

# ACCÉS A BBDD VIA ORM JPA

**ACCÉS A DADES (ADA) 23/24**  
CFGs DAM

**PRÀCTICA FINAL AVALUABLE (OBLIGATÒRIA)**

## VISIÓ GENERAL

Hauràs de crear una aplicació Java de manera individual que utilitzi els conceptes assolits durant la UNITAT JPA i complisca amb les especificacions, entregar el codi per AULES dins del termini i defensar-hi a una entrevista individual amb el responsable del mòdul.

## CALENDARI I EXPECTATIVES

- Aquesta pràctica suposa el 100% de la nota en la unitat de JPA.
- Lliurament: 21 de gener a les 23:55h
- Defensa (segons ordre previ): 23, 24 i 25 de gener
- Els endarreriments del lliurament suposaran penalitzacions en la nota segons les indicades en el document de presentació del mòdul.

## QUALIFICACIÓ

Sumant els punts indicats a cada requisit, has d'obtenir 5 sobre 10 i un APTÉ a la defensa per superar la unitat.

El projecte ha de ser FUNCIONAL per tal de poder ser corregit (ha de compilar correctament i poder ser executat)

Un codi perfecte però dubtosament defensat o sense justificacions clares suposarà el suspens directe!!!

**Un zero a un requisit marcat com M (mínim) invalida la prova i nota=0**

## RECURSOS

Has de fer una lectura exhaustiva de tots els materials proporcionats a AULES així com de les tasques no avaluable, però també cercar en Internet per trobar exemples que proporcionen resultats similars als requerits per aquesta tasca.

Copia i pega codi de QUALSEVOL recurs sempre que entengues cada part del mateix ja que se't demanarà que defenses el teu treball en una entrevista individual.

## PLAGI

En cas de sospita de plagi, podria exigir-se una entrevista oral addicional a TOTS els estudiants implicats.

## LLIURAMENT I RETROALIMENTACIÓ

Si no has obtingut un 5 sobre 10 a la activitat o la defensa fora a NO APTA hauràs de superar la UNITAT a la convocatòria ordinària o, amb la resta del mòdul, a la extraordinària.

## MOLT IMPORTANT

### RECOMANACIONS:

1. Revisa les puntuacions abans de començar per saber quins requisits essencials per superar la prova i quins son per pujar nota.
2. A les empreses el codi ha de ser auto explicatiu, i no hem d'abusar dels comentaris, però ací pots afegir tots els comentaris que consideres en castellà, valencià o anglés (sempre tots a la mateixa llengua). T'ajudaran MOLT a la defensa i farà la correcció més senzilla.

### OBLIGATORI (per poder superar la prova):

1. Usa SOLAMENT un idioma al codi (variables, mètodes, classes...).
2. Entrega únicament en zip/rar via AULES.
3. Usa PREFERENTMENT NETBEANS en Java Maven (amb arxiu POM).
4. Versió de Java 11 o inferior.
5. No es permet l'ús de finestres (entorn gràfic).

### A) L'APLICACIÓ

**Es demana un CRUD en Java que accedisca a MySQL per gestionar la base de dades que pots consultar al final d'aquest document.**

1. La base de dades haurà d'anomenar-se «SerpisFP» i l'accedirem des de Java Maven amb les credencials i els ports per defecte.
2. Hauràs de configurar la base de dades de manera que es puguin gestionar les 5 taules: GRUPO, ALUMNO, MODULO, MATRÍCULA i PROYECTO. Cada taula tindrà les dades que pots consultar al final d'aquest document. Cada taula tindrà com a sufix «\_XXZZ» on «XX» son la inicial del teu nom + la inicial del teu primer cognom i ZZ el teu DIA de naixement.
3. Cal proporcionar l'script d'inserció de dades per fer les proves.

**Exemple:** Com Ana Pina va naixer un 22 de març. La taula **GRUPO\_XXZZ** serà **GRUPO\_AP22**

## B) MENÚ PRINCIPAL

El menú serà opcional i tindrà els següents requisits:

1. Llistat d'opcions (menú) amb la darrera opció per EIXIR.
  - a) En processar l'opció, haurem de polsar una tecla (tecla d'espera).
  - b) Després de la tecla, tornarem al menú fins que es polse l'opció EIXIR.

## C) OPCIONS

[L'ORDRE DELS ELEMENTS **NO** ÉS RELLEVANT]

Menu hecho, falta la parte de JPQL

1. Capçalera amb contador d'elements a cada una de les 4 entitats.
2. Opció de menú / Mètode per **BUIDAR** totes les taules.
3. Opcions de menú / Mètodes per **AFEGIR** nous elements manualment (5 taules):
  - a) Demanar les dades de cada ítem en bucle fins que l'ID siga zero.
    - Assegurar-se que es requereixen els camps claus (PK, FK i NOT NULL)
  - b) Ha d'acceptar-se valors a blanc, excepte si és un indentificador o té restricció de nul.
4. Opcions de menú / Mètodes per **LLISTAR** els elements (5 taules):
  - Mostrarem tots els elements de l'entitat seleccionada:
    - a) **BÀSIC:** Les dades del ítem. Listar Normal
    - a) **COMPLET:** BÀSIC + Totes les dades de les **TAULES** relacionades.
5. Opcions de menú / Mètodes per **ESBORRAR** els ítems (5 entitats):
  - Opcionalment, podrem indicar quins ítems concrets volem esborrar
    - a) **BÀSIC:** Tots, els ítems. Si no hi ha cap, mostrar missatge.
    - b) **COMPLET:** BÀSIC + llistat de diversos ítems. Si no estan, mostrar missatge.
  - Tindrem aquestes precaucions: *Que quiere decir + listado de diversos items*
    - No podrem esborrar cap GRUP que tinga ja ALUMNES.
    - No podrem esborrar de cap GRUP els ALUMNES que presenten PROJECTE.
    - Si esborrem un MÒDUL, haurem d'esborrar tots els ALUMNES matriculats en aquest.
    - Excepció: Es podran esborrar només els ALUMNES que NO presenten PROJECTE si només estan matriculats del mòdul de codi "FCT", doncs no assisteixen a classe (ja que aquesta base de dades és fonamentalment per controlar l'assistència presencial a classe).

Importante

On delete cascade

## D) RÚBRICA 0,9

SETUP	M 0.20p Maven ✓	0.20p Contadors ✓	0,5p Menú ✓
	LLISTAT	ESBORRAT	CREACIÓ
GRUPO	M 0.15p Bàsic 0.25p COMPLET	M 0.50p Bàsic 0.10p COMPLET	M 0.25p Bàsic 0.25p COMPLET ✓
ALUMNO	M 0.15p Bàsic 0.25p COMPLET	M 0.50p Bàsic 0.10p COMPLET	M 0.25p Bàsic 0.25p COMPLET
MÓDULO	M 0.20p Bàsic 0.25p COMPLET	M 0.50p Bàsic 0.10p COMPLET	M 0.25p Bàsic 0.25p COMPLET
PROYECTO	M 0.20p Bàsic 0.35p COMPLET	M 0.50p Bàsic 0.10p COMPLET	M 0.25p Bàsic 0.25p COMPLET
MATRÍCULA	M 0.35p Bàsic 0.55p COMPLET	0.30p Bàsic 0.50p COMPLET	M 0.25p Bàsic 0.50p COMPLET
SUBTOTALS	M 1.25p + 1.85p	M p + 1.4p	M 1.25p + 2p
TOTALS	M 4,5.p + M 0.50p Modularitat, aplicació de conceptes JPA, clean code + 5.00p		

Un zero a un requisit marcat com M (mínim) invalida la prova

## E) LES DADES

Aquesta és l'estructura de dades que haureu de gestionar:

```
DROP DATABASE IF EXISTS SerpisFP;
CREATE DATABASE SerpisFP;
USE SerpisFP;
DROP TABLE IF EXISTS GRUPO_XXZZ;
DROP TABLE IF EXISTS ALUMNO_XXZZ;
DROP TABLE IF EXISTS PROYECTO_XXZZ;
DROP TABLE IF EXISTS MODULO_XXZZ;
DROP TABLE IF EXISTS MATRICULA_XXZZ;
```

```
CREATE TABLE GRUPO_XXZZ
(
  CODGRUPO INT,
  DESCRIPCION VARCHAR (50),
  AULA VARCHAR (10),
  PRIMARY KEY (CODGRUPO)
);
```

Todos los  
metodos de  
esta tabla estan  
hechos

Base de datos ya realizada, script  
mejorado guardado

```
CREATE TABLE ALUMNO_XXZZ
(
  NIA VARCHAR (10),
  NOMBRE VARCHAR (50),
  APELLIDOS VARCHAR (50),
  CODGRUPO INT NOT NULL,
  PRIMARY KEY (NIA ),
  FOREIGN KEY (CODGRUPO) REFERENCES GRUPO_XXZZ (CODGRUPO)
```

La clase matricula no se crea en el proyecto, se hace con un @JoinColumn

```
);
```

```
CREATE TABLE MODULO_XXZZ
(
  CODMODULO INT,
  DESCRIPCION VARCHAR (50),
  NUMHORAS INT,
  PRIMARY KEY (CODMODULO)
);
```

Cuando se suma un alumno, cuando  
acabe ,  
preguntamos si queremos matricularlo en  
un modulo, donde haremos  
que se cree la matricula automaticamente.

```
-- (N-N ALUMNO Y MÓDULO)
CREATE TABLE MATRICULA_XXZZ
(
  NIA VARCHAR (10),
  CODMODULO INT,
  DESCRIPCION VARCHAR (50),
  PRIMARY KEY (NIA,CODMODULO),
  FOREIGN KEY (NIA) REFERENCES ALUMNO_XXZZ (NIA),
  FOREIGN KEY (CODMODULO) REFERENCES MODULO_XXZZ (CODMODULO)
);
```

```
CREATE TABLE PROYECTO_CONVOCATORIA_XXZZ
(
  CODPROYECTO VARCHAR (10),
  TITULO VARCHAR (200),
  NIA VARCHAR (10) NOT NULL UNIQUE,
  PRIMARY KEY (CODPROYECTO),
  FOREIGN KEY (NIA ) REFERENCES ALUMNO_XXZZ (NIA)
);
```

**(\*) LA DEPENDÈNCIA D'EXISTÈNCIA A LA RELACIÓ 1:N I LA PARTICIPACIÓ MÍNIMA A LA RELACIÓ N:M ES GESTIONARAN DIRECTAMENT A JPA**

**D) LES DADES DE PROVA**

Com a mínim caldrà afegir les següents dades (completar en cas de camps claus o no nuls)

GRUP: 2DAMC, 2DAMR, 2ASIRC

ALUMNO: tu mateixa i els 4 companys que van darrere de tu per ordre alfabètic. Si no està el delegat, afegeix-lo. Entenem que es un llistat circular. Afegeix també a l'alumne NIA 123456

MODULO: tots els de segon curs de 2DAM

Finalment, matriculat a tu del mòdul ADA i al delegat de PSP, ADA i a mes d'un projecte amb codi 1234 i títol " App per a reptes personals"

Esto se deja para el final, ya que introducir datos no es relevante a la hora de desarrollar el codigo principal.

**E) LES PROVES SUGGERIDES (MÍNIMES) QUE ES REVISARAN A LA DEFENSA**

La aplicacion tiene que cumplir todos estos requisitos al acabarla. Probarlos de uno en uno.

Proves (per la defensa):

1. Afegir els elements de mostra en aquest ordre: grup, alumne, projecte mòdul.
  - **RESULTAT: Sense problemes**
2. Llistar, una a una, totes les taules (amb i sense filtres) **A que se refiere con filtros**
  - **RESULTAT: Sense problemes**
3. Afegir 1 nou element de cada entitat en aquest ordre: ALUMNE i GRUP
  - **RESULTAT: No es pot afegir un alumne si no existeix prèviament el grup.**
4. Afegir 1 nou element de cada entitat en aquest ordre: PROJECTE, ALUMNE
  - **RESULTAT: No es pot afegir un alumne si no existeix prèviament l'alumne..**
5. Esborrar un mòdul
  - **RESULTAT: Comprovar que s'esborren les matrícules dels alumnes en aquest mòdul**
6. Esborrar un grup que té alumnes,
  - **RESULTAT: No es pot esborrar aquest grup perquè té alumnes matriculats.**
7. Esborrar un alumne que està matriculat d'algun mòdul
  - **RESULTAT: Per favor, desmatricule a aquest alumne abans per poder esborrarlo.**
8. Esborrar un alumne que només està matriculat d'FCT i no presenta projecte.
  - **RESULTAT: Sense problemes**
9. Esborrar un alumne que té més mòduls matriculats a més d'FCT i/o presenta projecte.
  - **RESULTAT: Per favor, comprove que l'alumne compleix tots els requisits per poder ser esborrat**

Los resultados son las posibles excepciones que imprimiremos por pantalla