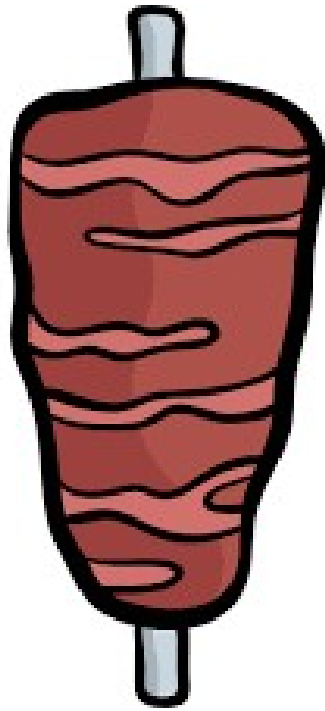


MYSKBAB



Marc Carrillo Ramírez
1er DAM: Base de Dades
Juan Talens Ferrer
IES Jaume II el Just



Sumario

Introducció.....	3
Ferramentes i mètodes.....	3
Software.....	3
Metodología.....	3
Perspectiva estàtica.....	4
Entintat / Relació.....	4
Pasos a taules.....	4
DDL (Data Definition Language).....	5
DML (Data Manipulation Language).....	6
DQL (Data Query Language).....	6
DCL (Data Control Language).....	7
Perspectiva dinàmica.....	7
Casos d'us.....	7
Sketch.....	9
Diagrama visual.....	9
Prototip.....	10
Conclusions.....	11
Bibliografia.....	12

Introducció

El projecte MySkebab naix de la necessitat d'ajudar al local de Kebab del poble "amigo", el qual soluciona i ens proporciona aliment, en moments desde ja meditats durant tot el día, a moments d'improvisació i d'extrema urgència.

Ens van demanar ajuda, perquè creuen que venen molta varietat de productes, però volen saber i controlar la quantitat exacta i variar els preus segons la llei d'oferta i demanda, i també aprofitar per a col·laborar a disminuir el desperdici de menjar.

Les necessitats del projecte són bàsiques, i tot va a girar, majoritàriament, en anotar de manera senzilla les vendes exactes dels productes i proporcionar l'informació de manera simple, concisa i comprensible.

L'objectiu principal del present projecte va a ser facilitar al local l'anotació i preservar de les dades necessàries per al posterior anàlisi i conclusió.

Es podria comentar que, un dels objectius secundaris és aplegar a comprendre com funciona un local que pot semblar tant simple però que, com tots els negocis, no comprèn fins que no està a peu de guerra.

Ferramentes i mètodes

Software

DIA: Ens permet realitzar diagrames d'Entitat Relació (E/R).

Visual Studio Code: utilitzarem el llenguatge de programació Python, el qual és molt senzill i ens obri moltes possibilitats, al igual que el Visual Studio Code s'adapta correctament a ell i el fa molt fàcil de programar.

MySQLite: ens permetrà crear la base de dades corresponent i accedirem mitjançant els codis que crearem en Python.

Draw.io: lloc web en el qual realitzarem els dibuixos de diagrames

Umbrello: podrem dissenyar de manera moderadament còmoda els casos d'ús que ens permetran dur a terme l'aplicació i el projecte.

Metodología

El programa ha sigut creat mitjançant el software Visual Studio Code, concretament amb el llenguatge de programació Python.

Com es pot visualitzar en el propi codi, s'aprofita el mateix programa per a crear la base de dades (a la qual podem accedir i també visualitzem amb el software SQLite). També, és mitjançant el propi codi que es configura l'accés a la base de dades i s'actualitza a la vegada que l'aplicació

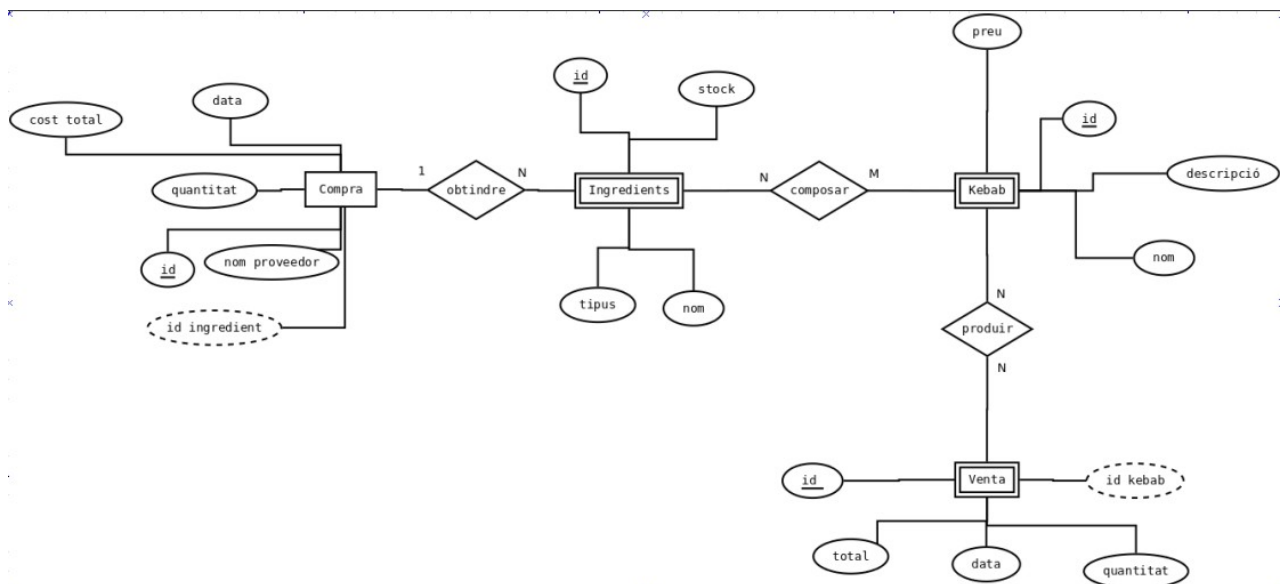
va funcionant, i es va retroalimentant la pròpia actualització de les dades amb la possibilitat d'utilitzar l'aplicació.

Tot açò s'ha dut a terme després de realitzar el disseny de com es volia implementar de manera gràfica (tal i com podem veure més avant) l'aplicació, cosa que ha servit com a guia de quins camins seguir per a desenvolupar l'aplicació.

Una vegada fet tot el procediment, s'ha procedit a provar l'aplicació per veure si es complien els objectius pels quals es va desenvolupar l'aplicació, i a explicar i detallar com ha sigut realitzat el treball en els seus conjunts.

Perspectiva estàtica

Entitat / Relació



Pasos a taules

1. Compra
 - Camps: *id*, *quan*, *data*, *nom_proveedor*, *cost_total*
 - Relacions: cada compra es relaciona amb varios ingredients mitjançant una taule intermija
2. Ingredients
 - Camps: *id*, *quan*, *data*, *nom_proveedor*, *cost_total*
 - Relacions: cada ingredient pot estar en varies compres i pot pertànyer a varios kebabs
3. Kebab
 - Camps: *id*, *preu*, *descripció*, *nom*

- Relacions: cada kebab es compon de varios ingredients i pot estar venut múltiples vegades.

4. Venda

- Camps: id, id_kebab, quantitat, data, total
- Relacions: cada venda està associada a un kebab específica.

DDL (Data Definition Language)

```
CREATE TABLE Compra (
    id INT PRIMARY KEY,
    quantitat INT,
    data DATE,
    nom_proveedor VARCHAR(255),
    cost_total DECIMAL(10, 2)
);
```

```
CREATE TABLE Ingredients (
    id INT PRIMARY KEY,
    stock INT,
    tipus VARCHAR(50),
    nom VARCHAR(255)
);
```

```
CREATE TABLE Kebab (
    id INT PRIMARY KEY,
    preu DECIMAL(10, 2),
    descripció TEXT,
    nom VARCHAR(255)
);
```

```
CREATE TABLE Venta (
    id INT PRIMARY KEY,
    id_kebab INT,
    quantitat INT,
    data DATE,
    total DECIMAL(10, 2),
```

```

    FOREIGN KEY (id_kebab) REFERENCES Kebab(id)
);

CREATE TABLE Compra_Ingredient (
    compra_id INT,
    ingredient_id INT,
    FOREIGN KEY (compra_id) REFERENCES Compra(id),
    FOREIGN KEY (ingredient_id) REFERENCES Ingredients(id),
    PRIMARY KEY (compra_id, ingredient_id)
);

CREATE TABLE Kebab_Ingredient (
    kebab_id INT,
    ingredient_id INT,
    FOREIGN KEY (kebab_id) REFERENCES Kebab(id),
    FOREIGN KEY (ingredient_id) REFERENCES Ingredients(id),
    PRIMARY KEY (kebab_id, ingredient_id)
);

```

DML (Data Manipulation Language)

```

-- Añadir un nuevo ingrediente
INSERT INTO Ingredients (id, stock, tipus, nom) VALUES (1, 50, 'Carne', 'Pollo');

-- Actualizar el stock de un ingrediente
UPDATE Ingredients SET stock = 30 WHERE id = 1;

-- Registrar una nueva compra
INSERT INTO Compra (id, quantitat, data, nom_proveedor, cost_total) VALUES (101, 10,
'2024-05-01', 'Proveedor X', 150.00);

-- Asociar compra con ingredientes
INSERT INTO Compra_Ingredient (compra_id, ingredient_id) VALUES (101, 1);

```

DQL (Data Query Language)

```

-- Consultar todas las ventas de un kebab específico
SELECT * FROM Venta WHERE id_kebab = 1;

-- Listar los ingredientes de un kebab particular

```

```
SELECT i.nom, i.tipus FROM Ingredients i
JOIN Kebab_Ingredient ki ON i.id = ki.ingredient_id
WHERE ki.kebab_id = 1;
```

DCL (Data Control Language)

```
-- Crear un nuevo usuario
CREATE USER 'kebab_admin' IDENTIFIED BY 'password123';

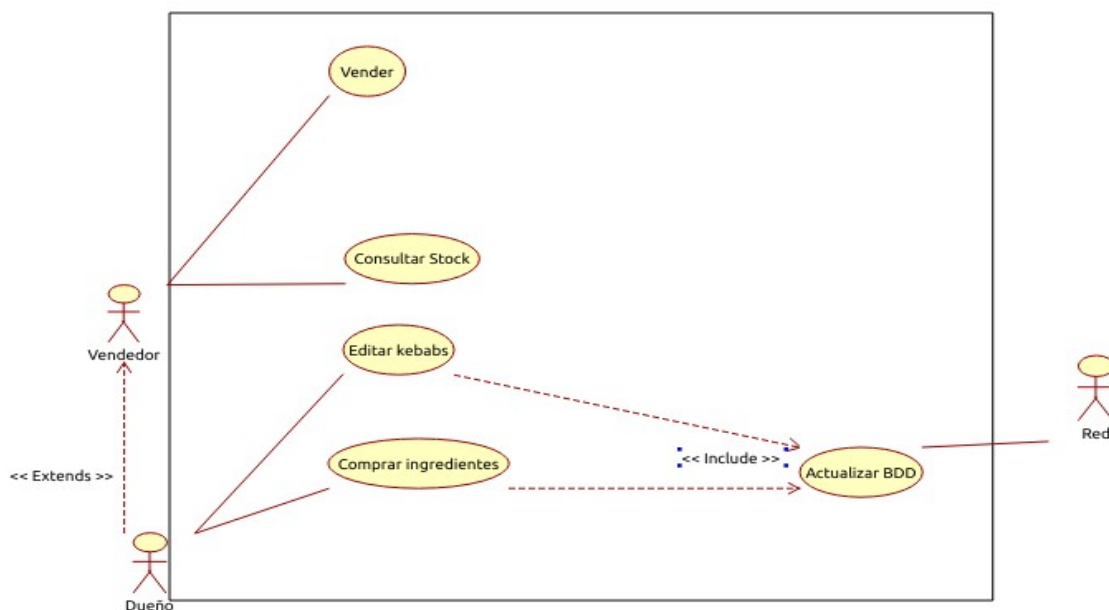
-- Dar permisos de consulta y modificación en la tabla Venta
GRANT SELECT, UPDATE ON Venta TO 'kebab_admin';

-- Revocar permisos de actualización
REVOKE UPDATE ON Venta FROM 'kebab_admin';
```

Perspectiva dinámica

Casos d'us

A continuació, anem a inserir un diagrama de casos d'us, en el que ixià una visualització simple dels casos d'us que poden tindre els dos usuaris a l'hora d'obrir l'aplicació creada, i procedirem a explicar breument el per que del diagrama.



En l'esquema, podem veure tres actors: l'usuari que entra a l'app com a venedor, el que entra com al propietari del negoci, i la propia red que va a utilitzar l'amo del negoci.

S'ha ficat el actor «Dueño» com a extens de «vendedor» perquè el amo del negoci sí que podra realitzar les comandes i editar els kebabs a vendre, però la resta de treballadors solament podran consultar l'stock dels ingredients per a saber si poden vendre els kebabs que se'ls demana.

Actor	Vendedor	Identificador: 6542
Descripción	Vendedor simple de productos	
Características	Usuari simple	
Relación	La base de la venta de l'amo del negoci	
Referencias	El actor intervé en la venta dels kebabs i, de manera indirecta influeix en la necessitat de que l'amo tinga que actualitzar les compres dels ingredients necessaris per a seguir realitzen vendes.	

Atributos	
Nombre	Descripción
Vendedor	El usuari «vendedor» amb contrasenya «dekebabs» tindrà capacitat per a afegir les vendes del local y consultar l'stock, cosa que serà de molta utilitat per al propietari

Actor	Propietari	Identificador: 6544
Descripción	Propietari del negoci que ens ha demant desenvolupar l'app	
Características	Usuari amb més possibilitats de les que té el propi vendedor. Té ampliació de funcions que a la vegada és necessaria per a que funcionen les funcions més bàsiques	
Relación	Dona la possibilitat per a que l'usuari «vendedor» siga d'utilitat.	
Referencias	Gracies al treball de la venta de Kebabs que s'han realitzat anteriorment per part dels vendedors i seua, tindrà necessitat d'adquirir nous ingedients i seguir composant els kebabs per a la seua venda. En quant a l'adquisició de productes, el propietari va necessitar de connexió a internet per a adquirir els productes, al igual que per a editar els kebabs.	

Atributos	
Nombre	Descripción
Propietari	El usuari «amigo» amb contrasenya «kebab» va a tindre la capacitat de dirigir les claus per a que funcione el negoci, editant tant els productes que van a poder vendre la resta de treballadors, com l'adquisició d'aquests. No obstant, existeix el requisit d'estar conectat a internet per a poder realitzar les comandes i les actualitzacions en les bases de dades.

Actor	Internet	Identificador: 8080
Descripción	Conexió base a internet	
Características	Capacita al propietari a realitzar les funcions extra que li permeten dur a terme les funcions necessàries per a que funcione correctament el local.	
Relación	Es relaciona unicament en els dos casos d'us afegits al propietari	
Referencias	Solament els inicis de sessió del propietari amb intenció de fer funcions d'administrador.	

Atributos	
Nombre	Descripción
Red	Es una connexió a internet simple, on no es necessita res més que connexió a l'hora d'armacenar tant les dades d'actualització dels kebabs com les característiques explícites de la comanda d'ngredients.

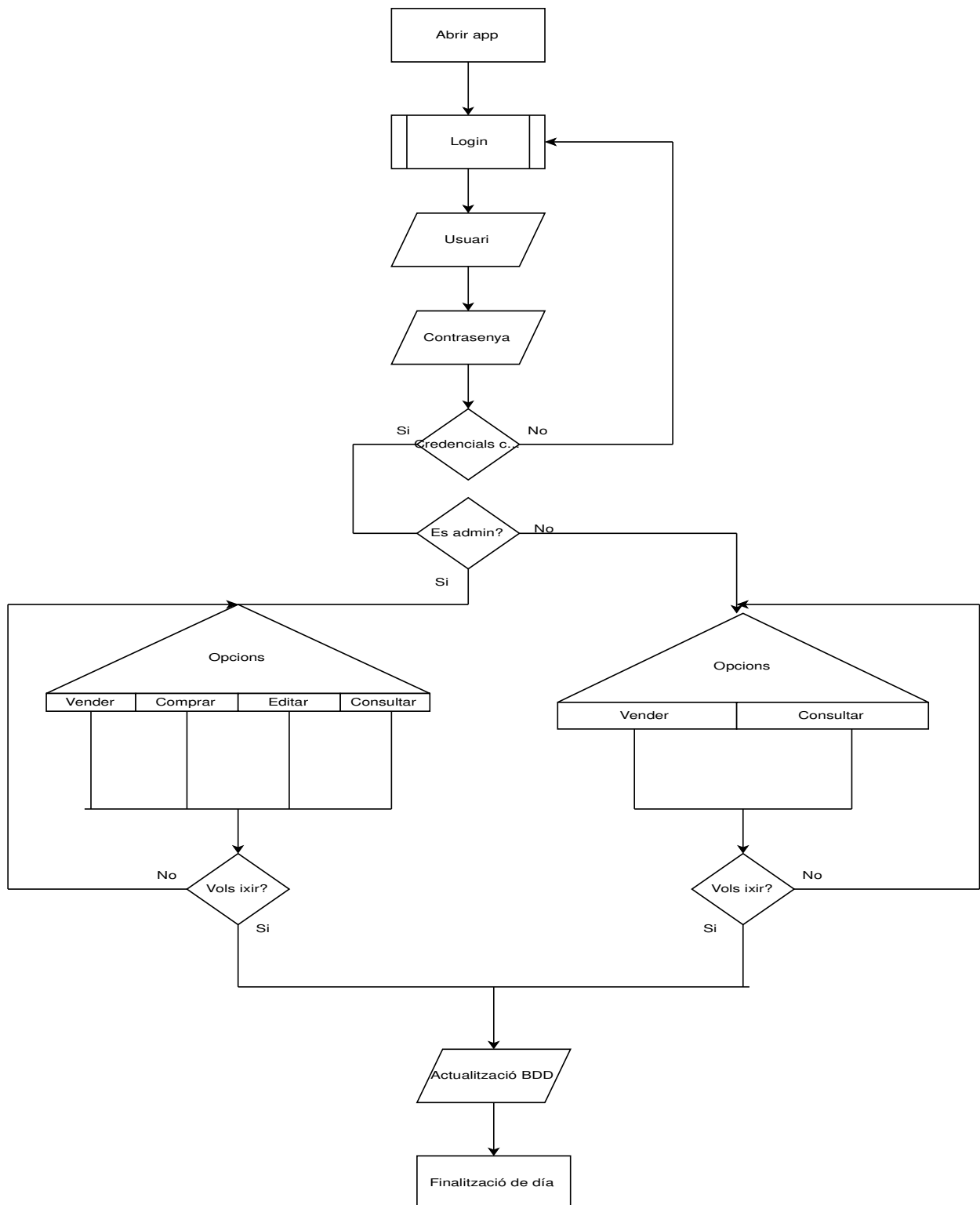
Sketch

Diagrama visual

Tal i com podem veure en el dibuix, el diagrama base és prou simple: depenent de si entres a l'aplicació com a administrador o simplement com a venedor, podràs realitzar dos funcions extra o no.

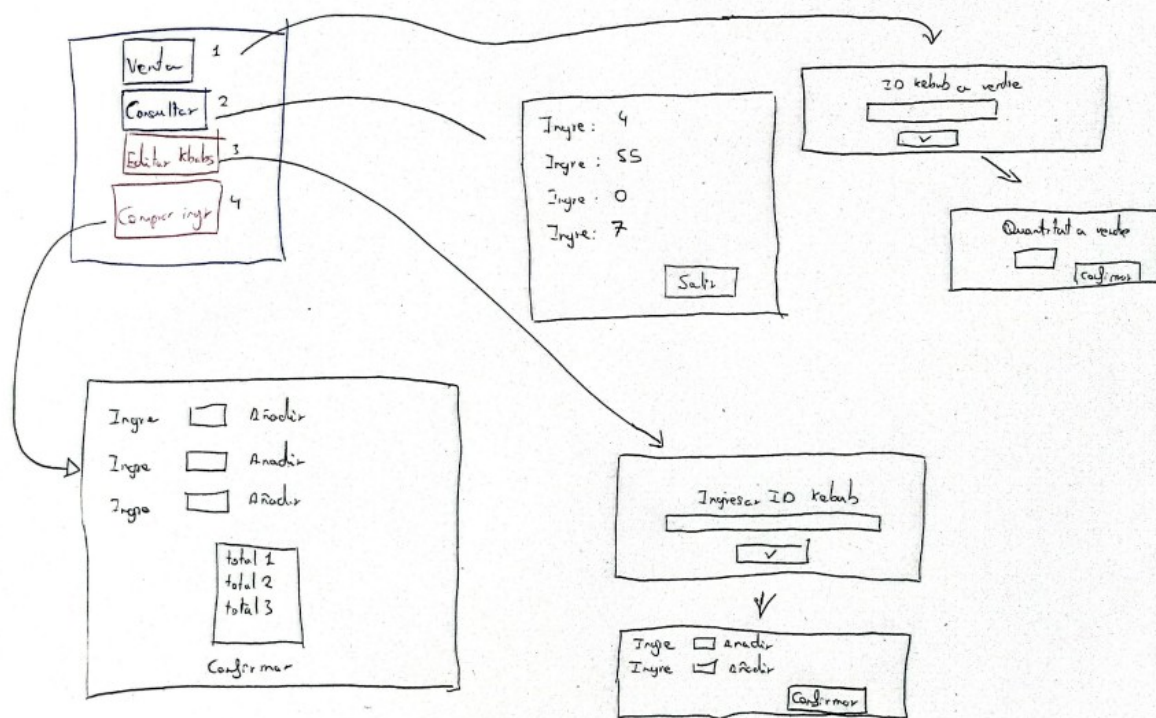
Además, podem visualitzar que entra tot dins d'un bucle (depenent del usuario, però cadascun amb el seu bucle) en el qual sempre tenim les mateixes opcions.

Tambe es pot destacar que, en el cas de login, no deixa cap opció d'entrar sense tindre usuari.



Prototip

En aquest cas, anem a explicar quina era la visualització que es tenia pensada per a l'aplicació i les seues funcions abans de picar codi.



Obviament, per a realitzar el prototip, s'ha fet el cas de l'usuari administrador perquè podem visualitzar totes les funcionalitats. La visualització és molt simple, ja que podem veure que en total s'obrien 7 finestres sense contar la d'entrada de dades per a accedir.

És molt simple, ja que podem veure que solament en dos casos, s'obrirà una finestra extra una vegada es selecciona l'opció que volem del menú.

Conclusions

Les conclusions a les que podem aplegar, al meu paréixer, son dos.

En primer lloc, podem concloure que no és necessari complicar-se molt per a realitzar una aplicació, i més en casos en els que la funcionalitat no és molt complexa: amb un accés a dades, introducció i edició senzilla d'aquestes, es compleixen les necessitats que creguem necessàries que tinga l'aplicació. A més, no cal fixar-se extremadament en el disseny sempre i que complisca les funcionalitats base del back-end i tinga distinció suficient de les funcions per part del front-end.

En segon lloc, des d'un punt de vista més crític, realista, i amb perspectiva de futur, s'ha de deixar clar que el present projecte té l'intenció més simplista i bàsica possible. En aquest cas, l'objectiu del projecte era connectar una base de dades amb un programa, i ser capaços d'explicar com funcionen els pilars de l'aplicació, però no podem deixar de costat l'evidència de que, en un cas real, no es pot realitzar un projecte tant simplista i el qual no s'ha intentat ampliar més del que ens demanaria un client que vol desenvolupar una aplicació.

Està clar que els objectius del projecte eren bàsics i simples, però per algo es comença i sempre és benvinguda una primera experiència per molt insignificant que puga semblar. A partir d'ara, l'únic objectiu ha de ser tindre en conter lo realitzat en el present treball, tant per a agafar els aspectes positius com per a millorar els aspectes que tenen cert marge per a realitzar-ho.

Bibliografía

Chat GPT. (s. f.). Open AI. <https://chat.openai.com>

Enllaç a GitHub:

<https://github.com/CarrilloMarc/MySKbab/tree/main>