



**Nom i Cognoms:** MONICA DOMINGUEZ AGUIRRE

**URL Repositori GitHub:** <https://github.com/monica438/AMS2-MP0486-PR2.2.git>

**Objectius:**

- Conèixer els arxius de configuració bàsics de Hibernate. Programar una aplicació que faci ús d'Hibernate per mapejar la relació entre dues entitats i gestionar les dades.

**Instruccions:**

- Es valorarà la presentació i els comentaris al codi

**Criteris d'avaluació:**

- La puntuació està especificada en cada exercici

**Entrega:**

- Repositori GitHub compartir amb l'usuari **jpala4-ieti**
- En directori "doc" dins del repositori:
  - Aquest document emplenat en format **.pdf** anomenat **memoria.pdf**

**Materials:**

- Necessiteu una eina per programar en JAVA
- Apunts de classe i repositoris d'exemple
- Cerca de tutorials alternatius

**Repositoris de referència:**

- <https://github.com/jpala4-ieti/DAM-M0486-Tema2-JavaHibernateXMLOneToMany>
- <https://github.com/jpala4-ieti/DAM-M0486-Tema2-JavaHibernateJPAOneToMany>
- <https://github.com/jpala4-ieti/DAM-JavaHibernateXMLOneToMany>
- <https://github.com/jpala4-ieti/DAM-JavaHibernateJPAOneToMany>

**Eines**

- <https://gitingest.com/jpala4-ieti/DAM-M0486-Tema2-JavaHibernateXMLOneToMany>
- <https://gitingest.com/jpala4-ieti/DAM-M0486-Tema2-JavaHibernateJPAOneToMany>
- <https://gitingest.com/jpala4-ieti/DAM-M0486-Tema2-PR22-Punt-Partida-25-26>

**Punt de partida:**

- <https://github.com/jpala4-ieti/DAM-M0486-Tema2-PR22-Punt-Partida-25-26>



### Exercici 0

Fes un programa amb les següents classes configurades per Hibernate amb arxius .hbm.xml (4 punts)

"Ciutat.java" que té:

- long ciutatId
- String nom
- String pais
- int poblacio (és el codi postal, només que en el Main està indicat com a població)

Afegir a ciutat els atributs de classe necessaris per establir la relació (Cart. Ítem)

"Ciutada.java" que té:

- long ciutadaId
- String nom
- String cognom
- int edat

Afegir a ciutada els atributs de classe necessaris per establir la relació (Cart. Ítem)

**Nota:** De moment, usarem hibernate però sense establir relacions entre les taules. La relació que s'estableix és lògica, per ids, però no física i més forta com seria el cas de foreign keys.

"Manager.java"

- Conté les funcions que permeten les operacions CRUD a través de Hibernate

"Main.java"

- Crea tres ciutats amb 2 ciutadans cada una (relació lògica a nivell d'identificadors)
- 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, enlloc d'usar el mètode antic per treballar amb hibernate, usa anotacions **JPA (4 punts)**

### Exercici 2

Crea un diagrama de classes que representi la relació entre Ciutat i Ciutadà i genera un diagrama de BD que crea la teva aplicació

- Per generar el diagrama de classes pots usar (o el programari que vulguis)  
<https://online.visual-paradigm.com/es/diagrams/templates/entity-relationship-diagram/erd-example-book-database/>

Enganxa aquí els dos diagrames (Classes i BD)

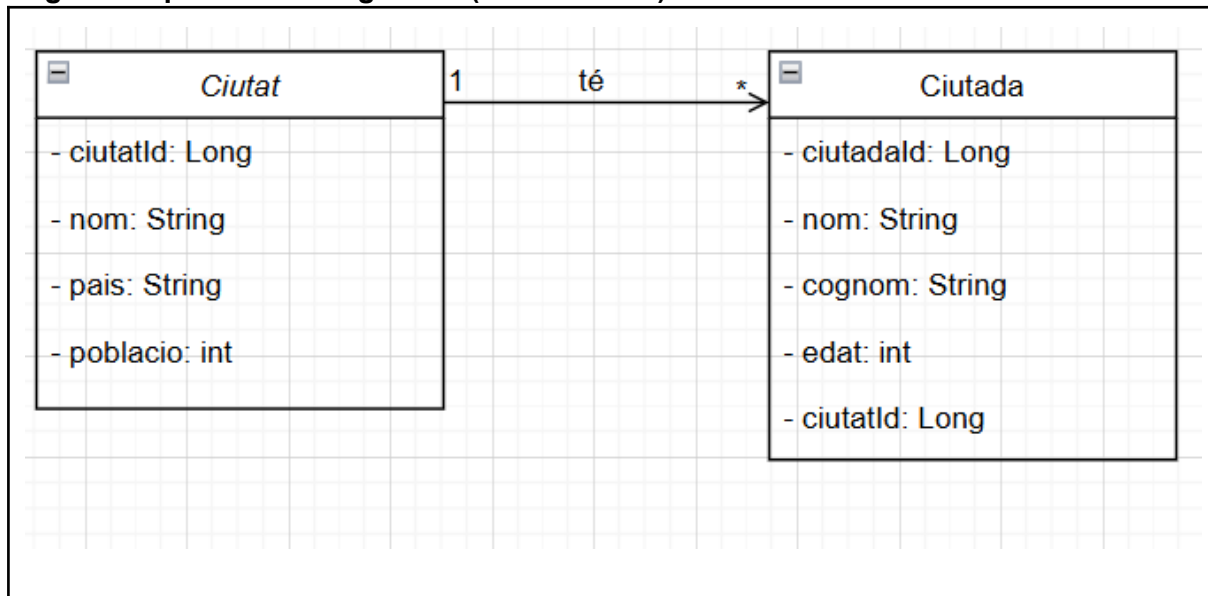




Table: Ciutat		
Columna	Tipus	PK/FK
ciutatId	BIGINT	PK
nom	VARCHAR	
pais	VARCHAR	
poblacio	INT	

Table: Ciutada		
Columna	Tipus	PK/FK
ciutadald	BIGINT	PK
nom	VARCHAR	
cognom	VARCHAR	
edat	INT	
ciutatId	INT	INT



### Exercicis ampliació

- Aconsegueix que el teu exemple pugui funcionar a la vegada amb SQLite i MySQL, permetent triar la BD usant un menú.
- Afegeix algun test simple que realitzi comprovacions d'alguna de les funcions del manager.



## Sortida esperada

Punt 1: Després de la creació inicial d'elements

- 1: Vancouver (Canada), Població: 98661, Ciutadans: []
- 2: Växjö (Suècia), Població: 35220, Ciutadans: []
- 3: Kyoto (Japó), Població: 5200461, Ciutadans: []
- 1: Tony Happy (20 anys)
- 2: Monica Mouse (22 anys)
- 3: Eirika Erjo (44 anys)
- 4: Ven Enrison (48 anys)
- 5: Akira Akiko (62 anys)
- 6: Masako Kubo (66 anys)

Punt 2: Després d'actualitzar ciutats

- 1: Vancouver (Canada), Població: 98661, Ciutadans: [Tony Happy | Monica Mouse | Eirika Erjo]
- 2: Växjö (Suècia), Població: 35220, Ciutadans: [Ven Enrison | Akira Akiko]
- 3: Kyoto (Japó), Població: 5200461, Ciutadans: []
- 1: Tony Happy (20 anys)
- 2: Monica Mouse (22 anys)
- 3: Eirika Erjo (44 anys)
- 4: Ven Enrison (48 anys)
- 5: Akira Akiko (62 anys)
- 6: Masako Kubo (66 anys)

Punt 3: Després d'actualització de noms

- 1: Vancouver Updated (Canada), Població: 98661, Ciutadans: [Tony Updated Happy | Monica Mouse | Eirika Erjo]
- 2: Växjö Updated (Suècia), Població: 35220, Ciutadans: [Ven Updated Enrison | Akira Akiko]
- 3: Kyoto (Japó), Població: 5200461, Ciutadans: []
- 1: Tony Updated Happy (20 anys)
- 2: Monica Mouse (22 anys)
- 3: Eirika Erjo (44 anys)
- 4: Ven Updated Enrison (48 anys)
- 5: Akira Akiko (62 anys)
- 6: Masako Kubo (66 anys)

Punt 4: després d'esborrat

- 1: Vancouver Updated (Canada), Població: 98661, Ciutadans: [Tony Updated Happy | Monica Mouse | Eirika Erjo]
- 2: Växjö Updated (Suècia), Població: 35220, Ciutadans: [Ven Updated Enrison | Akira Akiko]
- 1: Tony Updated Happy (20 anys)
- 2: Monica Mouse (22 anys)
- 3: Eirika Erjo (44 anys)
- 4: Ven Updated Enrison (48 anys)
- 5: Akira Akiko (62 anys)

Punt 5: Recuperació de ciutadans d'una ciutat específica

Ciutadans de la ciutat 'Vancouver Updated':

- Tony Updated Happy
- Monica Mouse
- Eirika Erjo