Documentación técnica

PLENNA

02/06/2022

Índice

Introducción

El propósito del presente documento es dar a conocer las distintas entregas del proyecto, así como divulgar el uso adecuado de estas mismas. El apartado de **Repositorios Github** contiene los enlaces para acceder a todos los archivos de código que componen el proyecto. Posteriormente se encuentran dos tipos de instructivos, primero el **Instructivo para ejecutar el proyecto de manera local** que enlista las herramientas de software y los datos necesarios para acceder la base de datos. En segundo lugar el **Instructivo para realizar pruebas** que contiene varias explicaciones de lo que hace cada una de las pruebas así como los pasos a seguir para ejecutarlas. Por último se encuentran los diagramas de entidad relación, de secuencia y los diagramas de flujo.

Repositorios Github

El repositorio de github contiene las siguientes secciones y se puede acceder con el link proporcionado.

.vscode	avances mañana 23/5	16 days ago
back	corrección actualización	44 minutes ago
front	html y css de algunos botones y el calendario	13 days ago
plenna_env	virtual environment	21 hours ago
sql	corrección actualización	44 minutes ago
Diagrama secuencia proyecto Plen...	Add files via upload	3 hours ago
PruebasPlenna.sql	Create PruebasPlenna.sql	1 minute ago
dir_ip.txt	Update dir_ip.txt	21 days ago
entidad-relacion.png	back2doctor	2 days ago
labsuser.ppk	Add files via upload	28 days ago

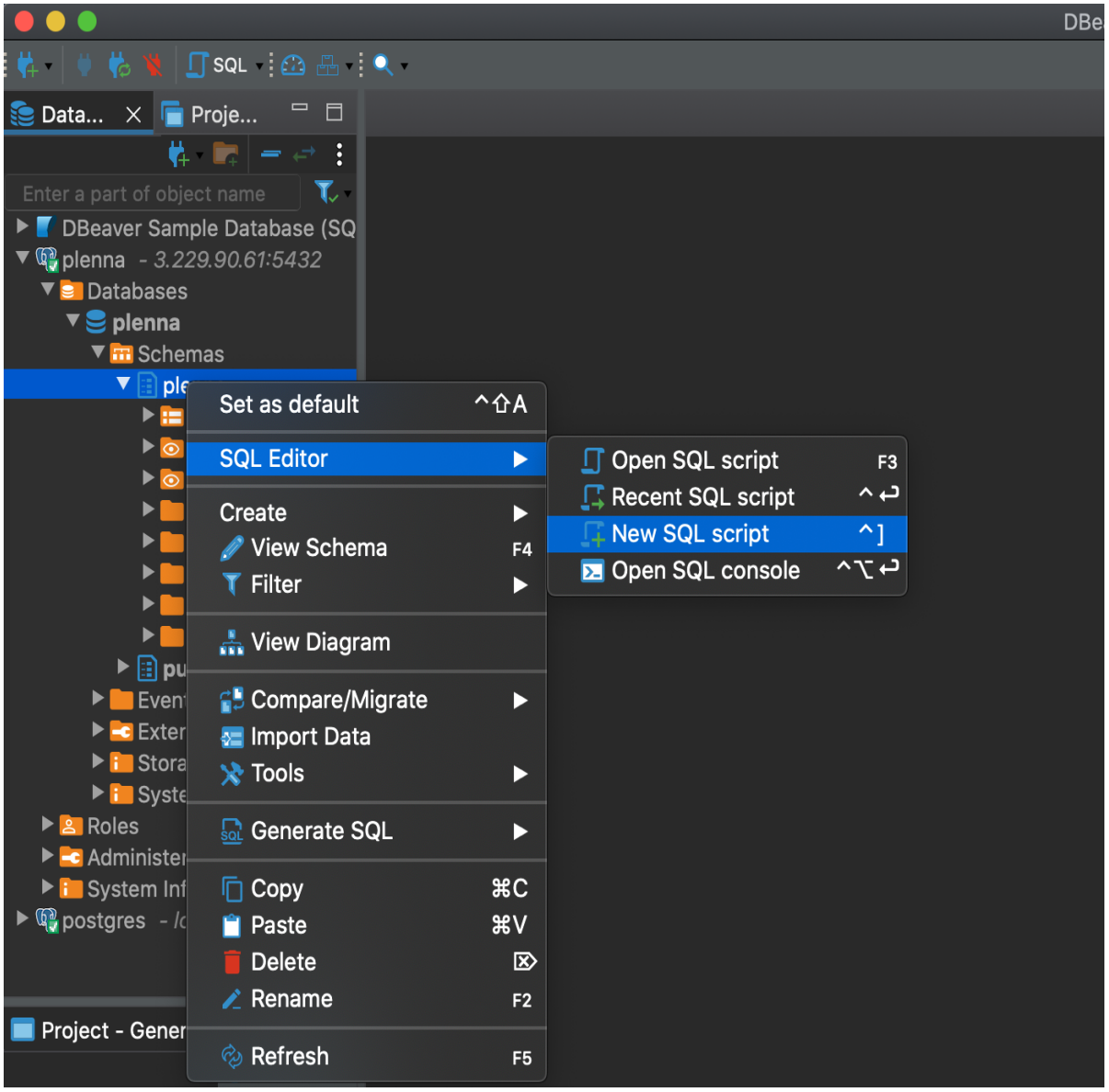
https://github.com/VicCont/Plenna

Instructivo para ejecutar proyecto de manera local

1. Asegurarse de tener python descargado
2. Jalar el repositorio de Git
3. Activar el ambiente virtual (en una línea de comandos de la computadora ejecutar *..\plenna_env\Scripts\activate.bat*)
4. Asegurarse que el servidor de la base de datos esté corriendo (consultar con nosotros - AWS)
5. Correr el servidor integrado de Django:
    a. Accediendo a *cd ..\GitHub\Plenna\back\plenna*
    b. Corriendo (desde dicha dirección) *python manage.py runserver*
6. Acceder desde el explorador a la página web: *127.0.0.1:8000*
7. Acceder con usuario: doctor 2 // contraseña: basura
8. Nota: para ver las gráficas en tiempo real se requiere de una cuenta en Tableau server. Se puede hacer una prueba por 14 días sin costo.
9. Nota 2: la contraseña de la base de datos es: *n0m3l0*

Instructivo para realizar pruebas

1. Establece conexión a la base de datos de plena.
2. Crea un nuevo *SQL script*: en este archivo se ejecutarán los queries de las pruebas.

a. 

3. Copiar las las pruebas que salen el archivo *PlennaPruebas:* https://github.com/emiliog01/Primera-Tarea-BD/blob/main/PlennaPruebas.sql
   Y pegarlas en el *SQL* script creado.

4. Para ejecutar cada query, sombrea el query completo con el cursor y presiona *control + enter.*

A continuación se encuentran algunas de las pruebas realizadas, y sus explicaciones, sobre la base de datos de Plenna. El archivo completo (que contiene todas las pruebas) se encuentra en github.

Pruebas básicas:

```
--Las preguntas opcionales por paciente
select * from opc_preg op join pregunta p using (id_preg)
join res_preg_opc rpo using (id_opc_preg);
```

Este query hace *match* entre los pacientes y sus correspondientes preguntas opcionales.

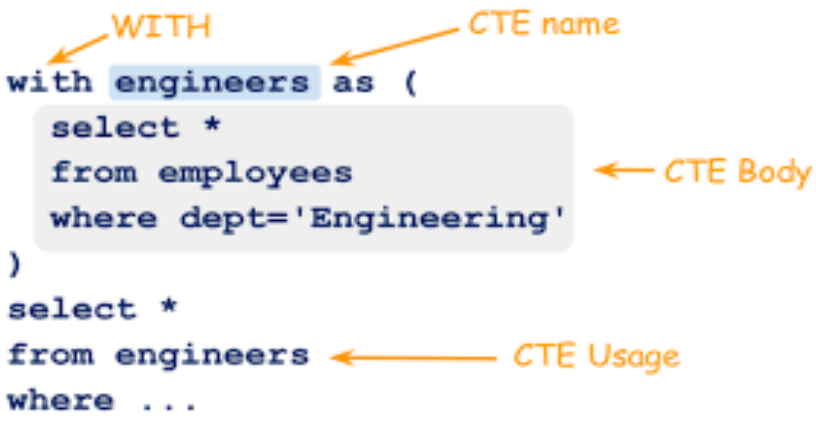
```
--Las preguntas abiertas por paciente
select p.nombre, p2.pregunta, ra.resp_preg
from paciente p join preg_pac pp using (id_pac)
join pregunta p2 using (id_preg)
join resp_abierta ra using (id_preg_pac)
group by p.nombre, p2.pregunta, ra.resp_preg
order by 1, 2;
```

Este query utiliza *Foreign Keys* para viajar a través de la base de datos y regresa las preguntas abiertas que se les hicieron a cada paciente.

Pruebas complejas:

Las pruebas en esta sección fueron elaboradas con *Common Table Expressions (CTE).* Este tipo de expresiones siguen una estructura que mejora la legibilidad y, en algunos casos, la eficiencia.

A continuación se da un ejemplo gráfico de la estructura de una CTE:

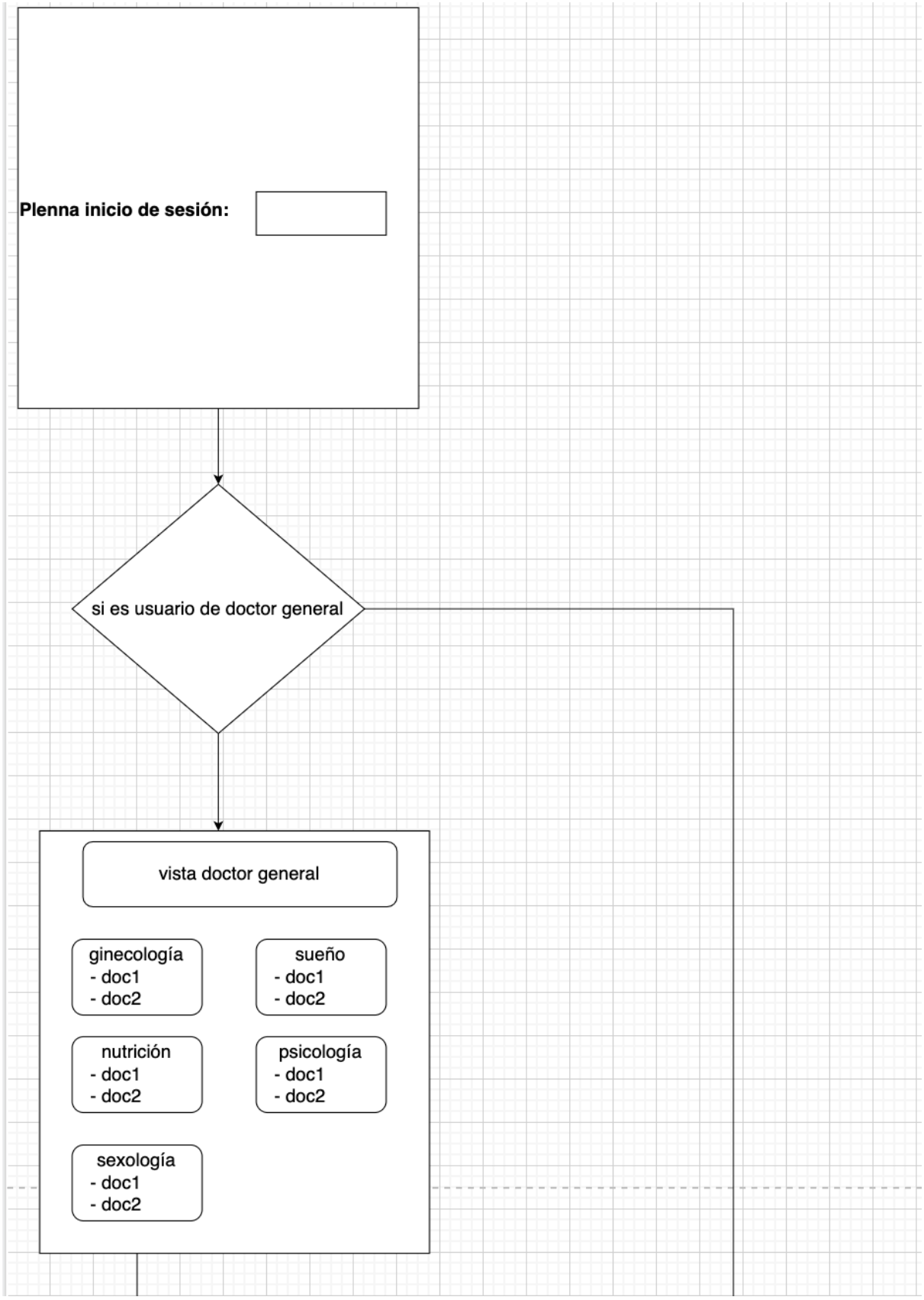


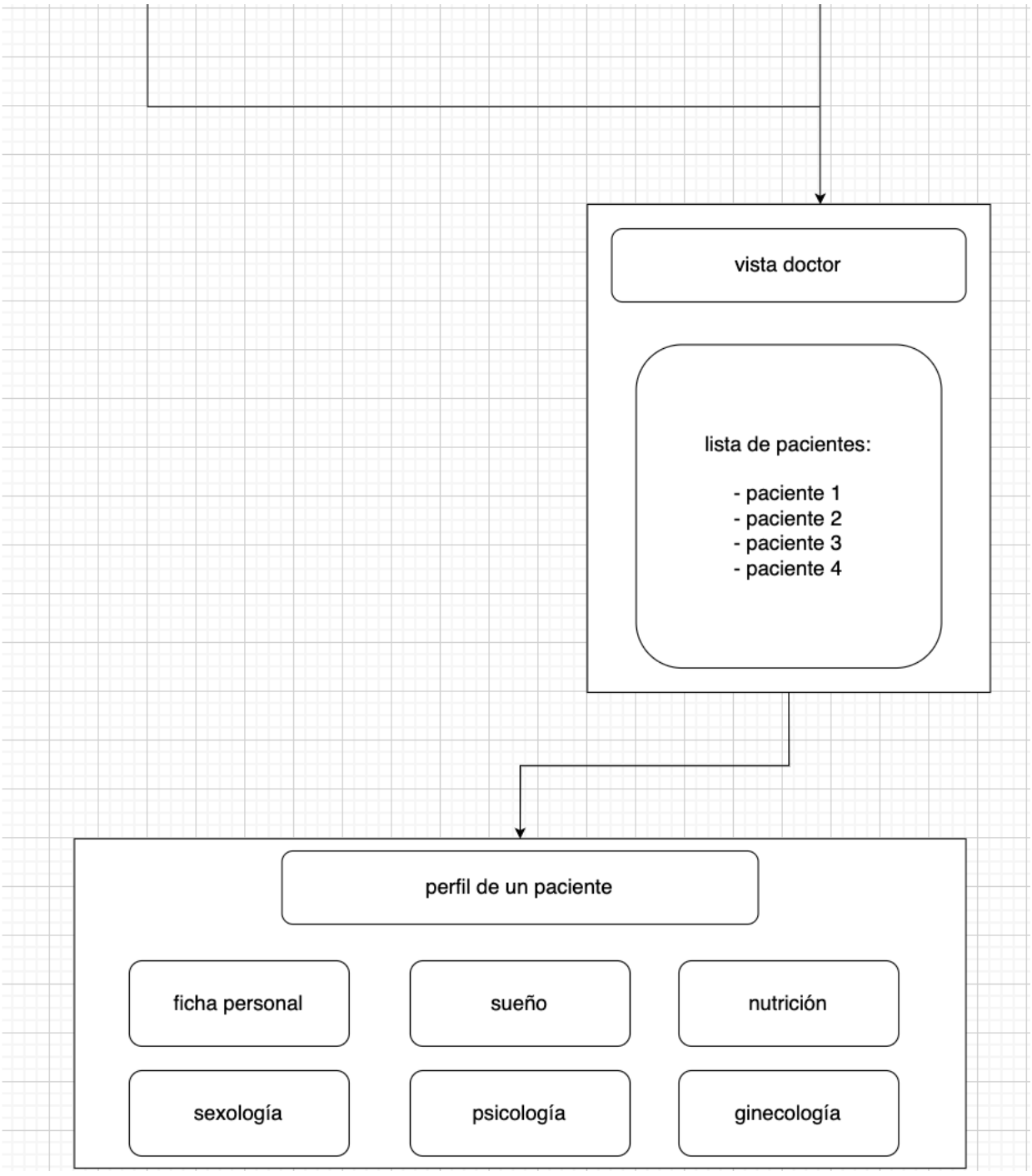
Veamos pruebas ejecutadas en la base de datos de Plenna que usan esta estructura:

```
-- Los permisos que tienen los doctores sobre los pacientes
with permiso_doc_pac as (
    select d.nom_doc, d.id_especialidad, p2.nombre
    from doctor d join permisos p using (id_doctor)
    join paciente p2 using (id_pac)
    group by d.nom_doc, d.id_especialidad, p2.nombre
)
select permiso_doc_pac.nom_doc, e.nombre_esp, permiso_doc_pac.nombre as nom_paciente, e.activa
from permiso_doc_pac join especialidad e using (id_especialidad)
```

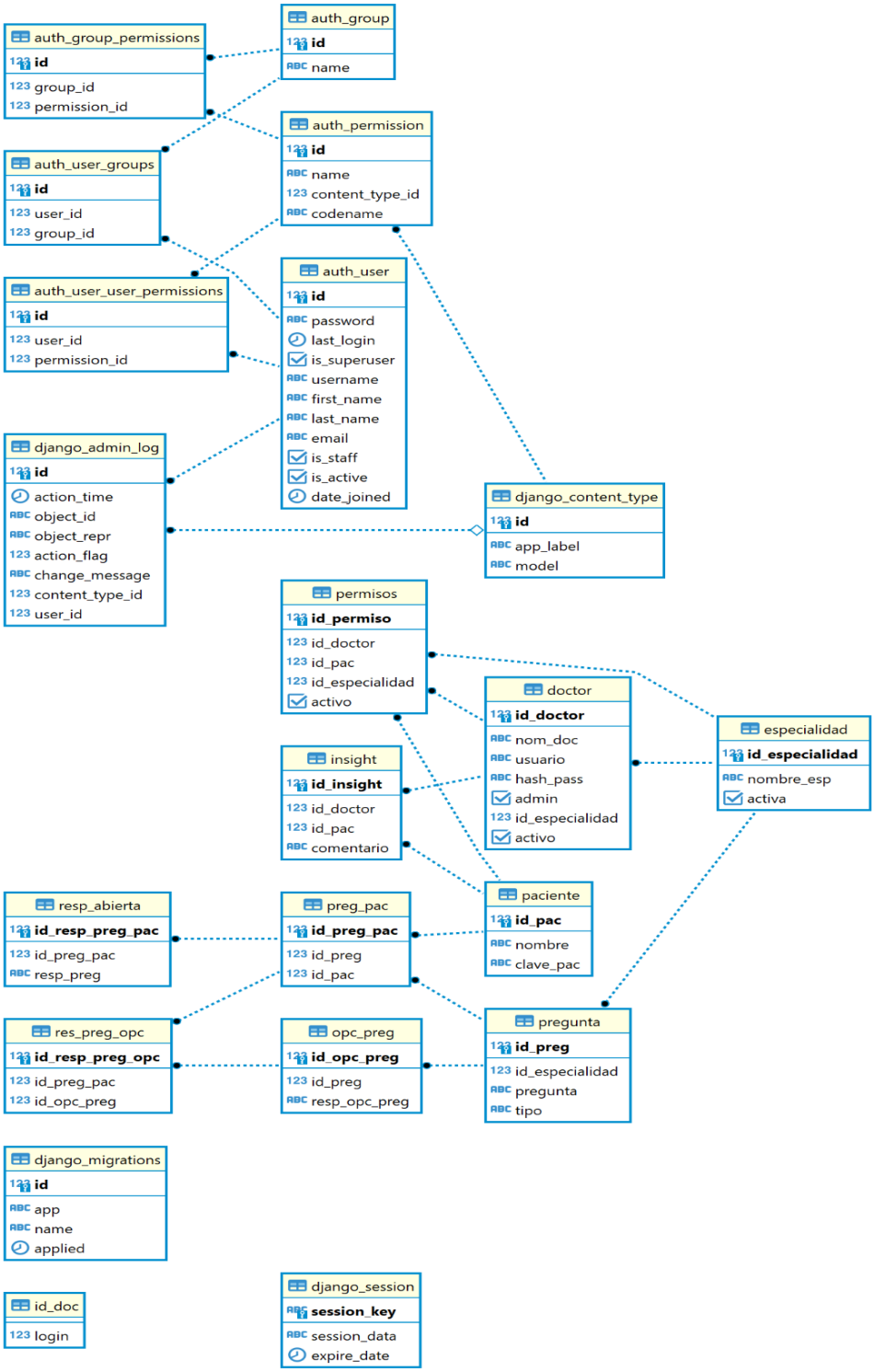
Este query regresa, en human readable, los permisos que tiene cada doctor para acceder información de los pacientes. Veamos que *permiso_doc_pac* es el nombre que se le da a la *CTE.* El cuerpo de la *CTE* regresa las columnas: *nom_doc, nombre_esp, nom_paciente, activa* (variable booleana).

Diagramas de flujo




vista doctor
lista de pacientes:
- paciente 1
- paciente 2
- paciente 3
- paciente 4
perfil de un paciente
ficha personal
sueño
nutrición
sexología
psicología
ginecología

Diagramas de entidad relación



Diagramas de secuencia

Acudir al repositorio de github para una visualización completa.

Referencias

Imagen (2022): “NUAA Wellbeing Group” extraído de: https://images.squarespace-cdn.com/content/v1/578cc1522e69cf573b568415/1529896367649-OKRXB6YUI7XCSKV7W60C/hands+with+love+hearts.png?format=1000w

Imagen (2017) “Non-recursive common table expression overview” extraído de: https://mariadb.com/kb/en/non-recursive-common-table-expressions-overview/