



<b>Nom i Cognom:</b>	<b>Christopher Richard Carrillo Crespo</b>
<b>Enllaç al repositori Git:</b>	<b><a href="https://github.com/ChristopherR17/DAM2_Deberes/tree/main/M06/RA2/Exercici_2.1">https://github.com/ChristopherR17/DAM2_Deberes/tree/main/M06/RA2/Exercici_2.1</a></b>

<b>Objectius:</b> <ul style="list-style-type: none"><li>- Gestionar BD sqlite amb JDBC</li></ul>
<b>Instruccions:</b> <ul style="list-style-type: none"><li>- Desenvolupa el codi necessari per a cada exercici seguint les especificacions indicades. El codi ha de superar els tests proporcionats per garantir-ne la funcionalitat.</li><li>- En cas d'haver de contestar alguna pregunta en aquest document, cal fer-ho dins del quadre indicat.</li></ul>
<b>Criteris d'avaluació:</b> <p>Cada exercici té especificada la puntuació. Es valorarà:</p> <ul style="list-style-type: none"><li>● Format correcte del codi (indentació i compliment de la guia d'estil de Java).</li><li>● Noms clars i descriptius per a mètodes i variables.</li><li>● Qualitat general del codi, amb comentaris explicatius quan sigui necessari.</li><li>● Les còpies seran penalitzades amb una puntuació de 0.</li></ul>
<b>Entrega:</b> <ul style="list-style-type: none"><li>- Repositori Git privat, compartit amb l'usuari jpala4-ieti, que contingui el codi complet de la pràctica.</li><li>- Subdirector "doc" dins del repositori amb el fitxer memoria.pdf.</li><li>- URL del repositori: S'ha de pujar a Moodle.</li></ul>

#### Recursos i materials:

- JDK de Java versió 21, Maven i Git instal·lats.
- Eina de programació: necessiteu una IDE per programar en Java (VS Code, IntelliJ, o una altra).
- Altres recursos: Podeu utilitzar Google per buscar tutorials que us ajudin a resoldre els exercicis.
- Repositori amb exemples d'SQLite:  
<https://github.com/jpala4-ieti/DAM-M0486-Tema2-RA234-ExempleJavaSQLite-25-26>

Punt de partida de la pràctica:

<https://github.com/jpala4-ieti/DAM-M0486-Tema2-Esquelet-Basic-Projecte-Java-Maven-25-26>



**Objectiu del programa:** Desenvolupar una aplicació Java que gestioni una base de dades de personatges i faccions basada en el videojoc "For Honor". Utilitza JDBC i SQLite.

Pots crear les classes que crequis necessàries per organitzar bé l'aplicació.

## Exercici 0 - Aplicació “PR210Honor.java”

### Inicialització de la Base de Dades:

- Verifica si la base de dades existeix.
- En cas contrari, inicialitza la base de dades amb les taules següents:
  - Taula Facció:
    - **id**: Clau primària, tipus enter.
    - **nom**: Cadena de text (varchar) de màxim 15 caràcters.
    - **resum**: Cadena de text (varchar) de màxim 500 caràcters.
  - Taula Personatge:
    - **id**: Clau primària, tipus enter.
    - **nom**: Cadena de text (varchar) de màxim 15 caràcters.
    - **atac**: Número real.
    - **defensa**: Número real.
    - **idFaccio**: Clau forània que enllaça amb l'id de la taula Facció.

### Població de la Base de Dades:

Des del mateix programa `PR210Honor.java` insereix dades a la BD. Pots usar les comandes SQL de l'annex [SQL Inserció de dades a la BD](#)

Per tenir més informació sobre el joc pots consultar: [For Honor - Wikipedia](#) i [fandom](#)

### Funcionalitats del Programa:

- **Mostrar una taula:** Permet l'usuari seleccionar quina taula vol visualitzar.
- **Mostrar personatges per facció:** Llista els personatges pertanyents a una facció específica. Indica el nom de la facció no només l'id.
- **Mostrar el millor atacant per facció\*:** Indica qui personatge té l'atac més alt dins d'una facció seleccionada.
- **Mostrar el millor defensor per facció:** Indica qui personatge té la defensa més alta dins d'una facció seleccionada.
- **Sortir:** Tanca l'aplicació.

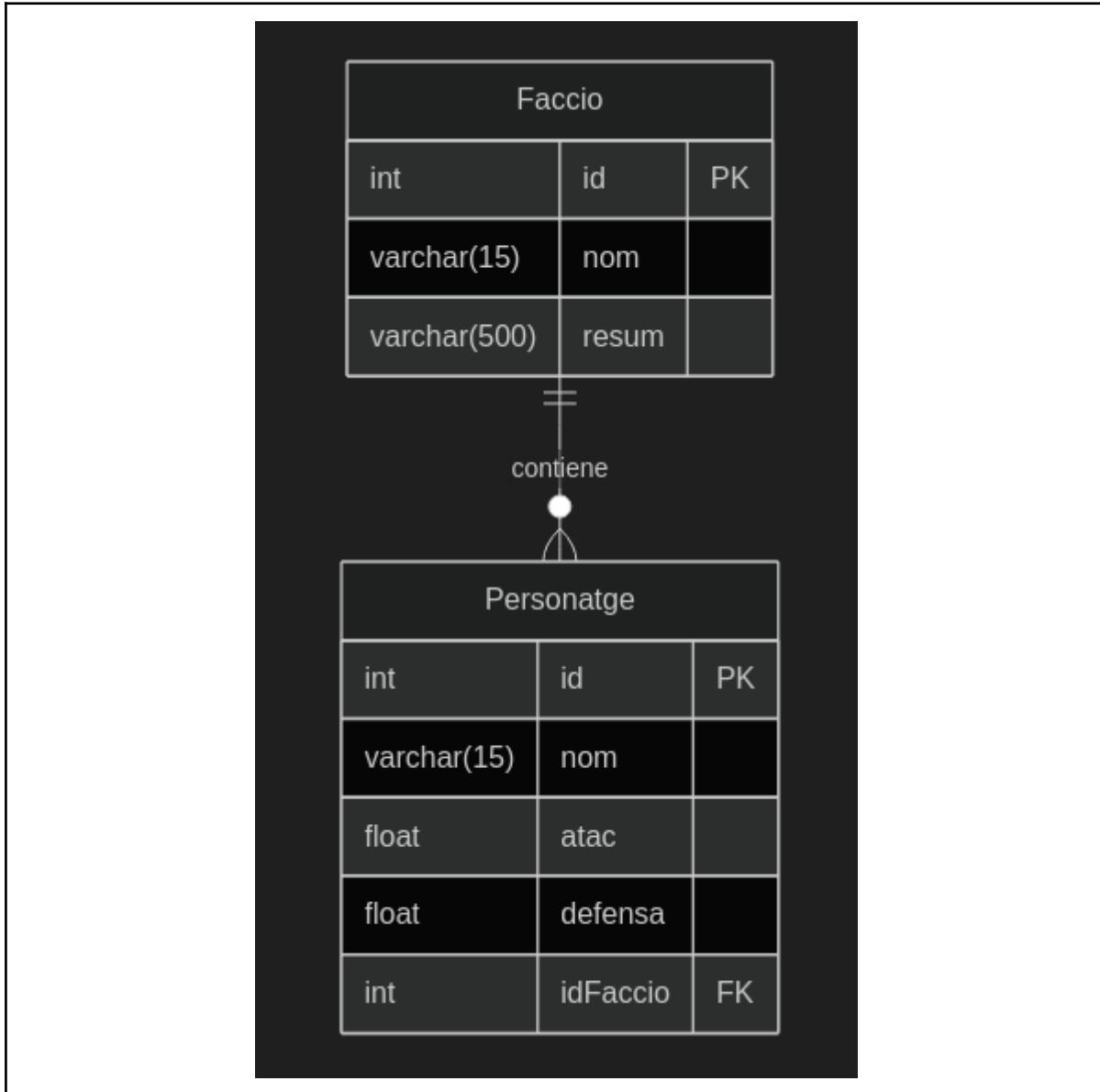
\*En cas d'empat en atac o defensa, mostrar només un dels personatges.

Caldrà que trobis una forma de generar un diagrama de la base de dades.

Ho he fet en markdown, també està en el repositori.

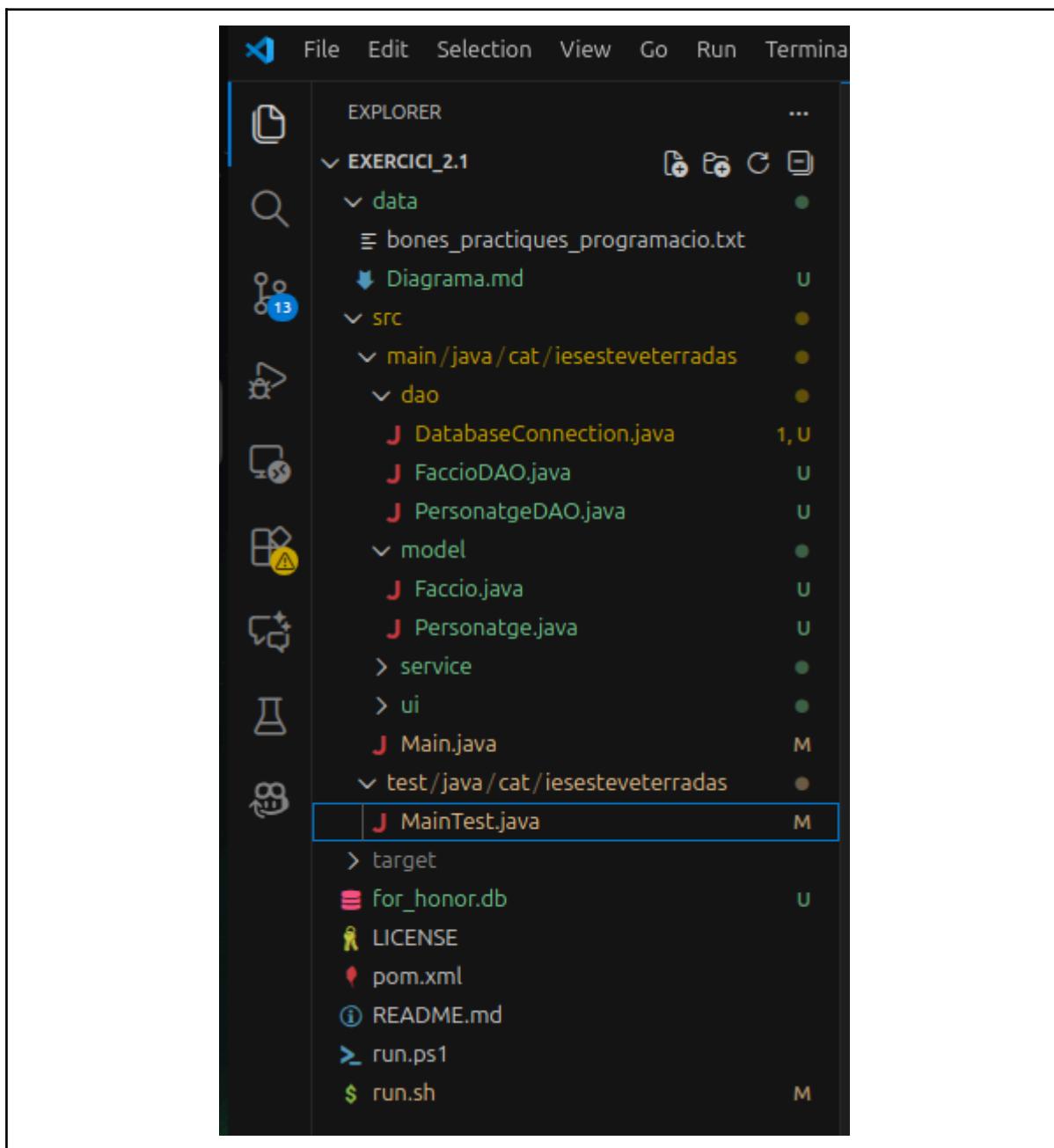


Enganxa en aquest quadre el diagrama de la base de dades



Enganxa una captura de pantalla que mostri l'organització del teu programa (packages, classes) i explica les principals decisions de disseny que has pres.

He implementat una arquitectura en capes que separa clarament les responsabilitats: la capa Model gestiona les entitats de domini (Faccio i Personatge), la capa DAO s'encarrega de l'accés a la base de dades SQLite amb patró Singleton, la capa Service conté la lògica de negoci, i la capa UI gestiona la interfície de consola. Aquesta estructura modular permet una alta mantenibilitat, testabilitat i extensibilitat, complint tots els requisits funcionals mentre aplica principis SOLID.



The screenshot shows a file explorer window with the following structure:

- EXERCICI\_2.1**
  - data**
    - bones\_practiques\_programacio.txt
  - src**
    - main/java/cat/iesesteveterradas**
      - dao**
        - DatabaseConnection.java (1, U)
        - FaccioDAO.java (U)
        - PersonatgeDAO.java (U)
      - model**
        - Faccio.java (U)
        - Personatge.java (U)
      - service**
      - ui**
      - Main.java (M)
    - test/java/cat/iesesteveterradas**
      - MainTest.java (M)
  - target**
  - for\_honor.db** (U)
  - LICENSE**
  - pom.xml**
  - README.md**
  - run.ps1**
  - run.sh** (M)

## Ampliació

- Afegeix sistema de logging similar al del repositori d'exemple
- Modifica l'aplicació per tal que pugui usar MySQL (a proxmox)



## Annexos

### SQL Inserció de dades a la BD

```
INSERT INTO Faccio (nom, resum) VALUES ("Cavallers", "Though seen as a single group, the Knights are hardly unified. There are many Legions in Ashfeld, the most prominent being The Iron Legion.");  
INSERT INTO Faccio (nom, resum) VALUES ("Vikings", "The Vikings are a loose coalition of hundreds of clans and tribes, the most powerful being The Warborn.");  
INSERT INTO Faccio (nom, resum) VALUES ("Samurais", "The Samurai are the most unified of the three factions, though this does not say much as the Daimyos were often battling each other for dominance.");  
  
INSERT INTO Personatge (nom, atac, defensa, idFaccio) VALUES ("Warden", 1, 3, 1);  
INSERT INTO Personatge (nom, atac, defensa, idFaccio) VALUES ("Conqueror", 2, 2, 1);  
INSERT INTO Personatge (nom, atac, defensa, idFaccio) VALUES ("Peacekeep", 2, 3, 1);  
  
INSERT INTO Personatge (nom, atac, defensa, idFaccio) VALUES ("Raider", 3, 3, 2);  
INSERT INTO Personatge (nom, atac, defensa, idFaccio) VALUES ("Warlord", 2, 2, 2);  
INSERT INTO Personatge (nom, atac, defensa, idFaccio) VALUES ("Berserker", 1, 1, 2);  
  
INSERT INTO Personatge (nom, atac, defensa, idFaccio) VALUES ("Kensei", 3, 2, 3);  
INSERT INTO Personatge (nom, atac, defensa, idFaccio) VALUES ("Shugoki", 2, 1, 3);  
INSERT INTO Personatge (nom, atac, defensa, idFaccio) VALUES ("Orochi", 3, 2, 3);
```