



3-5-2024

# Only Data



Carles Bernabeu Molió

## Índex

Índex.....	1
1.Introducció .....	2
2.Ferramentes i mètodes .....	3
3.Perspectiva estàtica .....	4
3.1.E/R (Entitat-relació) .....	4
3.2. Pas a taules .....	5
3.3.DDL (Data Definition Language) .....	5
3.4.DML (Data Manipulation Language) .....	6
3.5. DQL (Data Query Language) .....	7
4. Perspectiva dinàmica.....	7
4.1.Sketch .....	7
.....	8
4.2. Casos de Uso (Mètodes) .....	8
5.Conclusions 5.1.Resultats obtinguts.....	10
5.2. Reflexions sobre el procés i possibles millores futures .....	11
6. Bibliografía y Webgrafía .....	11

# 1.Introducció

L'aplicació va relacionada amb una aplicació actual que en l'any 2023 es calcula va facturar 1000 milions d'euros. Aquesta aplicació s'anomena OnlyFans i és coneguda com "la xarxa social del sexe". En l'aplicació, models pugen contingut sexual, i els clients paguen per consumir-lo.

L'intenció d'aquesta aplicació és crear una base de dades on es recullen dades de models, els managers i les agències de managers. Per tal que models poden buscar manager, o manager models, sempre que no estiguin treballant amb ningú, per treballar junts. A la vegada que es guarden les agències per saber si els managers treballen en alguna agència o treballen per ells mateixa. També es guardaran els ingressos que té cada model i els ingressos que cada manager pot aconseguir que guanyen els models.

L'aplicació facilitaria la comunicació i la recerca de beneficis tan per part de models, com de managers i agències. On podrien contrastar els beneficis actuals amb els que podrien aconseguir. Disposa de les opcions de mostrar el managers que no tenen agència, per poder ser contractats per aquestes, i de mostrar models que no disposen de manager, per també poder ser contractades.

## 2.Ferramentes i mètodes

Les ferramentes usades per a crear l'aplicació son les següents:

- Draw.io: Ferramenta utilitzada en la que elaborarem el diagrama d'entitat relació de la nostra base de dades.

- Visual Studio Code: Aquesta ferramenta serà l'entorn de desenvolupament de l'aplicació. Serà on està programat el codi fet amb python.

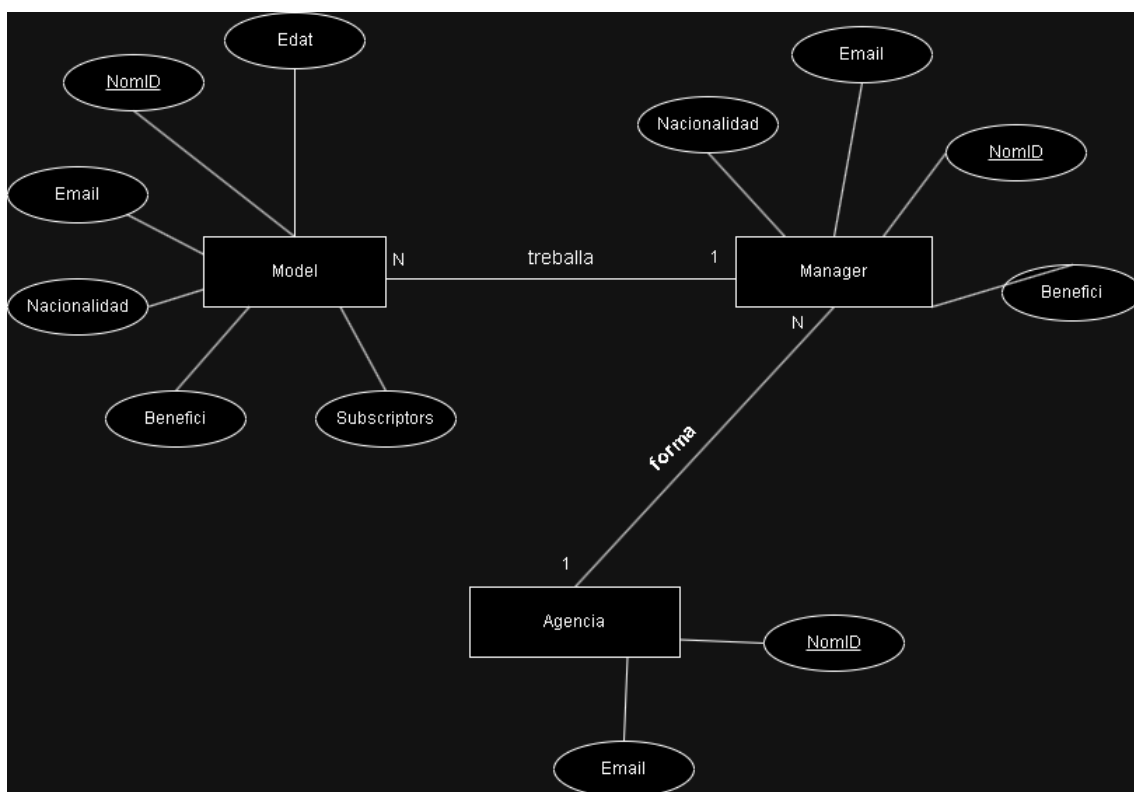
- Tkinter: Aquesta llibreria la importarem al codi de python, que és la que permetrà crear l'interfície gràfica amb la qual interactuarem en l'aplicació.

- DB Browser (SQLite): Serà el entorn de desenvolupament de la base de dades, la qual connectarem amb la nostra aplicació. S'ha usat aquesta aplicació per la facilitat de connexió entre el codi i la base de dades.

### 3.Perspectiva estàtica

#### 3.1.E/R (Entitat-relació)

Ací es veu el diagrama d'entitat relació



## 3.2. Pas a taules

El diagrama entitat-relació passa a taules com es veu a continuació.

S'han creat 3 taules, una per a cada entitat i amb els seus corresponents atributs.

AGENCIA = NomID + Email

MANAGER = NomID + AgencialID + Email + Nacionalitat + Benefici

C.Ali: AgencialID -> AGENCIA(NomID)

MODEL = NomID + ManagerID + Edat + Email + Nacionalitat + Benefici  
+ Subscriptors

C.ali: ManagerID -> MANAGER(NomID)

## 3.3.DDL (Data Definition Language)

En aquesta part es mostra com es defineix l'estructura de la base de dades i es crea en el codi de python.

```
5 # Connexió a la base de dades
6 conn = sqlite3.connect('database.db')
7 c = conn.cursor()
8
9 # Creació de las taules amb les modificacions sollicitades
10 c.execute('''CREATE TABLE IF NOT EXISTS AGENCIA
11             (AgenciaID INTEGER PRIMARY KEY, nom TEXT, Email TEXT)''')
12 c.execute('''CREATE TABLE IF NOT EXISTS MANAGER
13             (ManagerID INTEGER PRIMARY KEY, AgenciaID INTEGER, Nom TEXT, Email TEXT,
14              Nacionalitat TEXT, Benefici TEXT,
15              FOREIGN KEY(AgenciaID) REFERENCES AGENCIA(AgenciaID))''')
16 c.execute('''CREATE TABLE IF NOT EXISTS MODEL
17             (ModelID INTEGER PRIMARY KEY, ManagerID INTEGER, Edat INTEGER, Nom TEXT,
18              Email TEXT, Nacionalitat TEXT, Benefici TEXT, Subscriptors INTEGER,
19              FOREIGN KEY(ManagerID) REFERENCES MANAGER(ManagerID))''')
20 conn.commit()
21
```

### 3.4.DML (Data Manipulation Language)

Ací es mostra la part del codi on manipulem dades, es veu les insercions que fem a les taules.

```
# Funció per a inserir dades en la taula AGENCIA
def insert_agencia():
    # Obtener los datos de la ventana
    nom = agencia_nom.get()
    email = agencia_email.get()

    if nom and email:
        data = (nom, email)
        c.execute('INSERT INTO AGENCIA (nom, Email) VALUES (?, ?)', data)
        conn.commit()
        messagebox.showinfo('Éxito', 'Datos insertados en AGENCIA')
    else:
        messagebox.showerror('Error', 'Por favor ingresa el nombre y el email de la agencia')

# Funció per a inserir dades en la taula MANAGER
def insert_manager():
    # Obtener los datos de la ventana
    agencia_id = manager_agencia_id.get()
    nom = manager_nom.get()
    email = manager_email.get()
    nacionalitat = manager_nacionalitat.get()
    benefici = manager_benefici.get()

    if agencia_id and nom and email and nacionalitat and benefici:
        data = (agencia_id, nom, email, nacionalitat, benefici)
        c.execute('INSERT INTO MANAGER (AgenciaID, Nom, Email, Nacionalitat, Benefici) VALUES (?, ?, ?, ?, ?)', data)
        conn.commit()
        messagebox.showinfo('Éxito', 'Datos insertados en MANAGER')
    else:
        messagebox.showerror('Error', 'Por favor ingresa todos los datos del manager')
```

```
53 # Funció per a inserir dades en la taula MODEL
54 def insert_model():
55     # Obtener los datos de la ventana
56     manager_id = model_manager_id.get()
57     edat = model_edat.get()
58     nom = model_nom.get()
59     email = model_email.get()
60     nacionalitat = model_nacionalitat.get()
61     benefici = model_benefici.get()
62     subscriptors = model_subscriptors.get()
63
64     if edat and nom and email and nacionalitat and benefici and subscriptors:
65         if manager_id:
66             data = (manager_id, edat, nom, email, nacionalitat, benefici, subscriptors)
67             c.execute('INSERT INTO MODEL (ManagerID, Edat, Nom, Email, Nacionalitat, Benefici, Subscriptors) VALUES (?, ?, ?, ?, ?, ?, ?)')
68         else:
69             data = (None, edat, nom, email, nacionalitat, benefici, subscriptors)
70             c.execute('INSERT INTO MODEL (ManagerID, Edat, Nom, Email, Nacionalitat, Benefici, Subscriptors) VALUES (?, ?, ?, ?, ?, ?, ?)')
71             conn.commit()
72             messagebox.showinfo('Éxito', 'Datos insertados en MODEL')
73     else:
74         messagebox.showerror('Error', 'Por favor ingresa todos los datos del modelo')
75
```

### 3.5. DQL (Data Query Language)

Aquest punt es mostren les consultes a la base de dades, en les que mostrem les consultes de les models que no tenen manager per poder buscar per contractar. I també els managers sense agència.

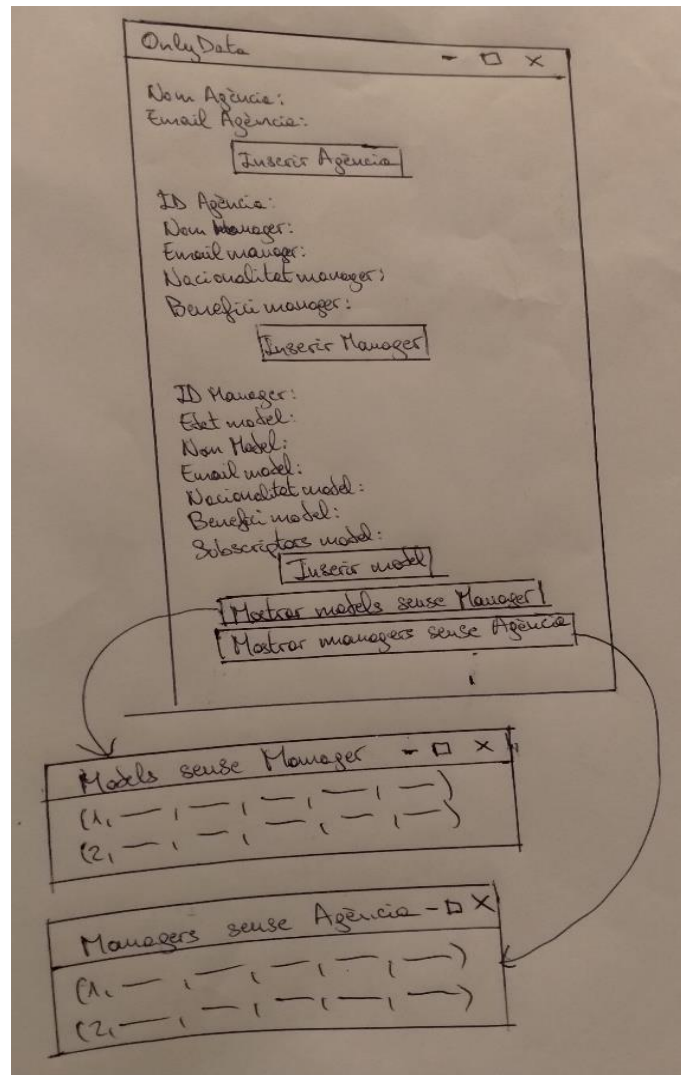
```
76 # Funció per a mostrar els models que no tenen manager assignat
77 def show_models_without_manager():
78     c.execute('SELECT * FROM MODEL WHERE ManagerID IS NULL')
79     data = c.fetchall()
80
81     show_window = tk.Toplevel()
82     show_window.title("Modelos sin Manager")
83
84     text_widget = tk.Text(show_window)
85     text_widget.pack()
86
87     for row in data:
88         text_widget.insert(tk.END, f"{row}\n")
89
90 # Funció per a mostrar els managers que no tenen agencia assignat
91 def show_managers_without_agency():
92     c.execute('SELECT * FROM MANAGER WHERE AgenciaID IS NULL')
93     data = c.fetchall()
94
95     show_window = tk.Toplevel()
96     show_window.title("Managers sin Agencia")
97
98     text_widget = tk.Text(show_window)
99     text_widget.pack()
100
101     for row in data:
102         text_widget.insert(tk.END, f"{row}\n")
103
```

## 4. Perspectiva dinàmica

### 4.1.Sketch

Esbos del que serà l'aspecte de l'aplicació on s'interactuarà amb ella. El disseny que es desitja obtenir, amb els camps per introduir de cada entitat i els botons respectius per inserir. Com també els botons per mostrar després el que interessa, que són els managers sense agència i els models sense manager que es volen contractar.





## 4.2. Casos de Uso (Mètodes)

Caso de Uso	Clicar inserir Agència	Identificador: Inserir Agència
Actores	Usuari	
Tipo	Primari	
Precondición	Inserir els camps obligatòris (tots).	
Postcondición	Inserir les dades en la base de dades i mostrar una finestra confirmant que s'han inserit.	

Descripción	S'introdueixen les dades necessàries, es clica el botó i s'insereixen les dades en la taula corresponent.
-------------	---

Caso de Uso	Clicar inserir Manager	<b>Identificador:</b> Inserir Manager
Actores	Usuari	
Tipo	Primari	
Precondición	Inserir els camps obligatoris (tots menys IDAgència).	
Postcondición	Inserir les dades en la base de dades i mostrar una finestra confirmant que s'han inserit.	
Descripción	S'introdueixen les dades necessàries, es clica el botó i s'insereixen les dades en la taula corresponent.	

Caso de Uso	Clicar inserir Model	<b>Identificador:</b> Inserir Model
Actores	Usuari	
Tipo	Primari	
Precondición	Inserir els camps obligatoris (tots menys IDManager).	
Postcondición	Inserir les dades en la base de dades i mostrar una finestra confirmant que s'han inserit.	
Descripción	S'introdueixen les dades necessàries, es clica el botó i s'insereixen les dades en la taula corresponent.	

Caso de Uso	Clicar botó Mostrar models sense Manager	<b>Identificador:</b> Mostrar models
Actores	Usuari	

Tipo	Primari
Precondición	Haver introduït algún model sense manager.
Postcondición	Mostrar les dades de la selecció.
Descripción	Clicar el botó de mostra models sense manager, el cual mostrarà aquelles dades dels models que tenen el camp IDManager buit.

Caso de Uso	Clicar botó Mostrar managers sense Agència.	<b>Identificador:</b> Mostrar managers
Actores	Usuari	
Tipo	Primari	
Precondición	Haver introduït algún manager sense agència.	
Postcondición	Mostrar les dades de la selecció.	
Descripción	Clicar el botó de mostra manager sense agència, el cual mostrarà aquelles dades dels managers que tenen el camp IDAgència buit.	

## 5.Conclusions

### 5.1.Resultats obtinguts

Referent als resultats obtinguts podem concluir que tenim la capacitat per poder realitzar un treball com aquest, amb un codi en python que cree una base de dades.

El treball l'hem realitzat de forma simple i senzilla. Amb 3 taules amb els atributs corresponents. El codi ha sigut simple per crear la base de dades i poder consultar -los. El resultat ha sigut sobretot d'aprenentatge amb el tema de documentació d'un projecte.

## 5.2. Reflexions sobre el procés i possibles millores futures

Durant els procés m'he donat conter que podria seguir aplicant funcionalitats al codi per tractar la base de dades. Com per exemple afegir mes botons de mostrar dades, per tindre-ho tot més clar i més visible per buscar qualsevol cosa. També afegir opcions per eliminar o modificar dades, etc...

També en quant al front-end s'hagués pogut millor molta cosa, fins afegir imatges o logotip per fer l'aplicació més atractiva.

Una bona planificació també és molt important, una possible millora en el futur seria una bona organització del pas a pas a realitzar.

## 6. Bibliografía y Webgrafía

-Chat gpt.

-Llibreria tkinter (<https://docs.python.org/es/3/library/tkinter.html>)

-Enllaç Github: <https://github.com/BernabeuCarles/MicroProjecte.git>