**Organización del código.**

Nuestra aplicación está basada en dos clases una que permite la creación de PDF utilizando una librería y otra Clase que es la aplicación en sí, a su vez se utiliza la librería SQLite3 para que los datos queden guardados en una base de datos local.

La clase de generación de PDF tiene métodos muy simples cómo agregar un título, subtítulo, texto y la generación del PDF como tal.

Por otro lado pero siguiendo el mismo esquema, la aplicación dispone de muchos métodos.teniendo como principal un main() que es el que va llevando el flujo del programa:

este pedazo de código muestra muy claro el flujo básico de nuestro programa

def main():

app = App() **// inicializamos la clase App**

[user\_role, dni] = login(app) **// usamos el método login para poder acceder a un portal**

if user\_role == 'medic':

medic\_portal(app) **// el login retorna un tipo de usuario y un dni asi podemos saber a que portal debe ir el usuario.**

elif user\_role == 'patient':

patient\_portal(app, dni)

else:

print('unknown user role')

el primer método que es muy importante es el de inicio de sesión. A partir de ahí la aplicación sigue un camino distintos para los médicos y otro para los pacientes.

Tenemos dos métodos principales de interacción los cuales llamamos portal y dependiendo de un caso o el otro tendrán métodos una gran cantidad de métodos distintos que permiten a los usuarios hacer una cosa u otra.

Para ser más específico los médicos tienen la opción para crear un usuario de tipo paciente o agregar información médica a un paciente cada uno con su métodos asociados para llevar a cabo esta tarea.

def medic\_portal(app: App):

print('\nMedic portal') **// le indicamos al usuario que efectivamente se encuentra en el portal médico.**

while True:

option = int(input('1. Create user\n2. Add vaccine to patient\n0. Exit\n\nEnter option: '))

**// aquí el médico deberá indicar que desea hacer si crear un usuario o agregar información médica relacionado con la vacuna covid**

if option == 0: **// llamamos a distintos métodos dependiendo de lo que desea hacer el médico.**

return

elif option == 1:

create\_user(app) **// metodo para crear pacientes.**

elif option == 2:

add\_vaccine\_to\_patient(app) **// información médica relacionado con la vacuna covid .**

else:

print('\nUnknown option\n')

Por otro lado los pacientes solamente tienen la opción de descargar un pdf con su certificado el cual hará una llamada a la clase de crear PDF conectándose con la base de datos en dónde un médico previamente ha introducido los datos.

def patient\_portal(app: App, dni: str):

print('\nPatient portal') **// le indicamos al usuario que efectivamente se encuentra en el portal de pacientes.**

while True:

option = int(input('1. Generate covid certificate\n0. Exit\n\nEnter option: ')) **// llamamos a distintos métodos dependiendo de lo que desea hacer elusuario.**

if option == 0:

return

elif option == 1:

app.get\_covid\_certificate(dni) **// metodo que hace una llamada a la clase de generar PDF que es la que se encarga de con los datos que hay en la base de datos generar el pdf.**

print('\nCovid certificate generated\n')

else:

print('\nUnknown option\n')