UNIVERSIDAD NACIONAL EXPERIMENTAL
PARA LAS TELECOMUNICACIONES E INFORMÁTICA (UNETI)
VICERRECTORADO ACADÉMICO
PROGRAMA NACIONAL DE FORMACIÓN EN INFORMÁTICA
U.C:Base de datos I (Informática) Sección 6B y 7B 24/2

Caso de Estudio 41: Sistema de Gestión de Pacientes para una Clínica Dental "Sonrisa Perfecta”

TUTORA:
Yuly Delgado

AUTOR:
Frederick Durán
CI. 30346056

Caracas, Marzo de 2025

Instrucciones Para la Evaluación 3 según la plataforma UNETI:

Diseño Lógico y Físico de la Base de Datos

Fase 1: Diseño de Tablas y Relaciones (Sección Didáctica 1)

Fase 2: Inserción, Consulta y Eliminación de Datos

Fase 3: Diseño Lógico y Físico de la Base de Datos

Tarea 3.1: Diseño lógico:

Refinar el diseño lógico de la base de datos basado en los resultados de las secciones anteriores.

Considerar la escalabilidad y el rendimiento a largo plazo.

Tarea 3.2: Diseño físico:

Optimizar el diseño físico de la base de datos, incluyendo la elección de tipos de almacenamiento, índices y particionamiento de tablas.

Configurar los parámetros de PostgreSQL para mejorar el rendimiento.

Tarea 3.3: Tuneado de consultas:

Analizar los planes de ejecución de las consultas y realizar ajustes para mejorar su rendimiento.

Tarea 3.4: Backup y recuperación:

Implementar una estrategia de backup y recuperación para proteger los datos.

Indicadores de Evaluación:

Comprensión del diseño de bases de datos: Demuestra un profundo entendimiento de los conceptos de diseño lógico y físico, normalización, índices, particionamiento y optimización de consultas.

Análisis crítico: Evalúa el diseño inicial y propone mejoras basadas en los resultados de las secciones anteriores y en los requisitos del sistema.

Toma de decisiones: Justifica las decisiones de diseño tomadas, considerando factores como el rendimiento, la escalabilidad y la mantenibilidad.

Optimización: Aplica técnicas de optimización para mejorar el rendimiento de la base de datos, tanto a nivel de diseño físico como a nivel de consultas.

Seguridad: Implementa medidas de seguridad adecuadas para proteger los datos de la base de datos.

Planes de ejecución: Análisis de los planes de ejecución de consultas para identificar cuellos de botella y proponer mejoras.

Estrategia de backup y recuperación: Un plan detallado para realizar copias de seguridad de la base de datos y restaurarla en caso de desastre.

Para ésta sección didáctica cada participante trabajará de forma individual

Desarrollará su caso de estudio, aplicando lo aprendido en las secciones didácticas 1 y 2 más lo correspondiente a la fase 3. Es decir el documento de entrega debe incluir las actividades descritas en dichas secciones aplicadas al nuevo caso de estudio.

En este informe diseñó e implementó un sistema de gestión de pacientes para la clínica dental "Sonrisa Perfecta". El sistema tiene como objetivo mejorar la organización y administración de citas, historiales clínicos, pagos, tratamientos, inventario y comunicación con los pacientes.

Se detallan el diseño lógico y físico de la base de datos, las estrategias de optimización, seguridad y mecanismos de respaldo y recuperación de datos.

Fase 1: Diseño de Tablas y Relaciones

Entidades y atributos principales usados en las tablas relacionales:

Pacientes (ID, Nombre, Apellidos, DNI, Teléfono, Email, Dirección, Fecha_Nacimiento, Género, Alergias, Antecedentes_Médicos, Activo, Fecha_Creación, Fecha_Actualización, Fecha_Eliminación)

Citas (ID, Paciente_ID, Dentista_ID, Fecha_Hora, Tipo_Cita, Estado, Notas, Fecha_Creación, Fecha_Actualización, Fecha_Eliminación)

Historial Clínico (ID, Paciente_ID, Dentista_ID, Fecha_Consulta, Diagnóstico, Tratamiento, Observaciones, Radiografía_URL, Prescripción, Medicamentos, Estado, Fecha_Creación, Fecha_Actualización, Fecha_Eliminación)

Tratamientos (ID, Nombre, Descripción, Precio, Duración_Aproximada, Activo, Fecha_Creación, Fecha_Actualización, Fecha_Eliminación)

Pagos (ID, Paciente_ID, Tratamiento_ID, Monto, Método_Pago, Estado, Fecha_Pago, Número_Factura, Notas, Fecha_Creación, Fecha_Actualización, Fecha_Eliminación)

Inventario (ID, Nombre, Descripción, Cantidad, Cantidad_Mínima, Proveedor, Fecha_Caducidad, Precio_Unitario, Ubicación, Activo, Fecha_Creación, Fecha_Actualización, Fecha_Eliminación)

Usuarios (o users) (ID, Nombre, Email, Contraseña, Rol, Fecha_Creación, Fecha_Actualización)

Modelo lógico del Diagrama entidad relación:

Pacientes – Citas (1:N) → Un paciente puede tener muchas citas.

Citas – Pagos (1:1) → Una cita puede estar asociada a un solo pago.

Citas – Historial Clínico (1:N) → Una cita puede generar varios registros en el historial.

Historial Clínico – Radiografías (1:N) → Un historial clínico puede incluir varias radiografías.

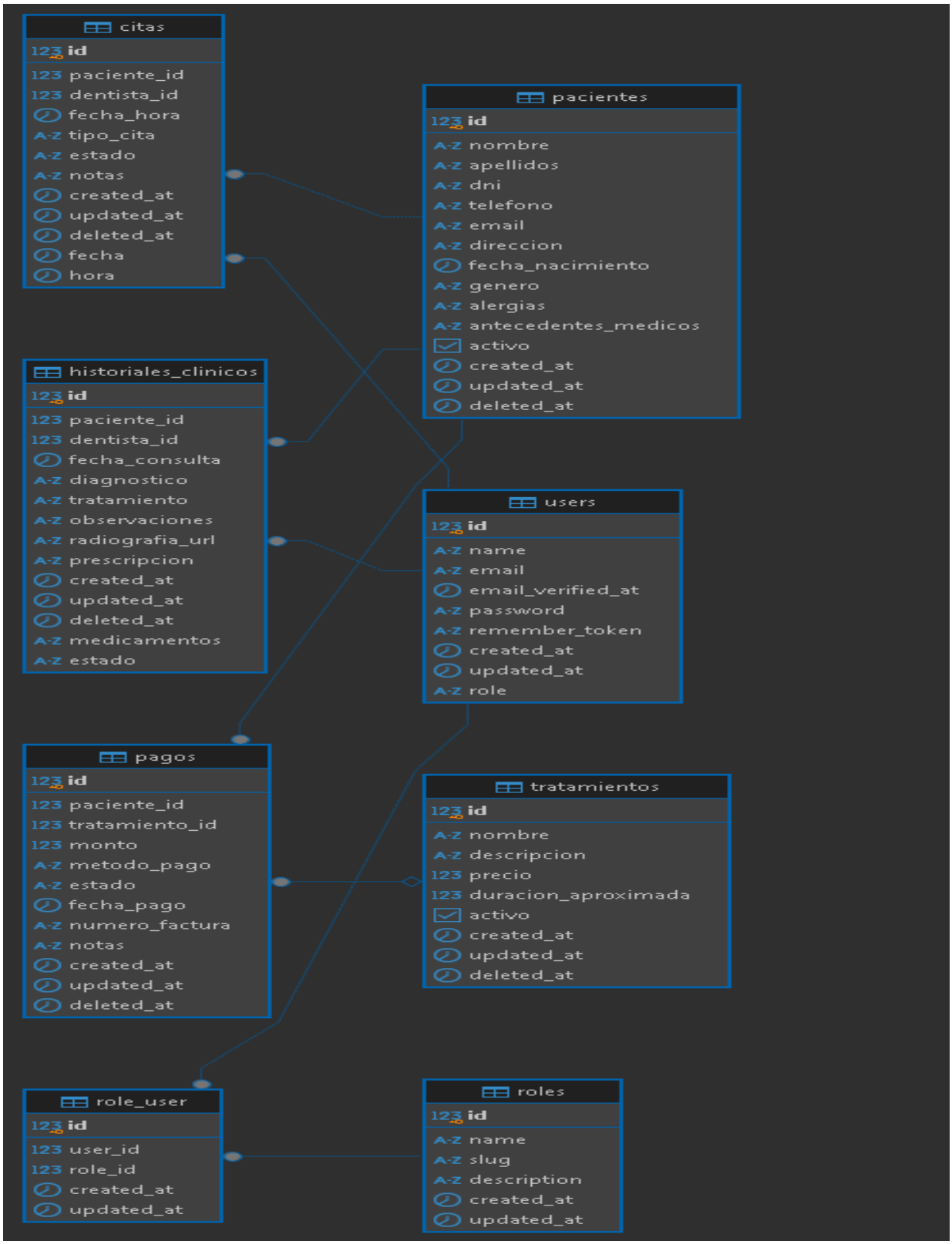
Usuarios – Citas (1:N) → Un dentista atiende muchas citas.

Usuarios – Especialidades (1:N) → Un dentista tiene una especialidad.

Usuarios – Horarios Dentistas (1:N) → Un dentista tiene un horario de atención.

Inventario – Tratamientos (1:N) → Un tratamiento puede requerir varios materiales del inventario.

Y a continuación un ejemplo del diagrama entidad relación que se pidió, extraído del Programa DBeaver:



Fase 2: Inserción, Consulta y Eliminación de Datos

Ejemplos de Consultas SQL en Pgadmin4:

Insertar un Paciente (ejemplo de inserción):

```sql
INSERT INTO Pacientes (Nombre, Apellidos, DNI, Telefono, Email)
VALUES ('Juan', 'Perez', '12345678A', '987654321', 'juan.perez@email.com');
```

Consultar las **Citas** Programadas:

```sql
SELECT * FROM Citas WHERE Estado = 'Programada';
```

Comando SQL para eliminar una cita en el Script **(tener cuidado):**

```sql
DELETE FROM Citas WHERE ID = 10;
```

Fase 3: Diseño Lógico y Físico de la Base de Datos

Diseño Lógico:

- Aplicación de la **Tercera Forma Normal (3NF)**.
- Definición de claves primarias y foráneas.
- Indexación para optimizar consultas frecuentes.

Diseño Físico:

- Base de datos implementada en **PostgreSQL**.
- Uso de **JSONB** para almacenar observaciones.
- Particionamiento de tablas por fecha.

Tuneado de Consultas:

```sql
EXPLAIN ANALYZE SELECT * FROM Citas WHERE Fecha >= '2025-01-01';

este comando sirve para consultar las citas por orden y fecha
```

Backup y Recuperación:

- Los backups incrementales diarios y completos semanales!
- Uso de pg_basebackup y PITR (Point-In-Time Recovery).

CASOS DE USO

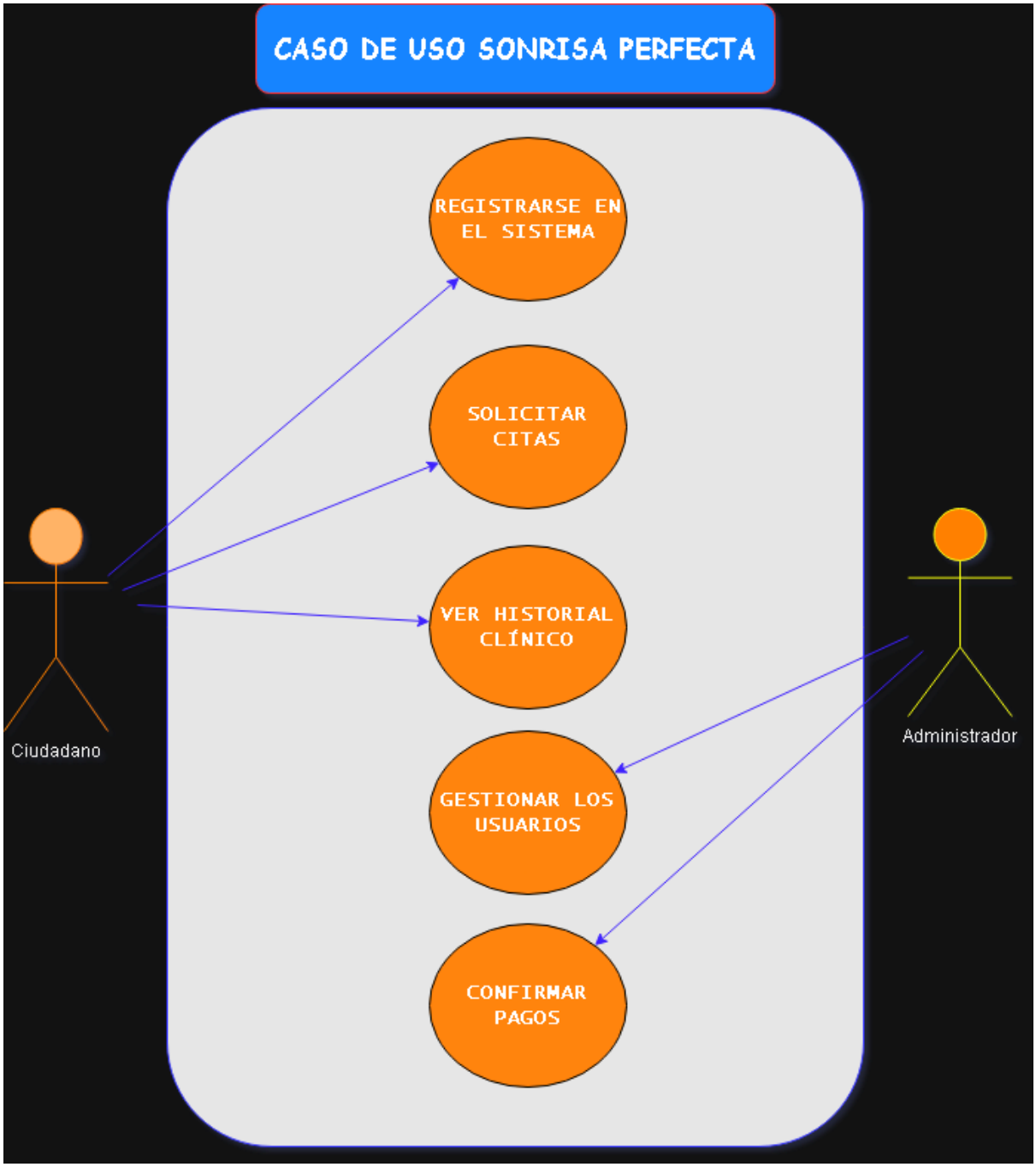
Registrar una Cita como ejemplo el usuario:

- **Actor:** Paciente.
- **Flujo:**
    1. Seleccionar paciente y dentista.
    2. Definir fecha y hora.
    3. Confirmar cita.

Administrar el Sistema:

- **Actor:** Administrador**.**
- **Flujo:**
    1. Seleccionar el paciente o registrarlo.
    2. Registrar monto y método de pago.
    3. Confirmar y generar recibo.

A continuación un diagrama de caso de uso elaborado en draw.io:



Seguridad y Privilegios

Se implementó roles en base a los usuarios y técnicos y administradores, como base de datos esto integré en la DB de PostgreSQL:

- Implementación de roles y permisos con el GRANT (super importante, al ser una sentencia de SQL que permite otorgar permisos a usuarios o roles en una base de datos)

```
1   GRANT SELECT, INSERT, UPDATE ON Pacientes TO Recepcionista;
```

- Encriptación de datos sensibles.
- Auditoría de accesos.

ANEXOS

Importante.

Links a Repositorios:

https://github.com/ragnarsson03/Sistema-de-Gesti-n-de-Pacientes-para-Cl-nica-Dental-Sonrisa-Perfecta---Caso-de-Estudio-41 (El Sistema de Sonrisa Perfecta)

https://github.com/ragnarsson03/Sonrisa-Perfecta-SD3 (Informe, Readme.md y Diagramas)

Scripts SQL para creación de la base de datos (no todos):

Capturas de pantalla del sistema en funcionamiento.

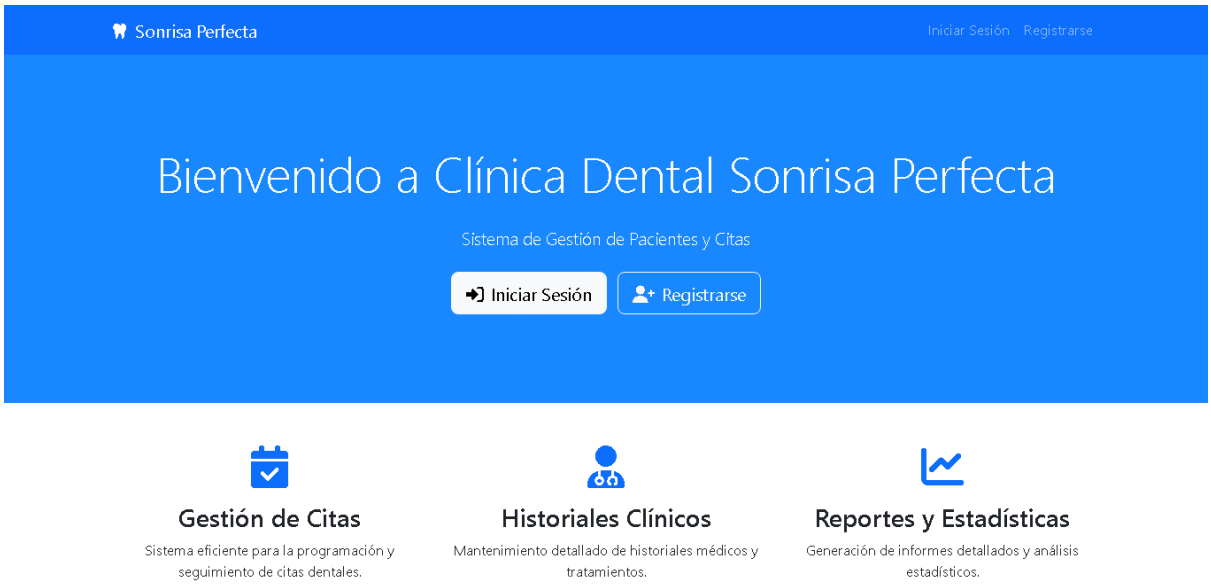
Repositorio GitHub con el código fuente y su documentación adicional

Demostración de la aplicación utilizada:

El sistema es posible correrlo en cualquier máquina local y es capaz de desarrollarse para producción, en este caso se utilizó el Framework de Laravel y conocimientos en Php para desarrollar el Login y la conexión a la base de datos y poder guardar los datos correctamente.También se utilizó HTML y CSS básico, no olvidar el script elaborado en PostgreSQL con las tablas ingresadas mediante la migración de las mismas en los archivos del proyecto.

Es posible registrarse como individuo para las citas.

Ejemplos:



Sonrisa Perfecta | Dashboard | Pacientes | Citas | Historiales | Pagos | Admin

¡Bienvenido/a, Admin!

jueves 6 de marzo de 2025

Nueva Cita | Nuevo Paciente | Nuevo Historial | Nuevo Pago

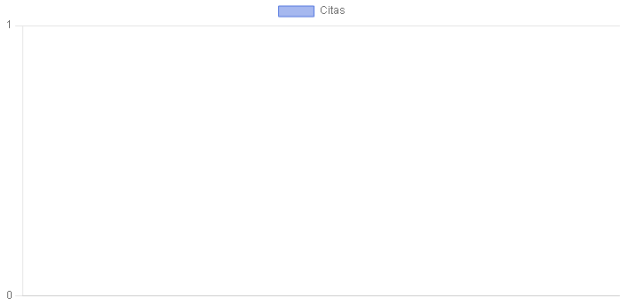
Citas Hoy
0

Pacientes
1

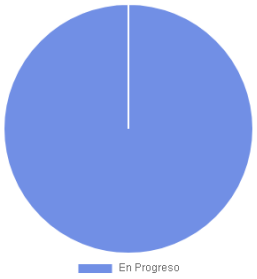
Tratamientos
0

Pagos Pendientes
$123,123.00

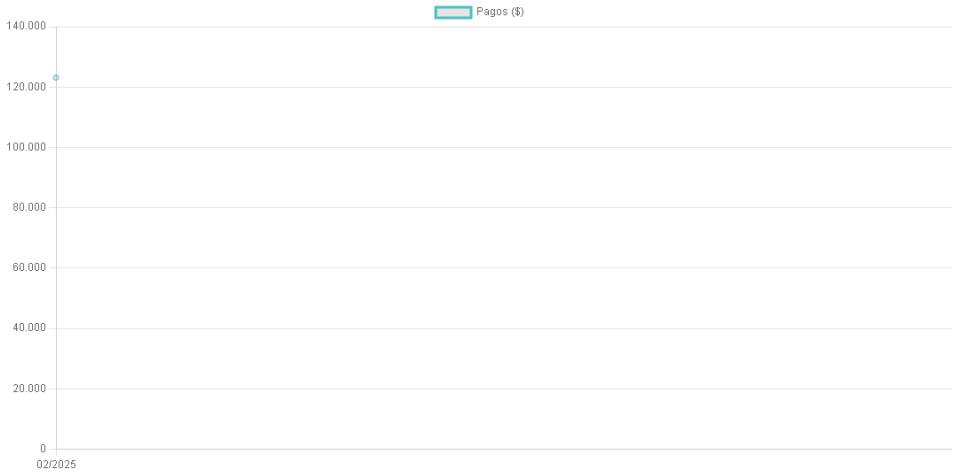
Citas por Día



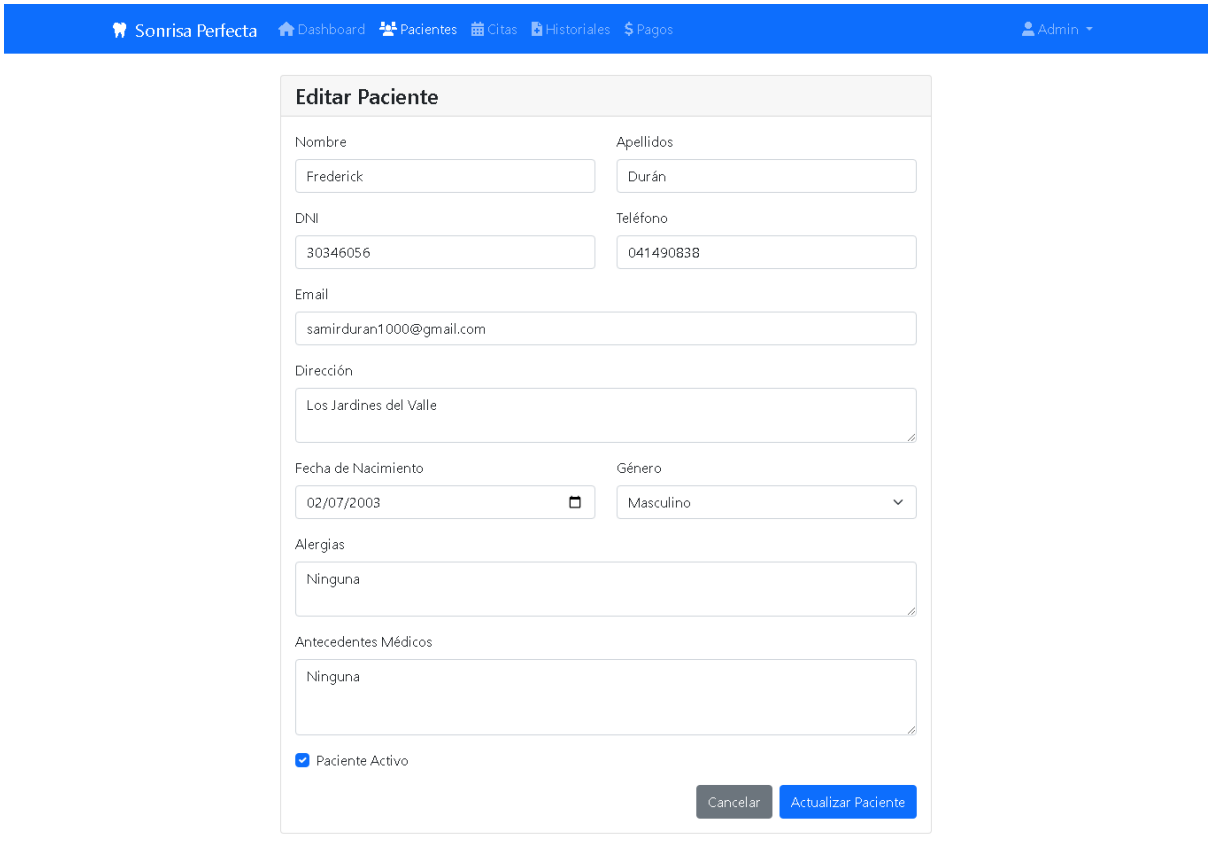
Tratamientos por Estado

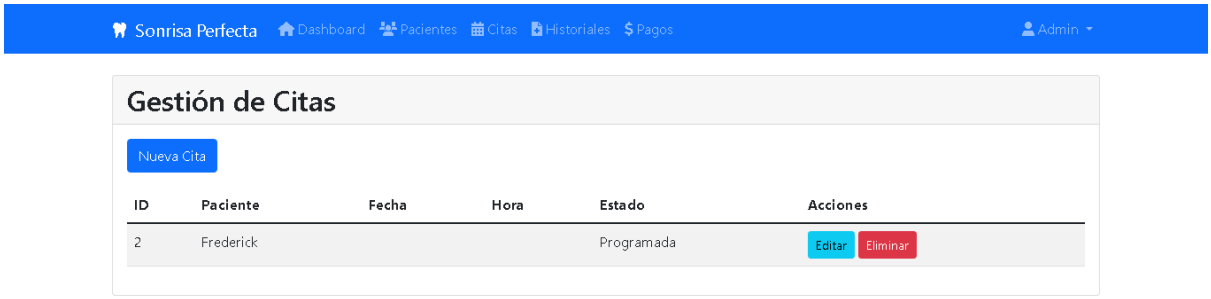


Pagos Mensuales

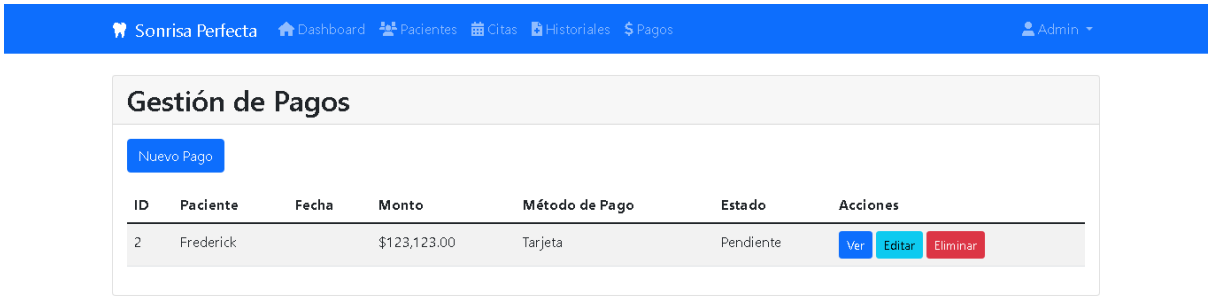



Sonrisa Perfecta
Dashboard
Pacientes
Citas
Historiales
Pagos
Admin
Pacientes
Nuevo Paciente
Buscar por nombre, DNI o email...
ID
Nombre
DNI
Teléfono
Email
Estado
Acciones
1
Frederick Durán
30346056
041490838
samirduran1000@gmail.com
Activo


Sonrisa Perfecta
Dashboard
Pacientes
Citas
Historiales
Pagos
Admin
Editar Paciente
Nombre
Frederick
Apellidos
Durán
DNI
30346056
Teléfono
041490838
Email
samirduran1000@gmail.com
Dirección
Los Jardines del Valle
Fecha de Nacimiento
02/07/2003
Género
Masculino
Alergias
Ninguna
Antecedentes Médicos
Ninguna
Paciente Activo
Cancelar
Actualizar Paciente



Importante: Se recomienda correr las migraciones y usar el comando de **Php artisan serve** para correr el servidor, nuevamente recalcando que para migrar la base de datos y sus debidas tablas desarrolladas debe utilizarse el comando **Php artisan migration**



Sonrisa Perfecta | Dashboard | Pacientes | Citas | Historiales | Pagos | paciente2

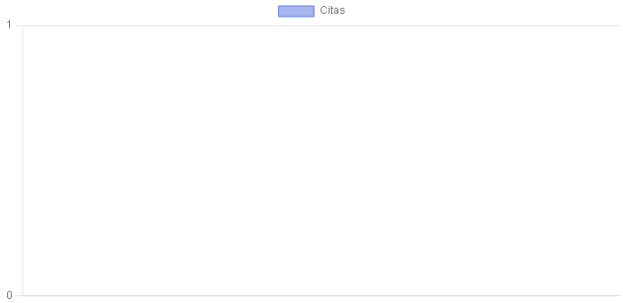
¡Bienvenido/a, paciente2!

jueves 6 de marzo de 2025

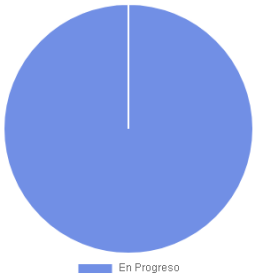
Nueva Cita | Nuevo Paciente | Nuevo Historial | Nuevo Pago

Citas Hoy	Pacientes	Tratamientos	Pagos Pendientes
0	1	0	$123,123.00

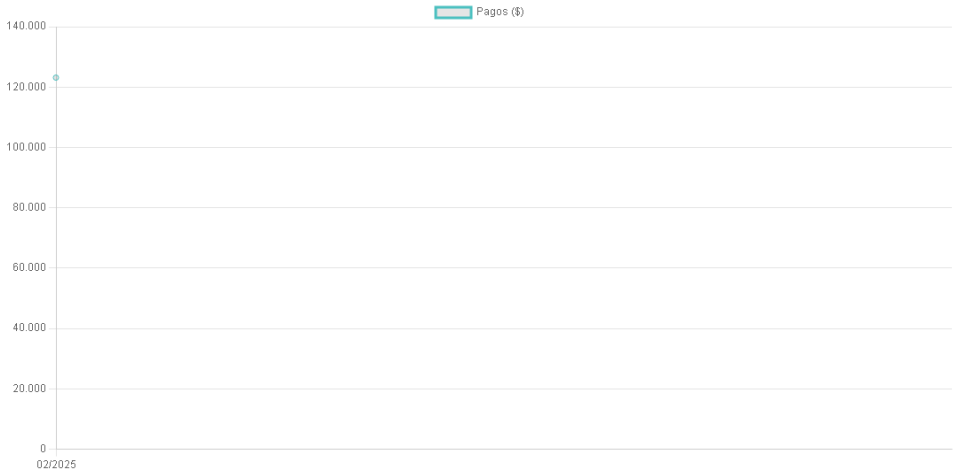
Citas por Día



Tratamientos por Estado



Pagos Mensuales



Ejemplo de los pacientes no pueden ingresar a ese apartado: