

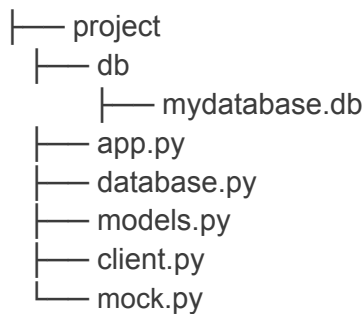
# Implementació CRUD amb FastAPI i SQLAlchemy

## Settings

### 1. Creeu un projecte PyCharm nou amb els següents requisits:

- Dependencies: fastapi["standard"], sqlalchemy, requests

- Estructura mínima:



- VCS: git/github

### 2. Definir una ruta d'inici:

- A la ruta / (root), retorneu un missatge de benvinguda i informació bàsica sobre la vostra API.
- Comproveu que tot funciona correctament
- el fitxer mock.py contindrà un script per crear i inicialitzar la base de dades amb alguns exemples. **Important:** aquest script ha de funcionar per tal que es pugui crear i inicialitzar la base de dades, a l'hora de fer l'avaluació.

### 3. Startup:

- definiu el startup de la base de dades amb @asynccontextmanager i lifespan (hauria d'estar definit a database.py)

## Data Model

Heu d'escollir vosaltres una entitat o més entitats del vostre interès, dissenyar i implementar el model de dades amb SQLAlchemy. Opcions:

1. Model de dades amb una única taula: **nota màxima 6**
2. Model de dades amb dues taules o més i relació one\_to\_many (1:N): **nota màxima 9**
3. Model de dades amb dues taules o més i relació many\_to\_many (N:M): **nota màxima 10**

## Clients

- Feu servir Thunder Client o el mateix docs de FastAPI per testejar els endpoints.
- A més, implementeu a client.py crides a tots els endpoints implementats. Gran part de l'avaluació es basarà en el correcte funcionament del client, és a dir, el **client ha de funcionar correctament**.

## Implementació

- Implementeu els endpoints bàsics per realitzar les operacions CRUD sobre el conjunt de dades.
- Implementeu **cinc endpoints més** amb queries interessants sobre la base de dades.

## Entrega

Pugeu el projecte pycharm en format .zip, sense el venv ni base de dades.  
Passeu l'enllaç github del vostre repositori.