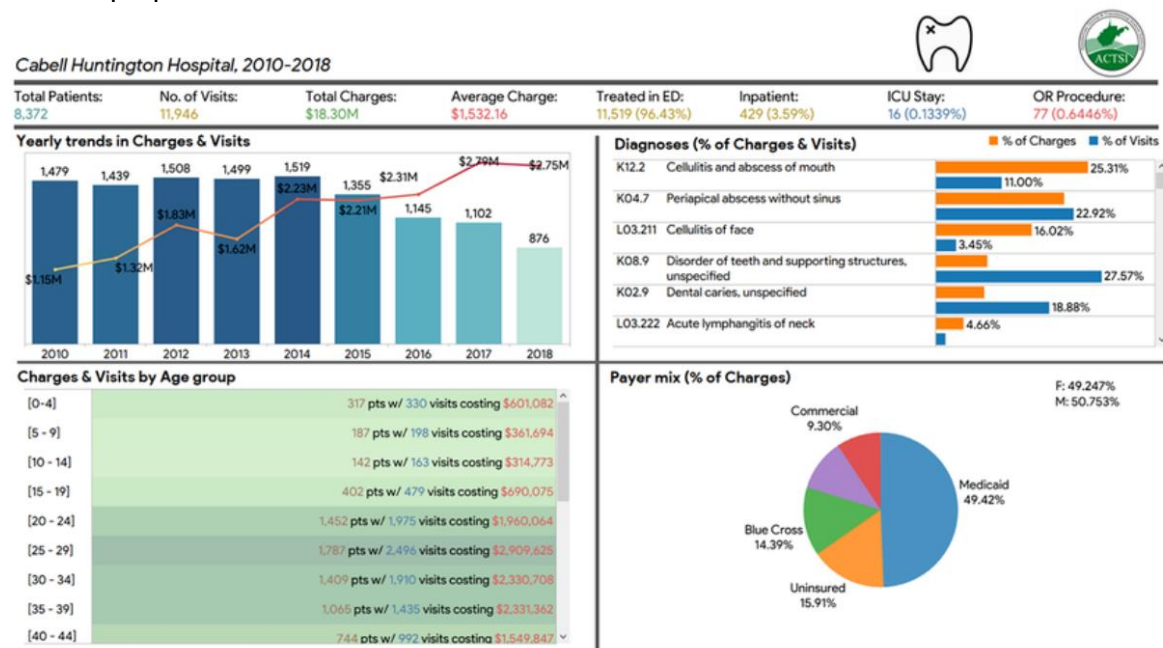


Figure 1 Dashboard del Cabell Huntington Hospital

Código y Funcionalidades Principales:

- Sistema de autenticación y autorización para diferentes roles de usuario (pacientes, especialistas/administrador).
- Endpoints de FastAPI para realizar cualquier acción CRUD hacia las tablas de usuarios, especialistas, citas, procedimientos, historial.
- Interfaz de React para la interacción del usuario con el sistema, proporcionando una experiencia de usuario fluida y accesible.
- Automatización de recordatorios y notificaciones utilizando la infraestructura de FastAPI para comunicarse con los usuarios.
- Pago en línea, para que el usuario solo se presente a su cita.
- Sección para reportar algún error dentro del sistema y/o dar feedback correspondiente.

Pruebas

Para asegurar la funcionalidad, seguridad, y usabilidad del sistema, se implementará un plan de pruebas intensivo que incluirá:

Pruebas Unitarias Backend: Verificarán la lógica individual de los componentes de FastAPI, utilizando frameworks de prueba como Pytest. Estas pruebas ejecutarán todos los endpoints como si un usuario estuviera siguiendo algún flujo en específico, por ejemplo, crear un usuario, login con ese usuario, agendar una cita y luego modificarla, etc.

Pruebas Integración Frontend: Aseguran que todos los componentes de React interactúen correctamente, se usaran herramientas como Jest que es muy popular para realizar pruebas en javascript y ya viene preconfigurado en React.

Resultados de las Pruebas: Se documentarán los resultados de las pruebas, incluyendo errores detectados, áreas de mejora identificadas, y cómo se resolvieron o planean resolver estos problemas. Este registro ayudará a informar las decisiones de desarrollo futuro y asegurar un ciclo de mejora continua.

Conclusiones y Trabajo Futuro

A lo largo del desarrollo del proyecto, se han cumplido de manera satisfactoria los objetivos establecidos inicialmente, demostrando la viabilidad y la eficacia de una solución digital en un entorno clínico tradicionalmente operado de manera manual.

El uso de FastAPI y SQLite ha permitido desarrollar una base sólida que soporta eficientemente el alto volumen de operaciones diarias, mientras que el frontend desarrollado en React ha facilitado una interfaz amigable y eficaz para los usuarios. La metodología de CI/CD adoptada ha sido crucial para el mantenimiento de la calidad del código y la agilidad en el despliegue de nuevas funcionalidades, lo cual ha resultado en un desarrollo continuo sin interrupciones significativas.

Por lo que podemos concluir que el proyecto de implementación del sistema de reservas online para la Clínica Star Platinum ha sido un emprendimiento significativo que ha logrado transformar fundamentalmente la manera en que la clínica gestiona sus citas y se comunica con sus pacientes.

Sugerencias para Mejoras Futuras: Se analizará con mucho detalle el feedback proporcionado por los usuarios y los resultados de las pruebas para poder incluir sus ideas en futuras actualizaciones, algunas sugerencias que ya se tienen pensadas agregar incluyen:

- **Integración con Sistemas de Pago:** Para facilitar el proceso de pagos por servicios en línea, mejorando así la experiencia del usuario.
- **Implementación de Inteligencia Artificial:** Para optimizar la programación de citas, su horario, la asignación de procedimientos u ofrecer posibles tratamientos analizando los historiales de los usuarios o incluso ofrecer precios dinámicos basándose en patrones de demanda.
- **Mejoras en la Seguridad:** Continuar evaluando y reforzando las medidas de seguridad del sistema, especialmente en lo que respecta a la protección de datos personales y transacciones financieras.

Todo lo anterior junto a un **sistema de actualización semanal** donde se darán solución a errores detectados o bugs encontrados.