

**Nom i Cognoms:** [Joel Berzal Álamo](#)

**URL Repositori GitHub (XML):** [AMS2-MP06-PR2.2-XML-BerzalJoel](#)

**URL Repositori GitHub (JPA):** [AMS2-MP06-PR2.2-JPA-BerzalJoel](#)

**Objectius:**

- Conèixer els arxius de configuració bàsics de Hibernate, i entendre l'entorn de i transactions

**Instruccions:**

- Responen a l'espai de cada pregunta, si ho feu amb diapositives enganxeu la diapositiva en aquest mateix espai.
- Es valorarà la presentació i els comentaris al codi

**Criteris d'avaluació:**

- Cada exercici té la mateixa puntuació

**Entrega:**

- Un arxiu .zip anomenat: **PRx.y-NomCognom.zip**
  - PRx.y correspon al codi de la pràctica, per exemple PR1.1
  - NomCognom correspon al nom i primer cognom de cada participant
- L'arxiu .zip conte:
  - Aquest document emplenat en format .pdf anomenat **memoria.pdf**
  - Els arxius necessaris per fer anar la pràctica

**Materials:**

Necessiteu una eina per programar en JAVA

Feu servir Google per buscar els tutorials que us serveixin millor

**- Exercici 0 -** Fes un programa amb les següents classes configurades per Hibernate amb arxius **.hbm.xml**

“Ciutat.java” que té:  
    long ciutatId  
    String nom  
    String pais  
    int codiPostal

“Ciutada.java” que té:  
    long id  
    long ciutatId  
    String nom  
    String cognom  
    int edat

**Nota:** De moment no posarem relacions entre les taules

“Manager.java”  
    Conté les funcions que permeten les operacions CRUD a través de Hibernate

“Main.java”  
    Crea tres ciutats amb 2 ciutadans cada una (tot i que no hi ha relacions)  
    Llista les ciutats i els seus ciutadans  
    Esborra el segon ciutadà de cada ciutat  
    Esborra la segona ciutat  
    Llista les ciutats  
    Llista els ciutadans

**Nota:** com que no hi ha relació esborrar no afecta les altres taules

**- Exercici 1 -** Fes un programa com l'anterior però amb notacions **JPA**

Per tots dos exercicis, mira aquest exemple de entrada/sortida esperada:

```
Manager.createSessionFactory();

Ciutat ciu0 = Manager.addCiutat("Vancouver", "Canada", 98661);
Ciutat ciu1 = Manager.addCiutat("Växjö", "Suècia", 35220);
Ciutat ciu2 = Manager.addCiutat("Kyoto", "Canada", 5200461);

Manager.addCiutada(ciu0.getCiutatId(), "Tony", "Happy", 20);
Ciutada pep1 = Manager.addCiutada(ciu1.getCiutatId(), "Monica", "Mouse", 22);
Manager.addCiutada(ciu1.getCiutatId(), "Eirika", "Erjo", 44);
Ciutada pep3 = Manager.addCiutada(ciu1.getCiutatId(), "Ven", "Enrison", 48);
Manager.addCiutada(ciu2.getCiutatId(), "Akira", "Akiko", 62);
Ciutada pep5 = Manager.addCiutada(ciu2.getCiutatId(), "Masako", "Kubo", 66);

Collection<?> ciutats = Manager.listCollection(Ciutat.class, "");
for(Object obj: ciutats) {
    Ciutat cObj = (Ciutat) obj;
    System.out.println("Ciutadans de " + cObj.getNom() + ":");
    Collection<?> ciutadans = Manager.listCollection(Ciutada.class,
                                                    "ciutatId=" + cObj.getCiutatId());

    for(Object obj2: ciutadans) {
        Ciutada cObj2 = (Ciutada) obj2;
        System.out.println("    " + cObj2.toString());
    }
}

Manager.delete(Ciutada.class, pep1.getCiutadaId());
Manager.delete(Ciutada.class, pep3.getCiutadaId());
Manager.delete(Ciutada.class, pep5.getCiutadaId());
Manager.delete(Ciutat.class, ciu1.getCiutatId());

System.out.println("Ciutats:");
Collection<?> llista = Manager.listCollection(Ciutat.class, "");
for(Object obj: ciutats) {
    Ciutat cObj = (Ciutat) obj;
    System.out.println("    " + cObj);
}
System.out.println("Ciutadans:");
llista = Manager.listCollection(Ciutada.class, "");
for(Object obj: llista) {
    Ciutada cObj = (Ciutada) obj;
    System.out.println("    " + cObj);
}

Manager.close();
```

```
INFO: HHH000232: Schema update complete
Ciutadans de Vancouver:
  Tony Happy, 20 - 1
Ciutadans de Växjö:
  Monica Mouse, 22 - 2
  Eirika Erjo, 44 - 2
  Ven Enrison, 48 - 2
Ciutadans de Kyoto:
  Akira Akiko, 62 - 3
  Masako Kubo, 66 - 3
Ciutats:
  Vancouver, Canada 98661
  Växjö, Suècia 35220
  Kyoto, Canada 5200461
Ciutadans:
  Tony Happy, 20 - 1
  Eirika Erjo, 44 - 2
  Akira Akiko, 62 - 3
Oct 31, 2022 2:50:01 PM org.hibernate.en
```