

INSTITUTO DE EDUCACIÓN SECUNDARIA ISIDRA DE GUZMÁN

PRÁCTICA DE EVALUACIÓN CONTINUA (PEC) UNIDAD DE TRABAJO 3

Módulo Profesional: Acceso a Datos

Ciclo Formativo: Desarrollo de Aplicaciones Multiplataforma (2º curso)

Profesor: Ruth Lospitao Ruiz (ruth.lospitao@educa.madrid.org)

Curso Académico: 2025-2026

Práctica de Evaluación Continua (PEC) - Unidad de Trabajo 3

Módulo: Acceso a Datos (2.^º DAM)

Valor: 60 % de la nota del RA3

Peso RA3: 20%

Contexto

En la **Unidad de Trabajo 2** desarrollaste una mini-aplicación Java utilizando **JDBC** para gestionar una base de datos relacional en MySQL.

En esta **Unidad de Trabajo 3**, deberás **evolucionar esa misma aplicación** para que el acceso a datos se realice mediante un **ORM**, utilizando **Hibernate como proveedor de Jakarta Persistence (JPA)**.

El objetivo es comprobar que sabes:

- sustituir JDBC por un ORM,
- definir correctamente el mapeo objeto-relacional,
- trabajar con entidades persistentes,
- gestionar transacciones,
- realizar operaciones CRUD y consultas mediante **JPQL/HQL**.

Deberás reutilizar **el mismo dominio de la PEC de JDBC**. Ejemplos válidos (según la PEC anterior):

- Gestor de pedidos de una cafetería.
- Sistema de reservas y clientes.
- Gestor de inventario y ventas.
- Agenda académica.

La base de datos podrá ser:

- la misma utilizada en JDBC, o
- una nueva base de datos equivalente adaptada a ORM.

Objetivos de la PEC

Desarrollar una **mini-aplicación Java (Maven)** que gestione la persistencia de datos mediante **Hibernate/JPA**, aplicando correctamente los conceptos de:

- entidades,
- mapeo ORM,
- estados de las entidades,
- relaciones,
- operaciones CRUD,
- consultas con JPQL,
- transacciones.

Requisitos Generales

- Versión de Java: 17 o superior.
- Proyecto Maven (estructura estándar).
- Hibernate como proveedor ORM
- Gestión de la persistencia
 - Uso obligatorio de EntityManagerFactory para la creación del contexto de persistencia.
 - Uso de EntityManager para todas las operaciones CRUD y consultas ORM.
 - Gestión explícita de transacciones mediante EntityTransaction en todas las operaciones que modifiquen la base de datos (commit / rollback).
- Jakarta Persistence (JPA) Configuración correcta del ORM en persistence.xml
- Base de datos MySQL
- No se permite usar break, continue ni return para salir de estructuras de control.
- Se recomienda usar try-with-resources en todas las operaciones de E/S.
- Comentarios internos y javadoc
- Comentarios en script sql
- Buenas prácticas de programación
- Fichero readme del Proyecto
- Validaciones de datos

Requisitos funcionales mínimos

La aplicación deberá implementar, como mínimo:

Entidades y mapeo

- Definición de **al menos dos entidades persistentes**.
- **Al menos una relación entre entidades.** Uso correcto de cascade cuando proceda.

Operaciones CRUD

Desde un **menú por consola**, deberán poder realizarse:

- Alta (CREATE)
- Consulta (READ)
- Modificación (UPDATE)
- Eliminación (DELETE)

Las operaciones de escritura deberán:

- ejecutarse dentro de transacciones,
- realizar commit o rollback según corresponda.

Consultas ORM

- Al menos **dos consultas JPQL/HQL**, por ejemplo:
 - listado filtrado,
 - búsqueda por criterio,
 - consulta sobre relaciones.
- Los resultados deberán almacenarse en **objetos o listas tipadas**, no en estructuras genéricas.

Entrega

El alumno deberá subir al aula virtual un archivo ZIP con la estructura del proyecto, la cual puede ser similar a:

Apellidos_Nombre_PEC_UT3.zip

```
|  
|--- ProyectoMaven/  
|     |--- src/main/java/com/.../...  
|     |--- pom.xml  
  
|--- Memoria_Tecnica.pdf
```

La memoria técnica, cumplir con la plantilla que se adjunta

Importante

Se pueden utilizar herramientas de apoyo, incluidas aplicaciones de inteligencia artificial, para resolver dudas o mejorar la presentación de su trabajo. Sin embargo, **todo lo que entreguéis debéis comprenderlo completamente.**

Se realizará una defensa individual del proyecto (código y memoria), donde se le harán preguntas conceptuales y funcionales de su solución. Si no se demuestra entender el funcionamiento del proyecto entregado o de las partes, implicará automáticamente la **calificación de suspenso** en la práctica, aunque el código funcione correctamente.