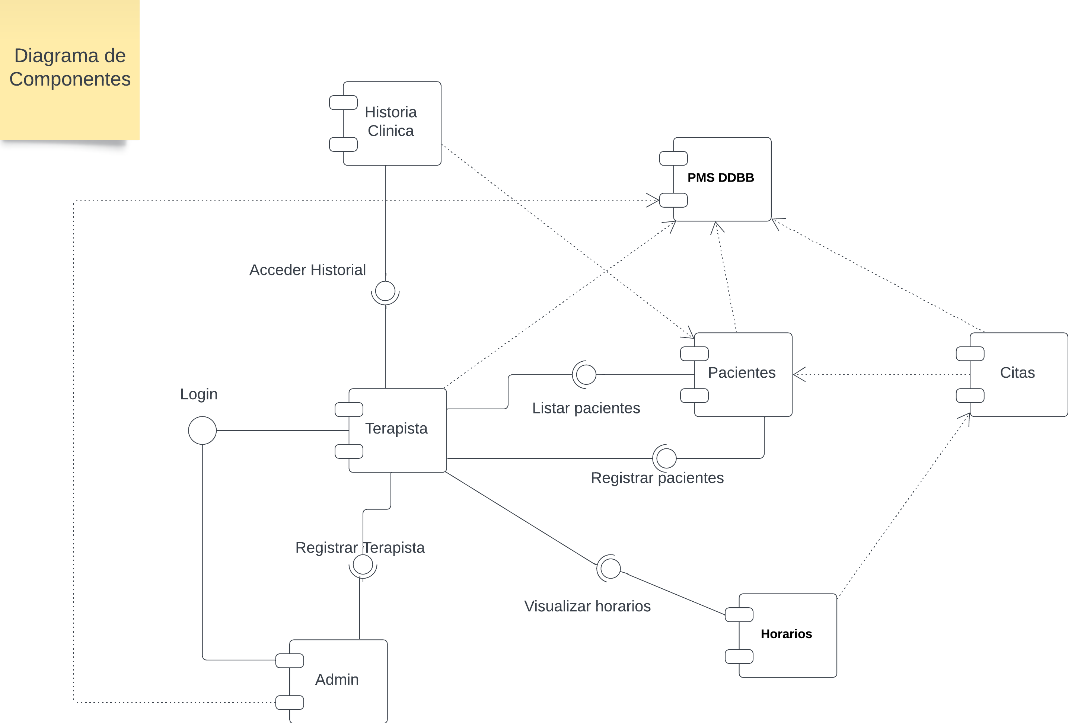
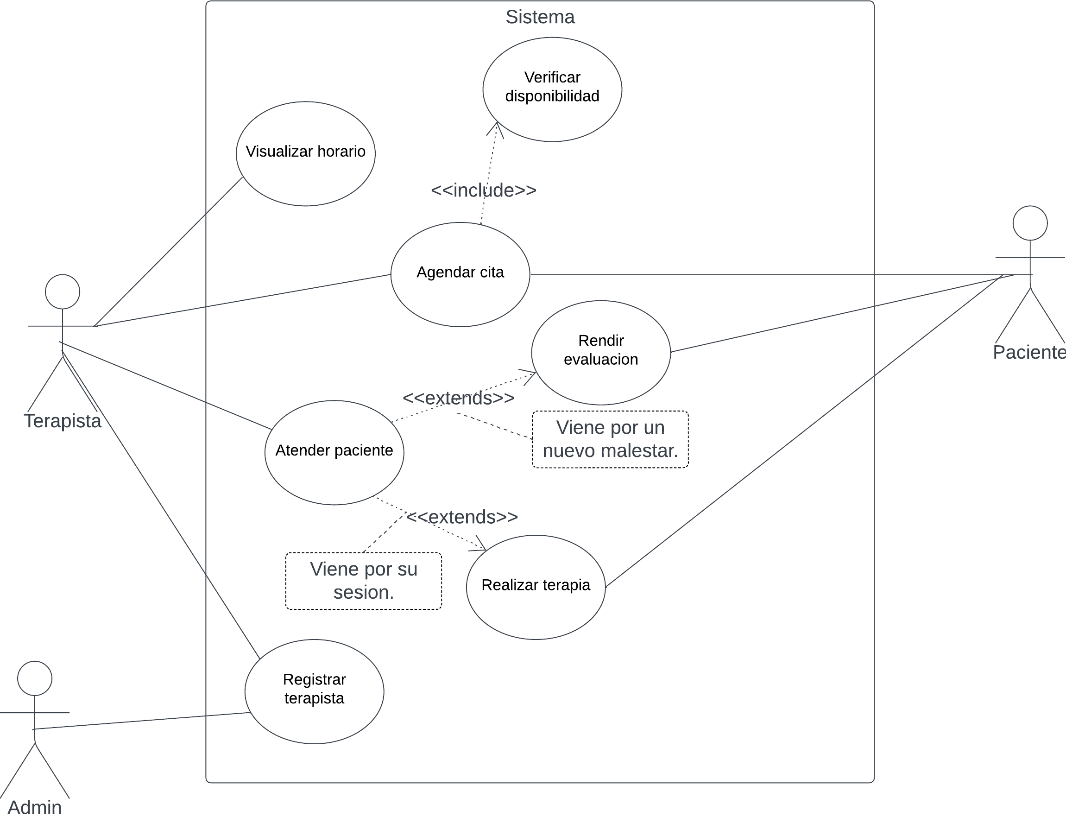
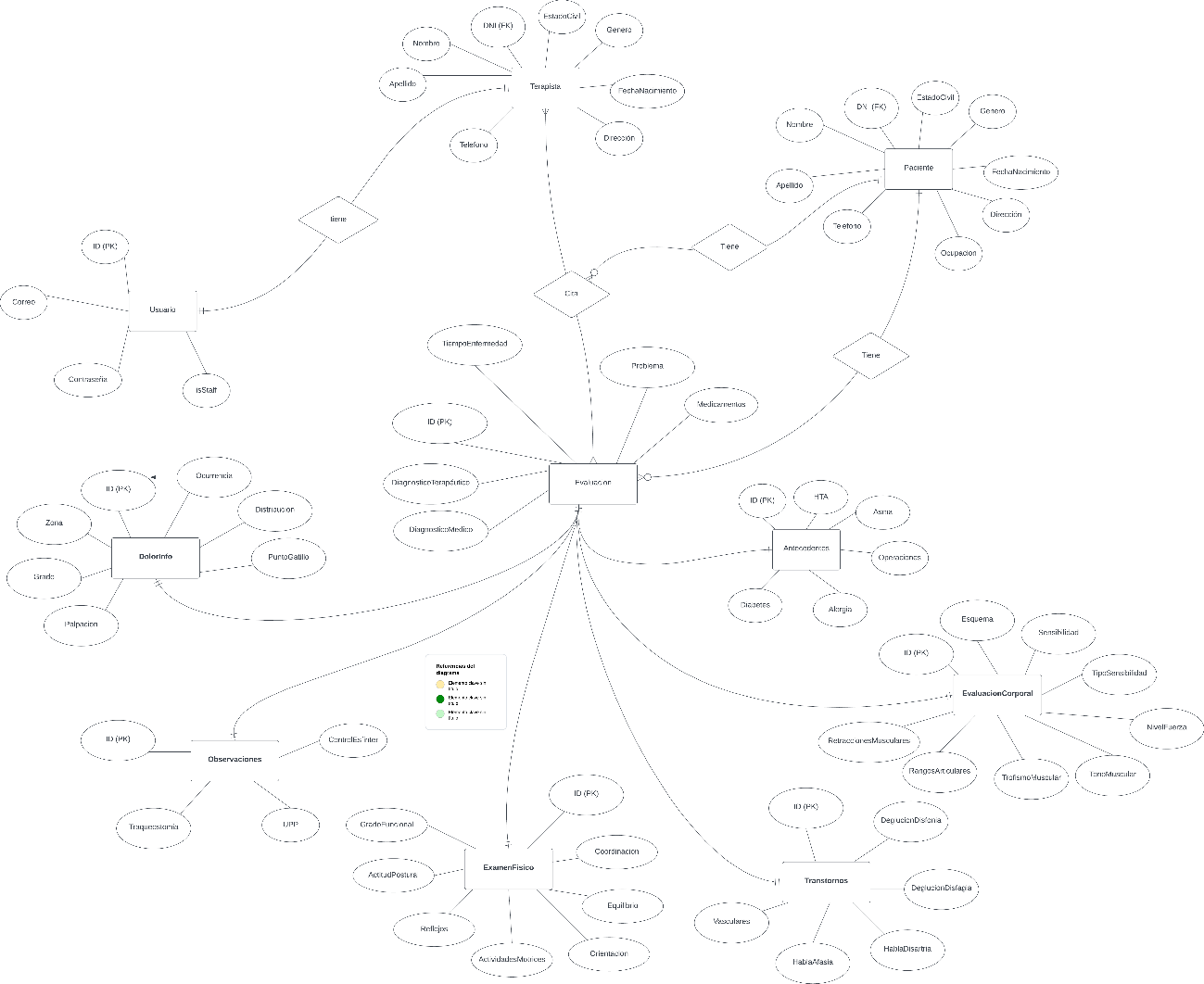
**DOCUMENTACIÓN DEL BACKEND DEL SOFTWARE “REHAVITAL’S - SISTEMA GESTIÓN PACIENTES”**

1. **Introducción:**
   * **Propósito del Documento:** Este documento tiene como objetivo principal proporcionar una guía completa y detallada sobre el desarrollo, implementación y funcionamiento del sistema de gestión para pacientes desarrollado para la empresa REHAVITAL’S. Está dirigido a los desarrolladores, ingenieros de software, y cualquier persona involucrada en el mantenimiento y evolución del sistema. Además, se pretende que sirva como recurso informativo para aquellos que deseen comprender la arquitectura y la funcionalidad del backend de la aplicación.
   * **Contexto del Proyecto:** La empresa REHAVITAL’S enfrenta desafíos en la gestión de pacientes, ya que actualmente depende de registros en papel y agendas para mantener un historial de pacientes y coordinar horarios médicos. Este sistema busca abordar esta problemática mediante el desarrollo de un sistema de gestión que permita programar citas, coordinar la disponibilidad de horarios de terapeutas físicos, establecer historiales médicos y simplificar el registro de pacientes.
2. **Arquitectura del Sistema:**
   * **Diagrama de Componentes:**



* + **Diagrama de Casos de Uso**

****

* + **Diagrama de Entidades:**

1. **Requerimientos del Sistema:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Requerimiento | Requerimiento | Prioridad | **Objetivo** |
| Funcional | El sistema recogerá los datos personales de los pacientes | ALTA | Obtener los datos necesarios de los pacientes |
| Se deberá permitir el acceso de los usuarios a la plataforma | MEDIA | Facilitar la interacción entre los usuarios en el sistema |
| Los trabajadores visualizarán todas las citas pendientes programadas para el día | ALTA | Mantener el orden de trabajo del personal de salud |
| Seguimiento del tratamiento del paciente y su evolución en el tiempo | MEDIA | Mantener un historial por cada paciente |
| No Funcional | Los datos de todos los usuarios estarán almacenados en una BBDD | ALTA | Integridad, perduración y facil acceso de los datos |
| Se usarán 3 aplicaciones Django: una para el módulo de citas, otra para los usuarios y otra para las evaluaciones/historiales clínicos. | MEDIA | Mantener un orden a la hora de programar los modelos. |
| Se lanzará una API web para el frontend pueda realizar un CRUD con los datos. | ALTA | Conectar la interfaz gráfica del frontend con la base de datos. |

1. **Configuración del Entorno:**

Para el desarrollo del backend se está empleando el lenguaje de programación Python junto a sus 2 framworks django y django rest framework.

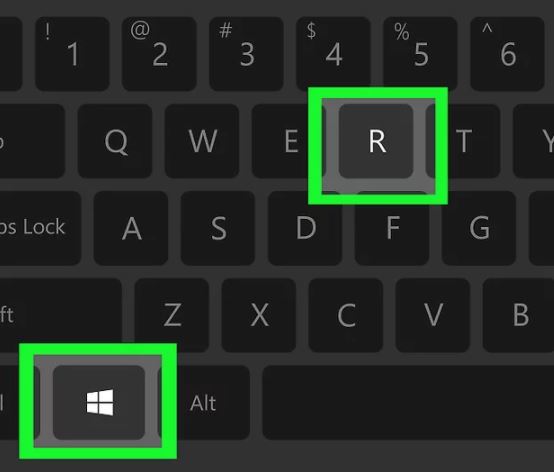
El gestor de base de datos a utilizar es SQLite 3 ya que este es el gestor con el que django trabaja de manera predeterminada. Sin embargo, para el despliegue se realizará una migración a PostgreSQL.

* + **Requisitos de Software:**

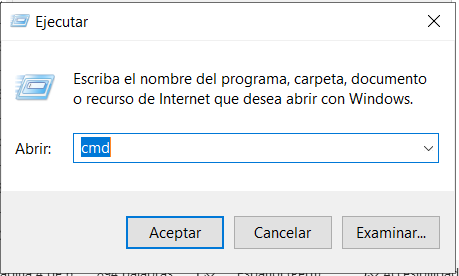
Para este proyecto usaremos la versión de Python 3.11.5 y la versión de pip 23.2.1

En Windows, para saber si tenemos instaladas las versiones de pip y Python que necesitamos solo tenemos que abrir el símbolo del sistema (CMD) e insertar los comandos correspondientes.

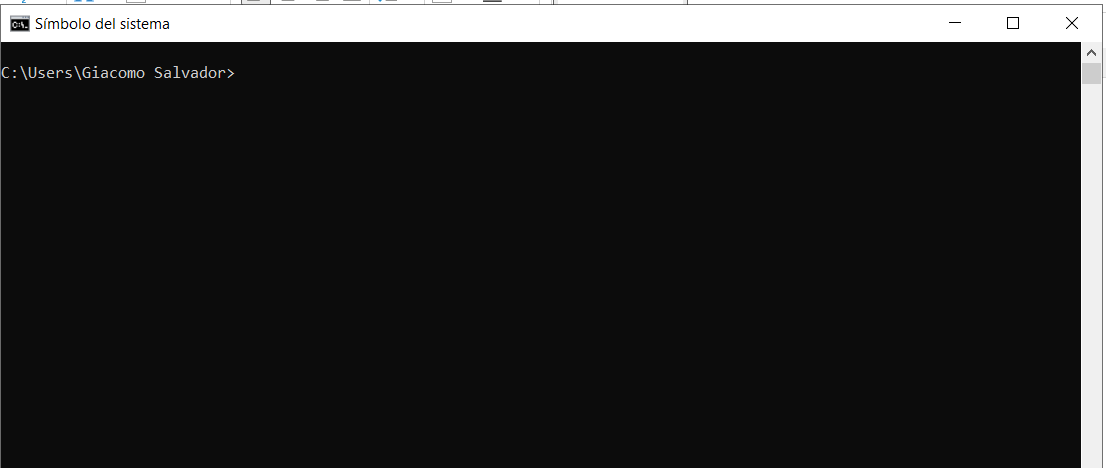
Para abrir el símbolo del sistema presionamos la combinación de teclas “Windows + R”:



Y acto seguido escribimos “cmd” y presionamos la tecla ENTER.



Se abrirá el símbolo del sistema.



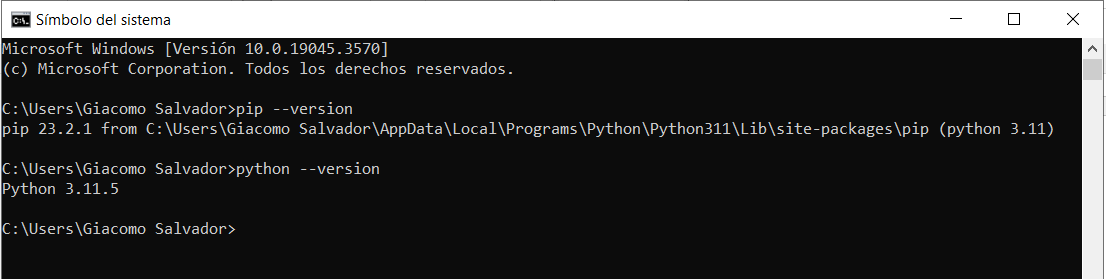
Escribimos los siguientes comandos y presionamos la tecla ENTER:

Para verificar la versión de pip:

pip --version

Para verificar la versión de Python:

python --versión



En caso de no tener instalados Python ni pip pueden descargarse gratuitamente desde su sitio web oficial:

Python: <https://www.python.org/downloads/>

Una vez que tengas Python instalado es probable que ya tengas pip, pero en caso de no ser así puedes seguir los siguientes pasos:

1. **Descarga el Instalador de ‘get-pip.py’:** Abre tu navegador web y dirígete a <https://bootstrap.pypa.io/get-pip.py>. Haz clic derecho en la página y selecciona "Guardar como..." para descargar el archivo ‘**get-pip.py’** en tu computadora.
2. **Ejecuta el Instalador de get-pip.py:** Abre la línea de comandos y navega al directorio donde guardaste **get-pip.py**. Luego, ejecuta el siguiente comando:

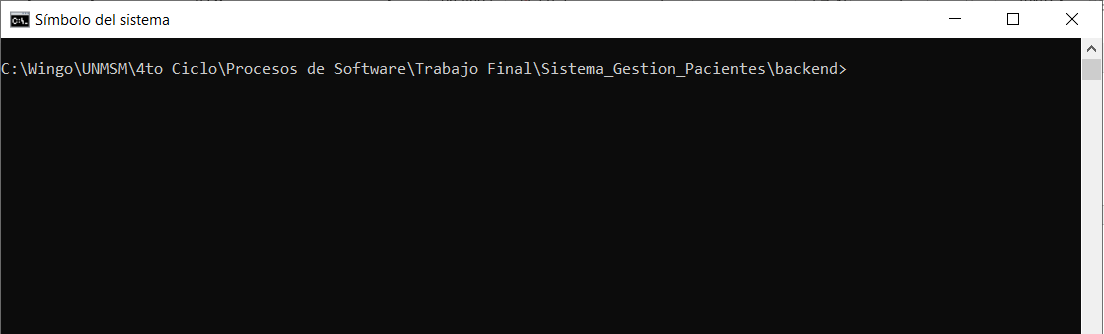
python get-pip.py

1. **Verifica la Instalación de pip:** Después de la instalación, verifica que **pip** se haya instalado correctamente:

pip --version

Esto debería mostrar la versión de **pip** que se instaló.

* + **Instrucciones de Configuración:** Una vez que tenemos Python y pip instalados procedemos a crear el entorno virtual, esto se puede hacer desde el cmd o desde el terminal del IDE que desee utilizar. Recomendamos usar [Visual Studio Code](https://code.visualstudio.com/download), sin embargo, no es obligatorio.
    1. En la consola del símbolo del sistema nos dirigimos a la carpeta del proyecto y entramos al directorio “backend”:



* + 1. Procedemos a escribir el siguiente comando:

pip install virtualenv

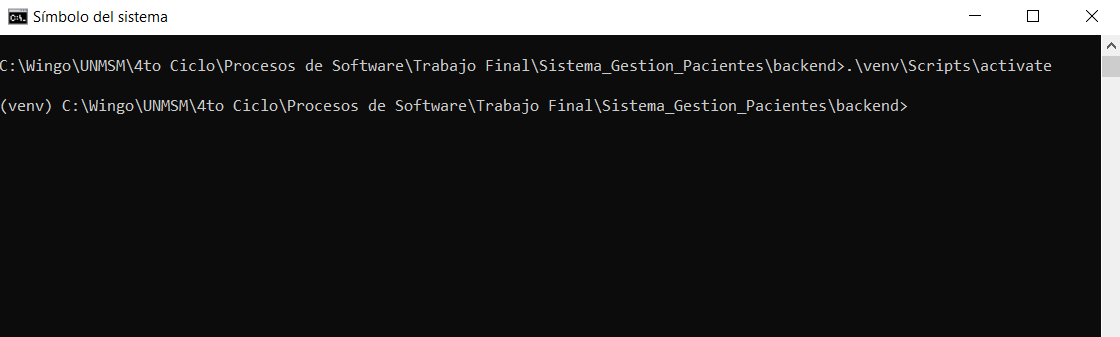
* + 1. Una vez instalado creamos el entorno virtual con el siguiente comando:

virtualenv venv

* + 1. Activamos el entorno virtual con el siguiente comando:

.\venv\Scripts\activate

Nota: Es importante recalcar que esto se deberá hacer desde el cmd, ya que para usar Powershell habría que hacer una seria de configuraciones adicionales que incluyen ejecutarlo como administrador.

* + 1. Si hicimos todo correctamente habremos activado el entorno virtual de manera satisfactoria:
    2. Una vez dentro del entorno virtual procedemos a instalar todas las dependencias y requerimientos con la siguiente línea de comandos:

pip install django django-filter djangorestframework gunicorn Markdown coreapi django-cors-headers

* + 1. De este modo, ya tendremos instaladas todas las dependencias necesarias para el desarrollo y mantenimiento del backend.
    2. Para levantar el servidor de manera local, solo hace falta escribir el siguiente comando:

python manage.py runserver

* + 1. Para salir del entorno virtual se escribe el siguiente comando:

.\venv\Scripts\deactivate

Nota: Los ‘\’ pueden cambiar por ‘/’ según el terminal o consola que se esté utilizando, esto aplica para todos los comandos que los incluyan.

1. **Estructura del Proyecto Django:**
   * **Explicación de la Estructura de Directorios:**

En Django, la estructura de directorios sigue un patrón específico para organizar el código y los recursos del proyecto:

**Sistema\_Gestion\_Pacientes/backend/backend/** (Nombre del Proyecto, en este caso lo nombramos “backend”):

Contiene archivos de configuración global del proyecto.

Ejemplos: settings.py (configuraciones del proyecto), urls.py (configuración de las URLs), asgi.py y wsgi.py (para ASGI y WSGI, respectivamente).

**/migrations/:**

Contiene archivos de migraciones generados automáticamente por Django cuando se realizan cambios en los modelos.

Estos archivos describen cómo modificar la base de datos para reflejar los cambios en el modelo.

**/venv/:**

Directorio virtualenv que contiene las dependencias específicas del proyecto.

No se versiona y se añade al archivo .gitignore o similar para evitar subirlo al control de versiones.

**/manage.py:**

Script de gestión para realizar tareas administrativas y ejecutar comandos específicos de Django.

**/db.sqlite3:**

Archivo de base de datos predeterminado que Django utiliza cuando se configura para usar SQLite como motor de base de datos. Este archivo es creado automáticamente por Django cuando ejecutas las migraciones por primera vez o cuando se usa el comando *migrate* para aplicar cambios en el esquema de la base de datos.

* + **Módulos y Aplicaciones:** El proyecto se divide en 2 módulos principales: el de gestión de citas y el de historial clínico, cada uno representado por una aplicacipn django:
    1. **Sistema\_Gestion\_Pacientes/backend/administracion\_de\_citas/:**

En esta aplicación se encuentran los modelos que tienen que ver con el módulo de creación y administración de una cita.

Los modelos que se encuentran son:

- Paciente

- Terapista

- Cita

* + 1. **Sistema\_Gestion\_Pacientes/backend/historial\_medico/:**

En esta aplicación se encuentran los modelos que tienen que ver con el módulo de historial clínico, así como aquellos que representan las dolencias y las enfermedades. Estos modelos fueron elegidos según los apartados de la hoja de evaluación entregada por el cliente.

Los modelos que se encuentran son:

- Trastornos

- ExamenFisico

- Observaciones

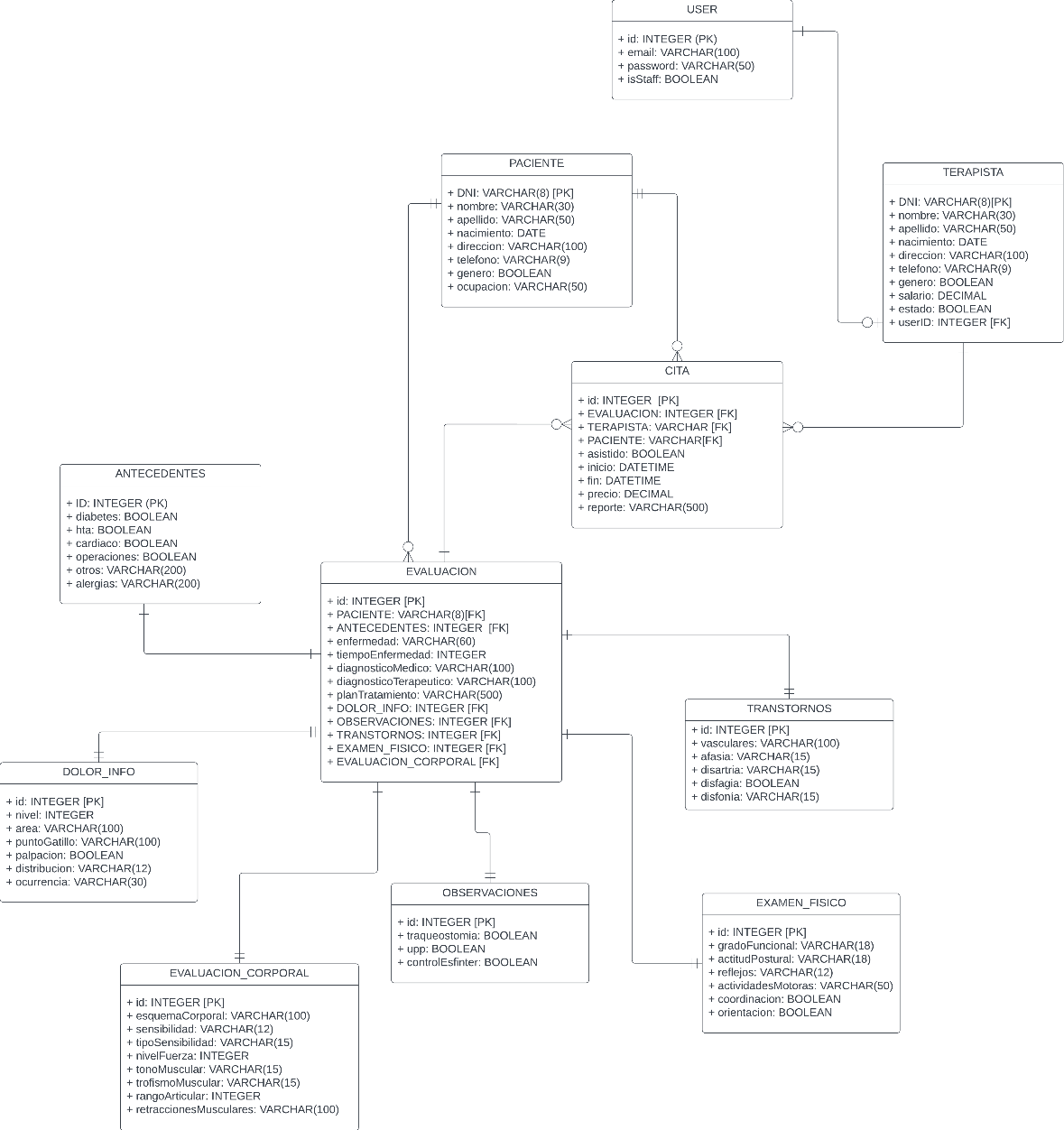
- EvaluacionCorporal

- DolorInfo

- Antecedentes

- Evaluacion

1. **Modelos de Datos:**
   * Diagrama de Modelo Lógico:



1. **API y Endpoints:**
2. **Despliegue:**
3. **Conclusiones:**