Manual técnico

SOFTWARE DENTIGRIN

Dentigrin es una solución integral para la administración de clínicas odontológicas, permitiendo la gestión eficiente de citas, pacientes, y expedientes médicos.

Características

- Gestión de citas
- Administración de perfiles de usuarios
- Panel de control para administradores
- Interfaz intuitiva y fácil de usar
- Generar reporte de historia clínica del paciente (Función a escalar con microservicios)
- Configuraciones del sistema (Servicio a escalar con arquitectura de microservicios)

Prerrequisitos

Requisitos Previos

- Node.js
- pnpm
- PostgreSQL

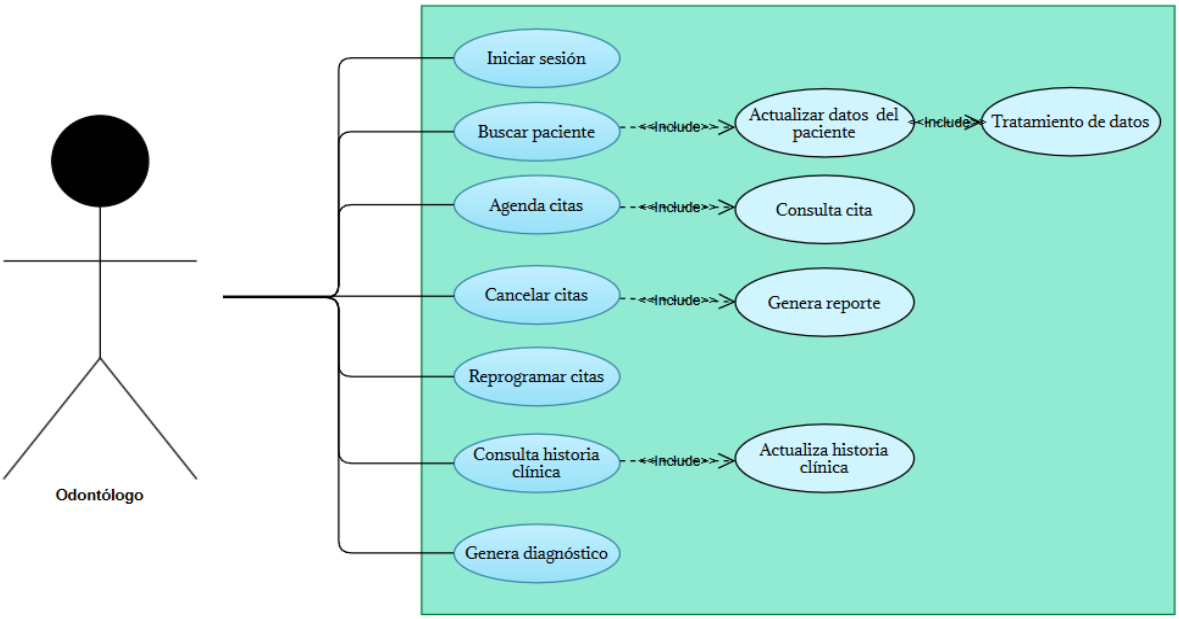
Frameworks y tecnologías utilizadas:

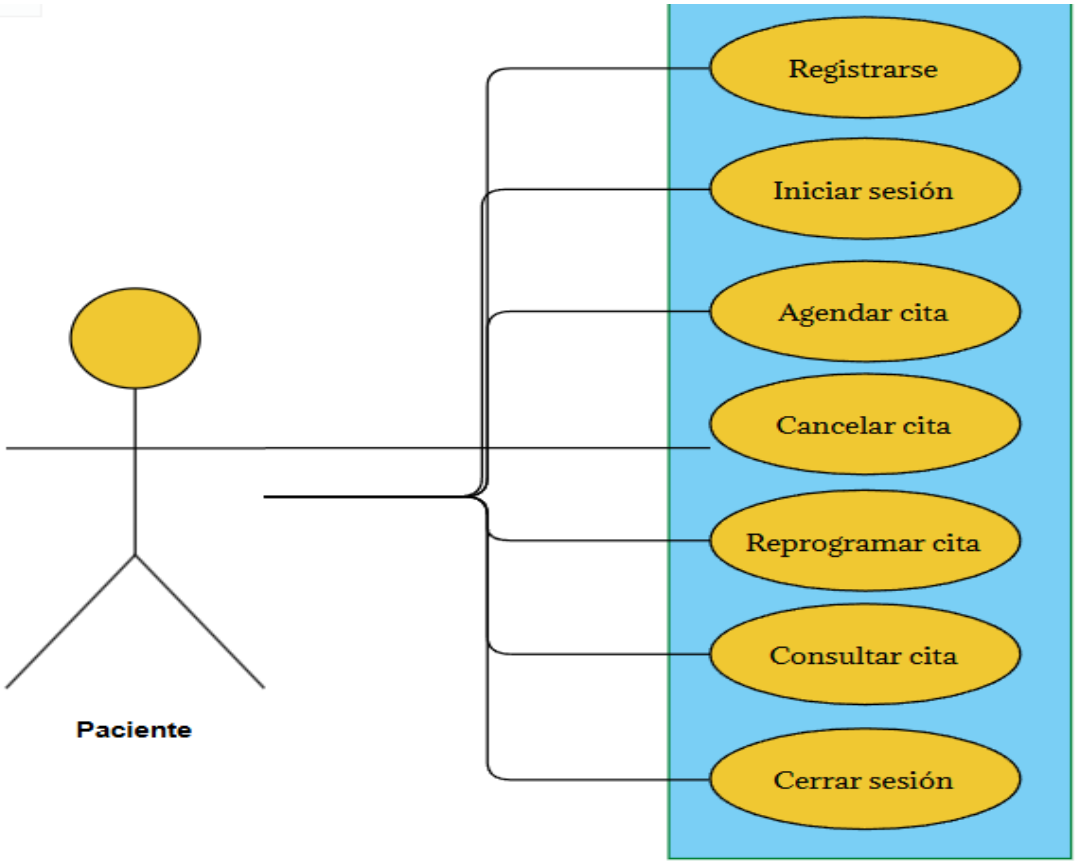
- React.js: Framework de JavaScript usado para construir la interfaz del usuario (front-end). Es ideal para crear aplicaciones interactivas y dinámicas.
- Vite: Herramienta de desarrollo que optimiza el proceso de construcción del front-end, permitiendo una carga más rápida y una experiencia de desarrollo más fluida.
- Node.js: Entorno de ejecución para JavaScript en el servidor (back-end). Permite manejar peticiones y respuestas del servidor.
- Express.js: Framework para Node.js que simplifica el desarrollo del back-end mediante una estructura ligera y funcional.
- PostgreSQL: Base de datos relacional utilizada para almacenar y administrar la información de manera segura y eficiente.
- Sequelize: ORM (Object-Relational Mapping) para interactuar con la base de datos PostgreSQL de manera más sencilla, utilizando objetos en lugar de consultas SQL puras.
- Docker: Herramienta de contenedorización que facilita la implementación del sistema en diferentes entornos, asegurando consistencia y portabilidad.
- Vercel: plataforma de despliegue en la nube que permite a los desarrolladores implementar aplicaciones web rápidamente, sin tener que preocuparse por la infraestructura subyacente. Es conocida por su optimización para aplicaciones front-end y su integración perfecta con frameworks como Next.js.
- Supabase: Plataforma de desarrollo backend open-source que permite crear aplicaciones de forma rápida y sencilla. Tiene herramientas para usar como: Base de datos, autenticaciones, almacenamientos, funciones serverless, APIs automaticas.
- Railway: plataforma en la nube que te permite desplegar, alojar y escalar tus aplicaciones y bases de datos de forma muy simple, sin importar la infraestructura.

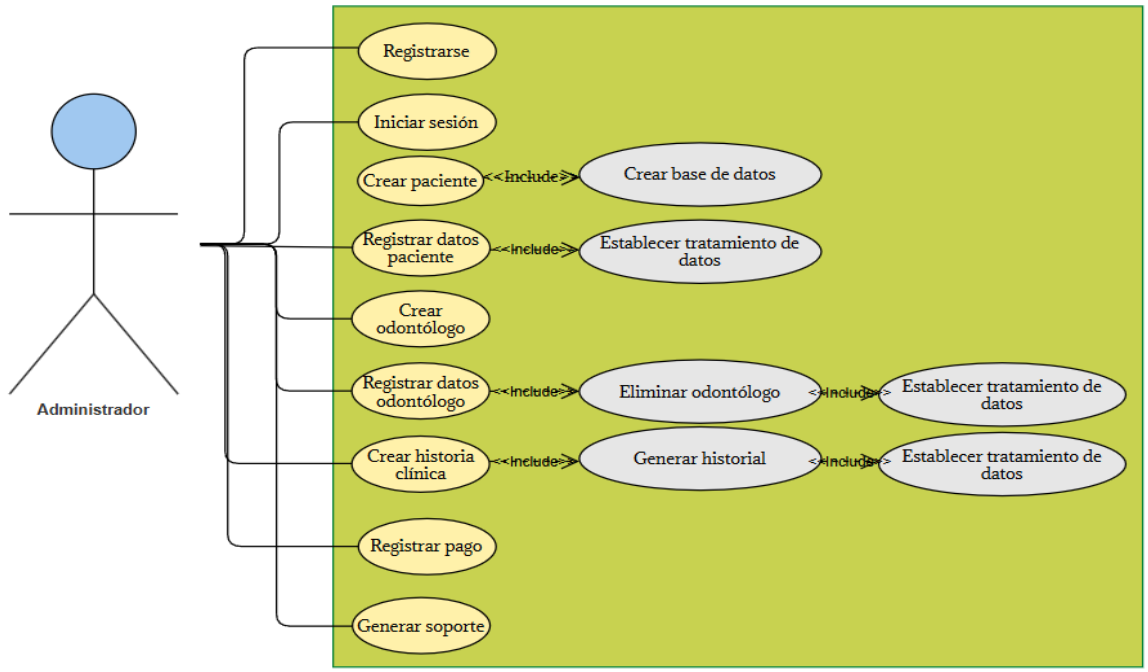
Estándares:

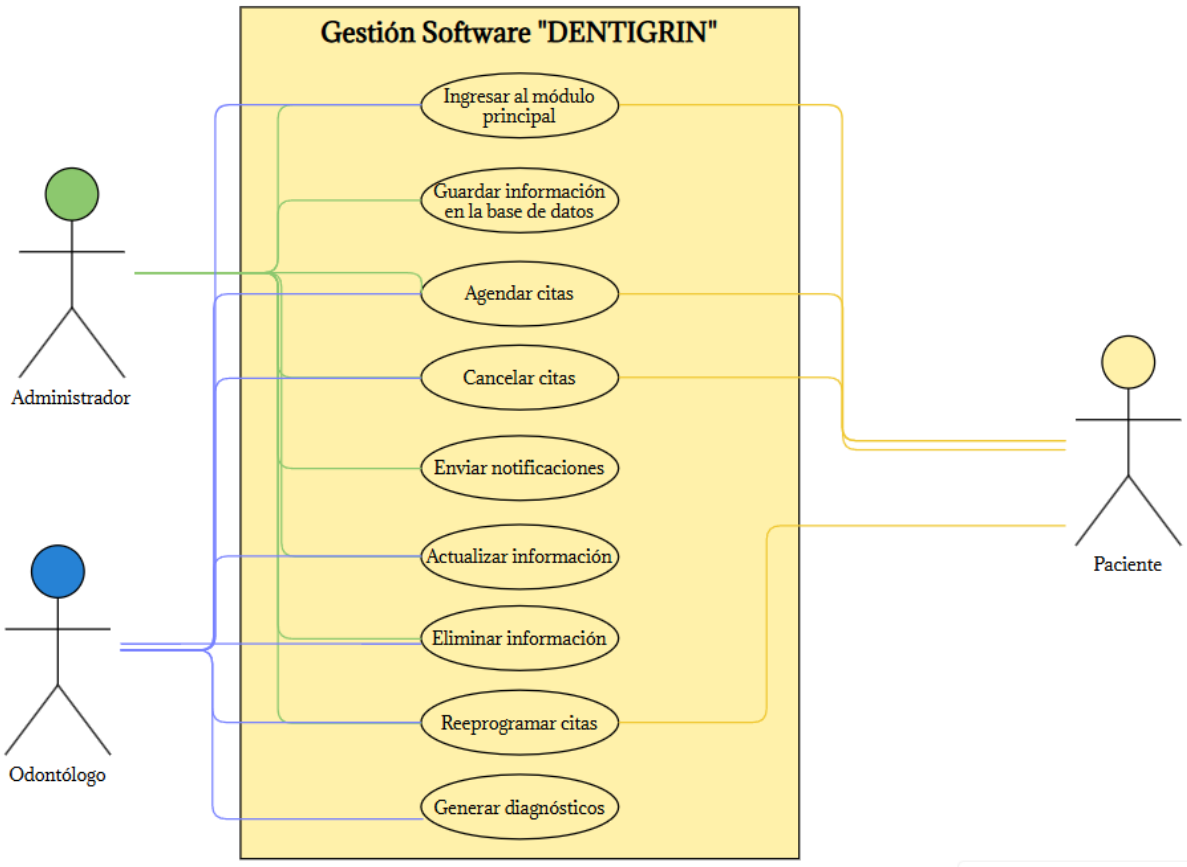
1. **Buenas prácticas de programación**: Código limpio, modular y reutilizable.
2. **Documentación adecuada**: Uso de README.md y otros archivos para detallar los pasos de instalación, configuración y uso del sistema.
3. **Versionamiento y control de código**: Utilización de Git/GitHub para la colaboración entre desarrolladores y la administración de cambios.
4. **Seguridad**: Implementación de estándares como la protección de datos sensibles y conexiones seguras (por ejemplo, HTTPS y cifrado).
5. **Accesibilidad**: Asegurar que la plataforma sea fácil de usar para diferentes tipos de usuarios.

Diagramas de caso de uso

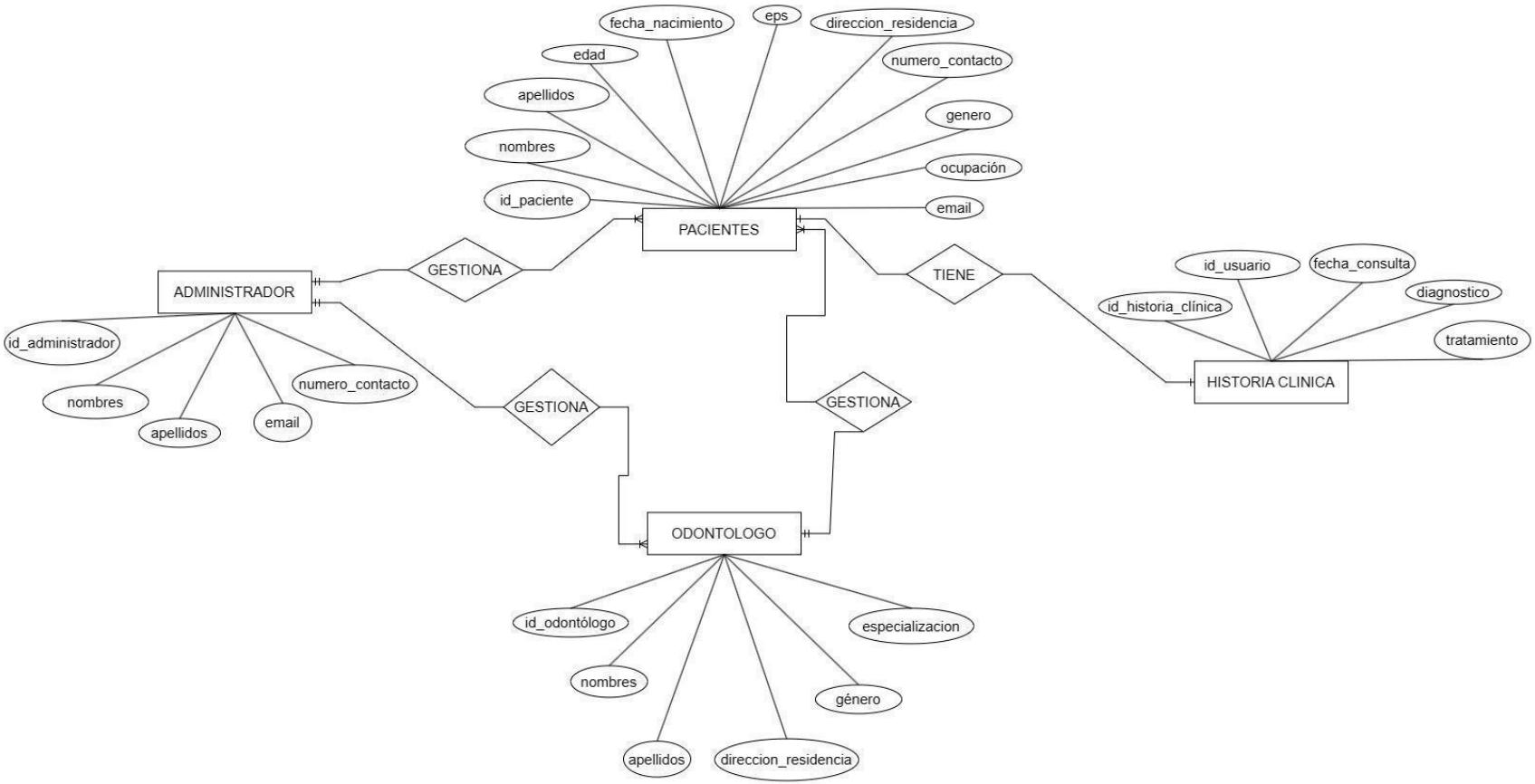




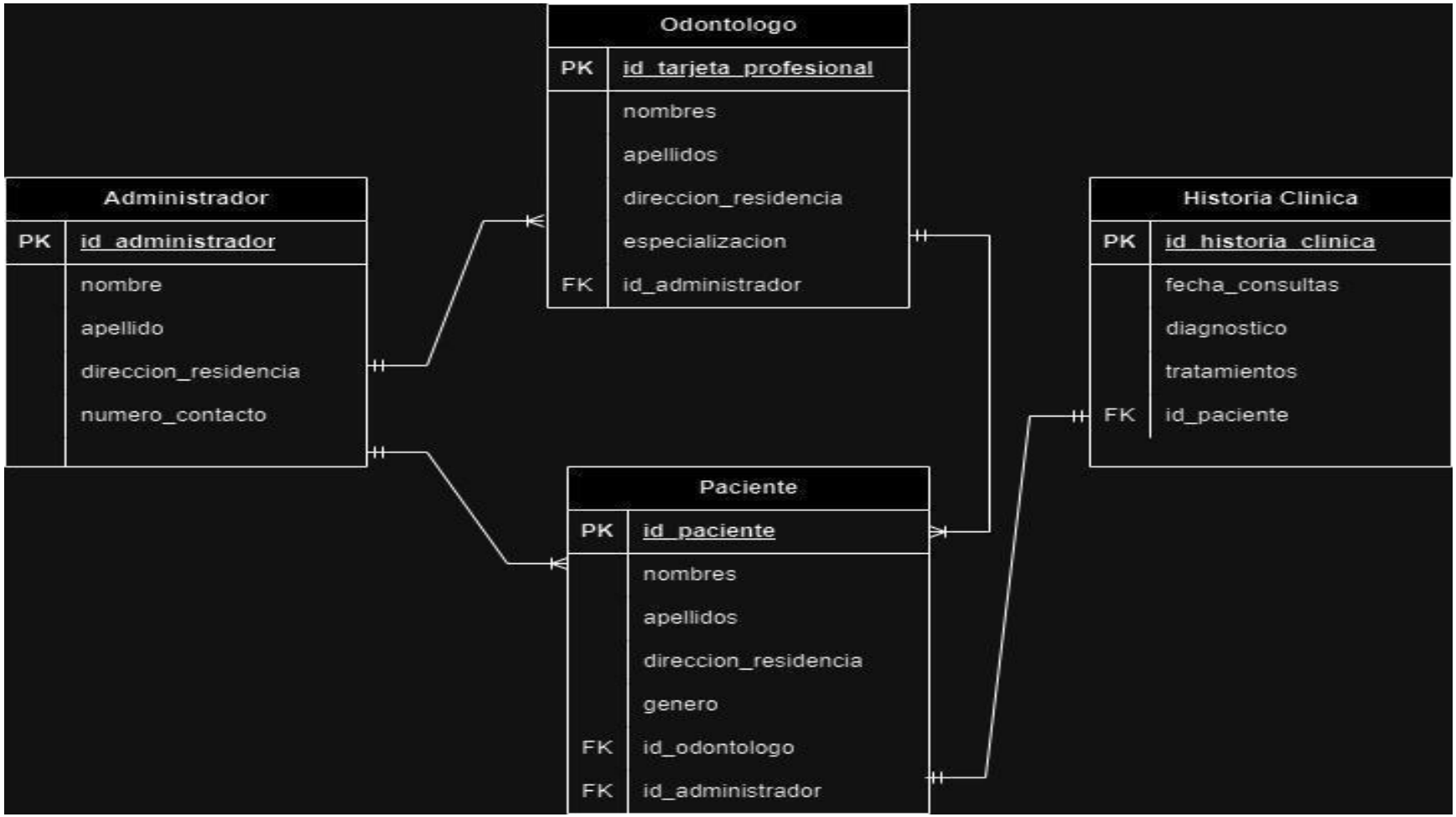

Administrador
Registrarse
Iniciar sesión
Crear paciente
<<Include>>
Crear base de datos
Registrar datos paciente
<<Include>>
Establecer tratamiento de datos
Crear odontólogo
Registrar datos odontólogo
<<Include>>
Eliminar odontólogo
<<Include>>
Establecer tratamiento de datos
Crear historia clínica
<<Include>>
Generar historial
<<Include>>
Establecer tratamiento de datos
Registrar pago
Generar soporte


Gestión Software "DENTIGRIN"
Ingresar al módulo principal
Guardar información en la base de datos
Agendar citas
Cancelar citas
Enviar notificaciones
Actualizar información
Eliminar información
Reeprogramar citas
Generar diagnósticos
Administrador
Odontólogo
Paciente

Modelo entidad relación de la base de datos (DER)



Modelo Lógico



Diccionario de datos

Diccionario de datos según el modelo lógico.

Diccionario: Pacientes	
id_paciente	**Número de identificación del paciente.**
nombres	**Nombres completos del paciente.**
apellido	**Apellidos completos del paciente.**
edad	**Edad del paciente.**
fecha_ nacimiento	**Fecha de nacimiento del paciente.**
eps	**Entidad Prestadora de salud del paciente.**
dirección_residencia	**Dirección actual donde vive el paciente.**
numero_contacto	**Número del teléfono actual del paciente.**
género	**Género del paciente (femenino, masculino, otros)**
Ocupación	**Actividad que ejerce el paciente.**
email	**Email del paciente.**

Diccionario: Odontólogo	
id_odontólogo	**Número de identificación del odontólogo.**
nombres	**Nombres completos del odontólogo.**
apellidos	**Apellidos completos del odontólogo.**
dirección_residencia	**Dirección actual donde vive el odontólogo.**
género	**Género del odontólogo (femenino, masculino, otros)**
especialización	**Especialidad del odontólogo (ortodoncia, odontología general, periodoncista, cirujano oral "general, maxilofacial", endodoncista)**

Diccionario: Administrador	
id_administrador	**Número de identificación del Administrador.**
nombres	**Nombres completos del Administrador.**
apellidos	**Apellidos completos del Administrador.**
correo	**Email del Administrador.**
numero_contacto	**Número del teléfono actual del Administrador.**

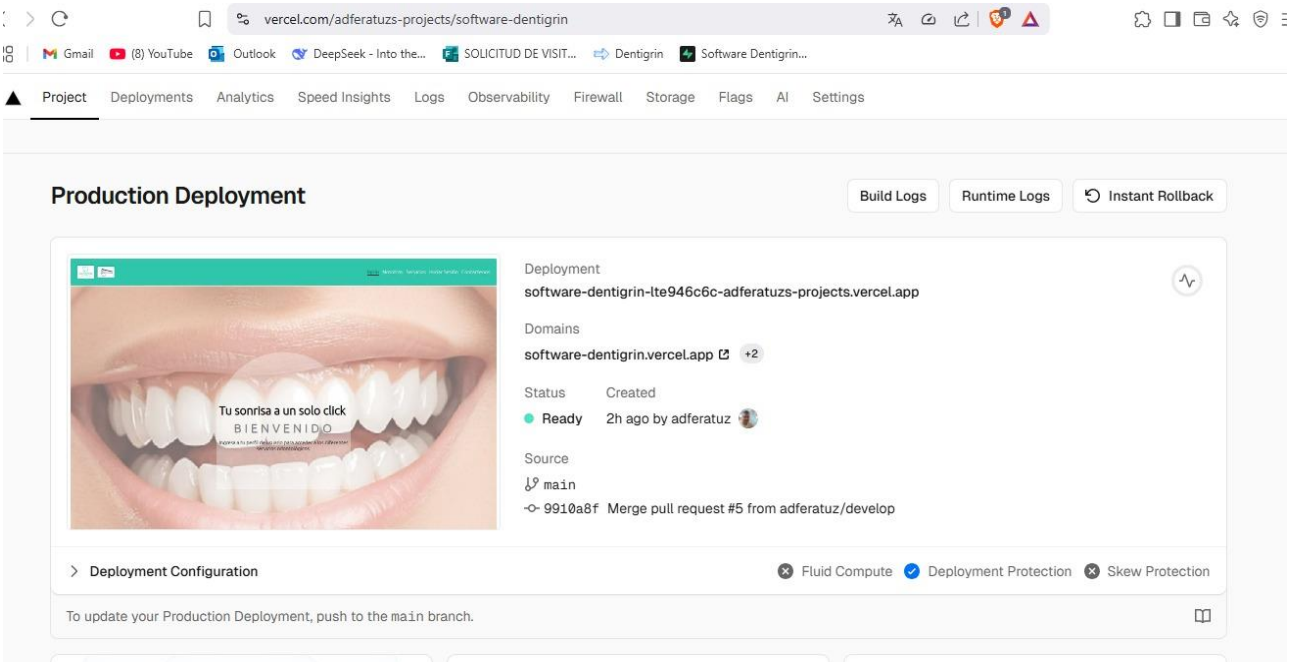
Diccionario: Historia Clínica	
id_historia_clínica	**Identificador único de la historia clínica del paciente.**
id_usuario	**Número de identificación del usuario.**
fecha_consulta	**Fecha en que se consulta la historia clínica.**
diagnóstico	**El diagnóstico realizado durante la consulta.**
tratamientos	**Los tratamientos recomendados o los tratamientos realizados a los pacientes.**

Scripts de instalación

Front-end:

Link del paso a paso del despliegue de la aplicación por lado del cliente en vercel:

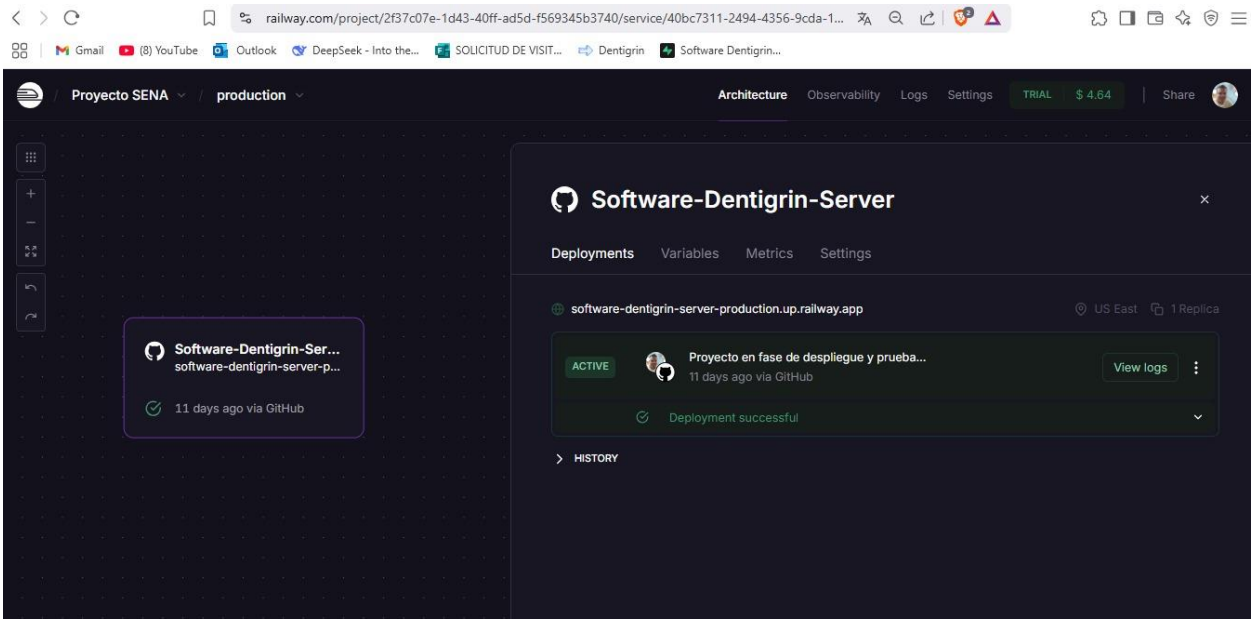
https://youtu.be/fZa0_OML1Zc



Back-end

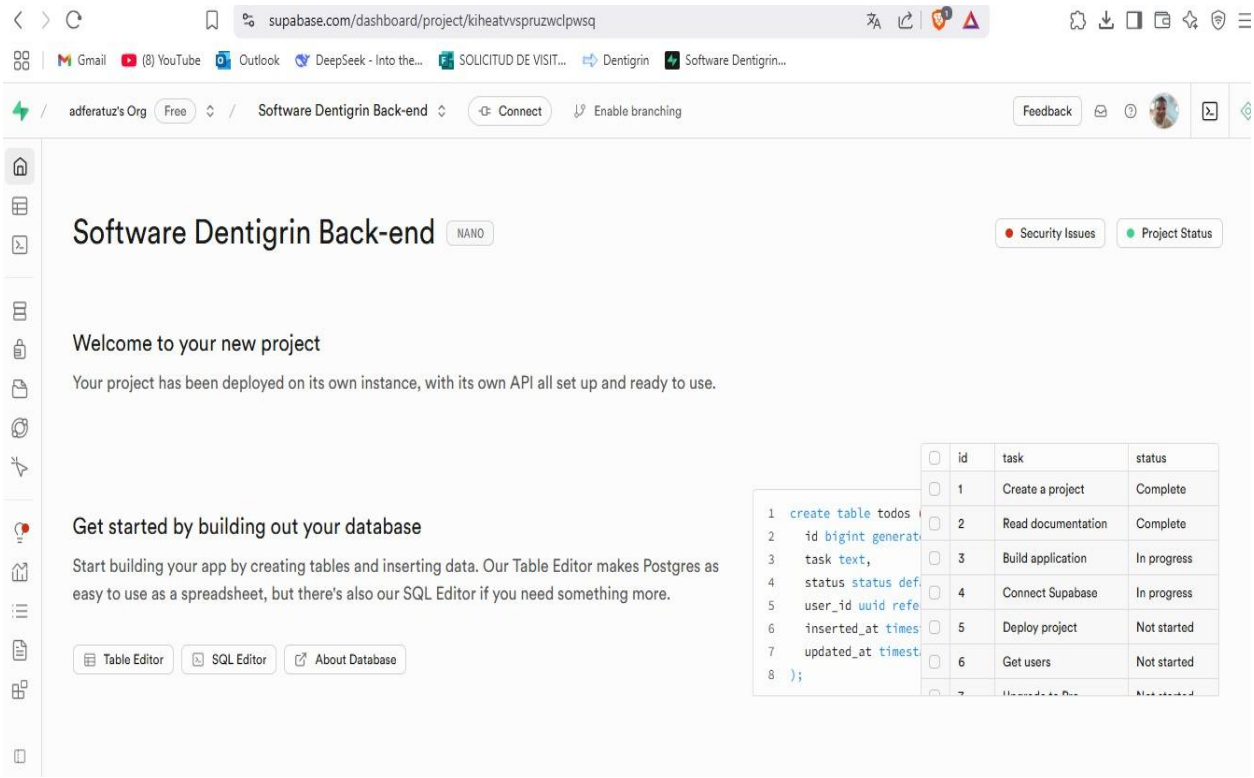
Link del despliegue del servidor en Railway:

https://youtu.be/_qMEFWmKr_I



Enlace del alojamiento de la base de datos en Supabase:

https://youtu.be/kiQ0TeclesA



DDL BASE DE DATOS DENTIGRIN

Creación tabla usuarios:

```
CREATE TABLE IF NOT EXISTS usuarios
(
    id_usuario SERIAL,
    username character varying(255) NOT NULL,
    email character varying(255) NOT NULL,
    rol character varying(255) NOT NULL,
    "fecha_creacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    "fecha_actualizacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT usuarios_pkey PRIMARY KEY (id_usuario)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS usuarios
    OWNER to postgres;
```

Creacion tabla pacientes:

```
CREATE TABLE IF NOT EXISTS pacientes
(
    id_paciente SERIAL,
    id_usuario integer NOT NULL,
    nombres character varying(255) NOT NULL,
    apellidos character varying(255) NOT NULL,
    edad integer NOT NULL,
    fecha_nacimiento character varying(255) NOT NULL,
    eps character varying(255) NOT NULL,
    direccion_residencia character varying(255) NOT NULL,
    numero_contacto character varying(255) NOT NULL,
    genero character varying(255) NOT NULL,
    ocupacion character varying(255) NOT NULL,
    "fecha_creacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    "fecha_actualizacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT pacientes_pkey PRIMARY KEY (id_paciente),
    CONSTRAINT pacientes_id_usuario_fkey FOREIGN KEY (id_usuario)
        REFERENCES usuarios (id_usuario) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS pacientes
    OWNER to postgres;
```

Creacion tabla odontologos:

```
CREATE TABLE IF NOT EXISTS odontologos
(
    id_odontologo SERIAL,
    id_usuario integer NOT NULL,
    nombres character varying(255) NOT NULL,
    apellidos character varying(255) NOT NULL,
    direccion_residencia character varying(255) NOT NULL,
    genero character varying(255) NOT NULL,
    especializacion character varying(255) NOT NULL,
    "fecha_creacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    "fecha_actualizacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT odontologos_pkey PRIMARY KEY (id_odontologo),
    CONSTRAINT odontologos_id_usuario_fkey FOREIGN KEY (id_usuario)
        REFERENCES public.usuarios (id_usuario) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS odontologos
    OWNER to postgres;
```

Creacion tabla administradores:

```
CREATE TABLE IF NOT EXISTS administradores
(
    id_administrador SERIAL,
    id_usuario integer NOT NULL,
    nombres character varying(255) NOT NULL,
    apellidos character varying(255) NOT NULL,
    numero_contacto character varying(255) NOT NULL,
    "fecha_creacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    "fecha_actualizacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT administradores_pkey PRIMARY KEY (id_administrador),
    CONSTRAINT administradores_id_usuario_fkey FOREIGN KEY (id_usuario)
        REFERENCES public.usuarios (id_usuario) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS administradores
    OWNER to postgres;
```

Creacion tabla historias_clinicas:

```
CREATE TABLE IF NOT EXISTS historias_clinicas
(
    id_historia_clinica SERIAL,
    id_paciente integer NOT NULL,
    fecha_consulta character varying(255) NOT NULL,
    diagnostico character varying(255) NOT NULL,
    tratamientos character varying(255) NOT NULL,
    "fecha_creacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    "fecha_actualizacion" timestamp DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT historias_clinicas_pkey PRIMARY KEY (id_historia_clinica),
    CONSTRAINT historias_clinicas_id_paciente_fkey FOREIGN KEY (id_paciente)
        REFERENCES pacientes (id_paciente) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS historias_clinicas
    OWNER to postgres;
```

DML base de datos dentigrin:

Tabla usuarios:

```sql
INSERT INTO usuarios(
    username, email, rol)
    VALUES ('pepito', 'pepe.perez@ejemplo.com', 'USER'),
           ('juana.florez','juana@otroejemplo.com','USER'),
            ('filadelfio','fila.del@ejemplo.com','ADMIN'),
            ('Jose.med','Jose.med@ejemplo.com','ODONTOLOGO')


SELECT * FROM usuarios;

SELECT id_usuario, username, email, rol
    FROM usuarios;

UPDATE usuarios
    SET username='pedro.perez'
    WHERE id_usuario= 1;

DELETE FROM usuarios
    WHERE id_usuario =2;
```

Tabla pacientes:

```sql
INSERT INTO pacientes
(
    id_usuario,
    nombres,
    apellidos,
    edad,
    fecha_nacimiento,
    eps,
    direccion_residencia,
    numero_contacto,
    genero,
    ocupacion
)
    VALUES (7, 'Pedro', 'Florez', 45, '24/03/1979', 'Servisalud', 'AV Esperanza Diagonal 3', '3005278854', 'M', 'Independiente')

SELECT * FROM pacientes;

SELECT nombres, apellidos, edad, fecha_nacimiento, eps, direccion_residencia, numero_contacto, genero, ocupacion
    FROM pacientes;

UPDATE pacientes
    SET nombres='Pedro Manuel', numero_contacto='3105554545'
    WHERE id_paciente= 1;

DELETE FROM pacientes
    WHERE id_paciente= 1;
```

Tabla odontólogos:

```sql
INSERT INTO odontologos
(
    id_usuario,
    nombres,
    apellidos,
    direccion_residencia,
    genero,
    especializacion
)
    VALUES (13, 'Jose', 'Peña', 'Conjunto flores apto 103', 'M', 'ODONTOLOGO');

SELECT * FROM odontologos;

SELECT nombres, apellidos, direccion_residencia, genero, especializacion
    FROM odontologos;

UPDATE odontologos
    SET direccion_residencia='Avenida Siempre viva conjunto florez apto 103'
    WHERE id_odontologo= 1;

DELETE FROM odontologos
    WHERE apellidos= 'Peña';
```

Tabla administradores:

```sql
INSERT INTO administradores
(
    id_usuario,
    nombres,
    apellidos,
    numero_contacto
)
    VALUES (12, 'Fernando', 'Hernadez', '3158796548');

SELECT * FROM administradores;

SELECT nombres, apellidos, numero_contacto
    FROM administradores;

UPDATE administradores
    SET numero_contacto='3209002587'
    WHERE id_administrador= 1;

DELETE FROM administradores
    WHERE id_administrador = 1;
```

Tabla historias_clinicas:

```sql
INSERT INTO historias_clinicas
(
    id_paciente,
    fecha_consulta,
    diagnostico,
    tratamientos
)
    VALUES (3, '28/07/2024', 'Limpieza oral', 'Limpieza profunda por caries');

SELECT * FROM historias_clinicas;

SELECT fecha_consulta, diagnostico, tratamientos
    FROM historias_clinicas;

UPDATE public.historias_clinicas
    SET tratamientos='Calza de maxilares'
    WHERE id_historia_clinica =1;

DELETE FROM public.historias_clinicas
    WHERE id_historia_clinica =1;
```

Diagrama de componentes