

Base de Dades

Avançades: Pràctica 1

Alumnes:

Carlos Cortés
Pol Girbal

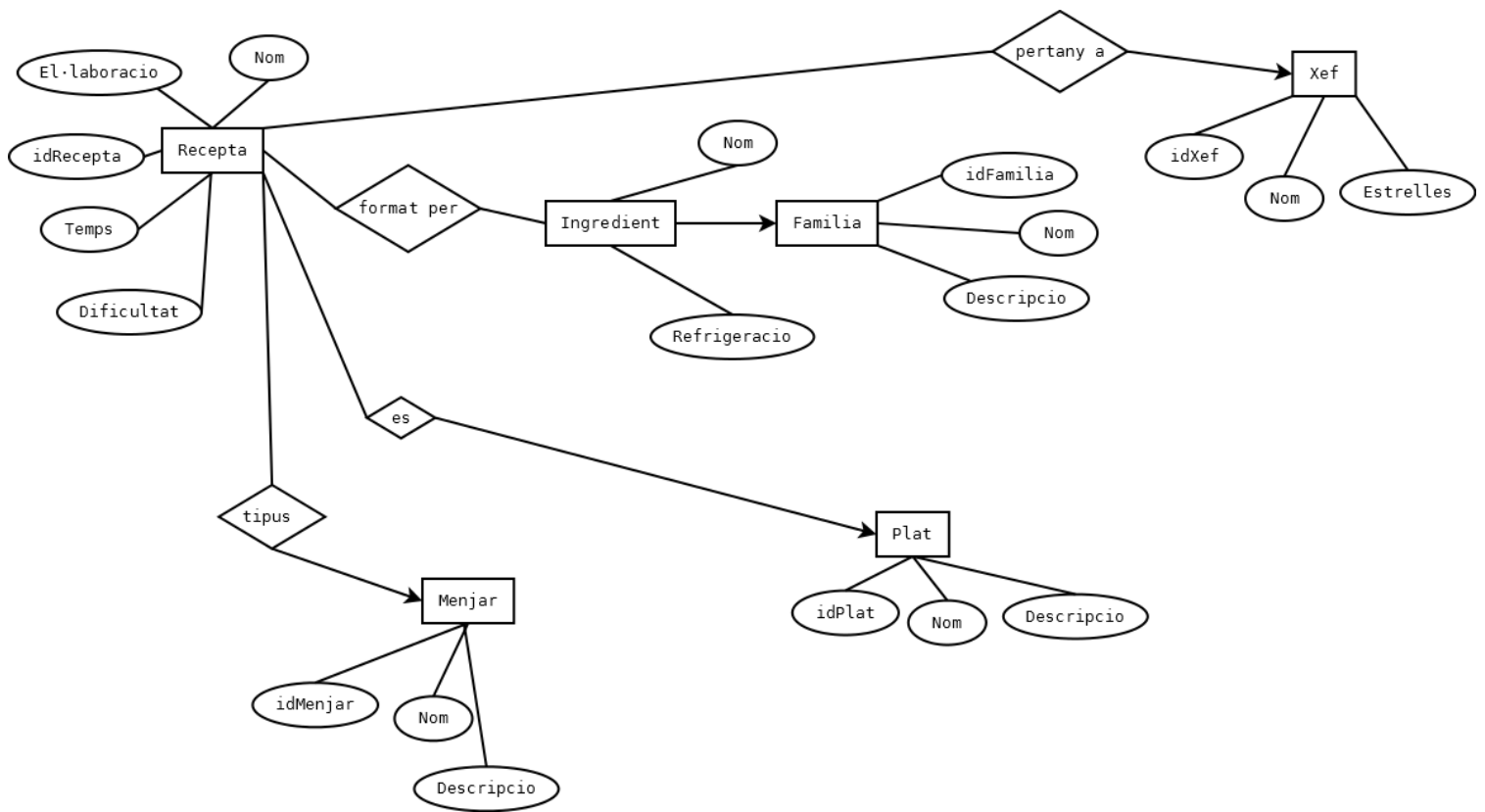
Introducció

En aquesta pràctica realitzarem una aplicació JAVA de receptes de cuina. La idea es aplicar un framework ORM (Hibernate en aquest cas) per tal de gestionar els objectes persistents i així poder treballar amb la base de dades desde l'aplicació JAVA i no amb el client SQL. Utilitzarem SQLite per crear les taules de la base de dades al iniciar l'aplicació.

Objectius

- Entendre que es un ORM i per a que serveix.
- Familiaritzar-nos amb l'ús de Hibernate.
- Integrar una base de dades en la nostra aplicació.

Diagrama Model ER



Desenvolupament

Base de dades amb SQLite

Hem creat una taula per cada entitat que es mostra al diagrama, especificant les primary keys, foreign keys i tots els atributs. Aquestes taules es troben en un fitxer sql (main.sql) que hem fet servir per crear i exportar la nostra base de dades.

Esquema de taules

Menjar (nom:String, id:int, descripcio:String)

Plat (nom:String, id:int, descripcio:String)

Xef (nom:String, id:int, estrelles:int)

Familia (nom:String, id:int, descripcio:String)

Ingredient (nom:String, id:int, refrigeracio:boolean, familia:int) : familia es una foreign key que relaciona les taules ingredient i familia per l'identificador de familia.

Recepta (nom:String, elaboracio:String, id:int, temps:int, dificultat:int, idXef:int, idPlat:int, idMenjar:int) : idXef, idPlat i idMenjar son foreign keys que relacionen la taula recepta amb les taules xef, plat i menjar respectivament.

Format_per(idR:int, idIngr:int) : aquesta taula relaciona una recepta amb els ingredients que la formen.

Aplicació

Model: en aquesta part es troben les classes JAVA i els fitxers xml que fan el mapping aquestes classes amb les taules de la nostra base de dades. Les relacions entre taules també queden configurades als fitxers xml.

També hem fet mapping dels atributs amb el tag "property" i de les primary keys amb el tag "id".

Cada classe JAVA té el seu constructor (amb paràmetres i sense) i els corresponents getters i setters.

Vista: en aquesta part tenim tres classes JAVA.

-ConnectHB: crea un objecte "SessionFactory" i el retorna.

-Menu: Controla les opcions del menú principal i dels submenús. També s'encarrega de retornar-les.

-CuinaUB: instancia l'objecte SessionFactory i llença el menú principal. Controla el flux dels menus i té implementades totes les CRUDs de les taules.

Manual de l'aplicació

Al iniciar l'aplicació s'ens demana que introduïm un user ("admin") i un password ("admin"). Si ens equivoquem a l'hora d'introduir aquestes dues dades, l'aplicació es tanca immediatament.

```
LOGIN
-----
User: admin
Pass: admin
```

Un cop introduïts correctament user i password, veiem el menú principal de l'aplicació.

```
-----
MENU PRINCIPAL
-----
1.- CRUD Recepta
2.- CRUD Menjar
3.- CRUD Plat
4.- CRUD Xef
5.- CRUD Ingredient
6.- Sortir
-----
Entra una opció >> |
```

Anem a veure per exemple "CRUD Menjar". Introduïm el número d'aquesta opció i veurem un submenú com el següent:

```
-----
OPERACIONS CRUD PER TIPUS DE MENJAR
-----
1.- Create
2.- Read
3.- Update
4.- Delete
5.- Tornar al menú anterior
-----
Entra una opció >> |
```

Tots els submenús tenen les mateixes opcions.

- Create: permet inserir a la base de dades un nou menjar.
- Read: mostra tots el menjars que hi ha guardats a la base de dades.
- Update: permet actualitzar el menjar de la base de dades que vulguis.
- Delete: permet eliminar qualsevol menjar que es trobi a la base de dades.