

Simulacro de Parcial – Backend de Aplicaciones

Pre - enunciado

Contexto

En esta simulación se propone preparar un proyecto que sienta las bases para el trabajo que será requerido en el día del simulacro. El objetivo es ejercitar la inicialización de una base de datos y el armado de la capa de acceso a datos en un entorno **Java + H2 + JPA**.

Como hemos mencionado el día del simulacro se entregará un archivo **CSV** con datos reales. A partir de este archivo deberá realizarse lo siguiente:

1. Popular la base de datos en memoria con los registros del CSV.
2. Desarrollar consultas sobre la base poblada.
3. Validar los resultados mediante **tests unitarios**.

Consigna del Pre-Parcial

1. Proyecto base

- El proyecto debe estar desarrollado en **Java 21** (Maven).
- La base de datos a utilizar debe ser **H2 en memoria**.
- Se deben incluir las dependencias necesarias para **JDBC, JPA/Hibernate, JUnit**.
- El registro de ejecución puede implementarse inicialmente con `System.out.println`, sin necesidad de utilizar aún un framework de logging.

2. Inicialización de la base

- Al iniciar la aplicación, la base de datos debe inicializarse con la estructura suministrada en el archivo `ddl.sql`.
- El archivo `ddl.sql` debe ejecutarse automáticamente, de modo que las tablas y secuencias queden disponibles al iniciar la aplicación.
- Esto debe funcionar del mismo modo que ya lo hemos documentado e implementado en clase.

El archivo `ddl.sql` será provisto junto con este enunciado.

3. Mapeos de entidades

- A partir de la estructura definida en `ddl.sql`, deben desarrollarse las entidades Java correspondientes con **JPA**.
- Los nombres de columnas y claves primarias deben respetarse según la definición del script, considerando que no siempre siguen la convención simple (`DESA_ID`, `GEN_ID`, `PLAT_ID`, `JUEGO_ID`).
- Debe implementarse el mapeo de relaciones entre entidades (`JUEGO` → `GENERO`, `DESARROLLADOR`, `PLATAFORMA`).

4. Capa de acceso a datos

- Puede desarrollarse una capa de persistencia y acceso a datos que permita:
 - Insertar nuevos registros.
 - Recuperar registros por clave primaria.
 - Listar colecciones completas de entidades.
 - Realizar consultas filtradas (por ejemplo, juegos de un género específico o de un desarrollador determinado).
- La implementación puede resolverse mediante el patrón de **repositorios JPA** documentado en la actividad de clase, o según el alumno desida, pero esto es todo lo que el alumno puede tener implementado el día del parcial.

Deseable en el momento de iniciar el simulacro

Se refiere a lo que el alumno debería tener implementado para que el tiempo suministrado en el simulacro (día del parcial) sea suficiente

- Proyecto con dependencias configuradas.
- Archivo `ddl.sql` dentro de `src/main/resources`.
- Clases de entidades con sus mapeos.
- Capa de acceso a datos implementada.
- (Opcional) Test inicial que valide la correcta inicialización de la base y la persistencia/recuperación de al menos una entidad.

Se deja constancia que los tiempos calculados para la evaluación asumen que el alumno ya implementó y tienen funcionando sin problemas lo que aquí mencionamos y recordamos que es responsabilidad del alumno que el esquema implementado funcione sin problemas.

Recomendaciones

- Se recomienda revisar el script `ddl.sql` y analizar cómo se mapearán las claves primarias y foráneas.
- Debe prestarse atención a los nombres de columnas, que no siguen siempre la convención más simple.
- Conviene practicar el uso de `enum` para mapear la columna `CLASIFICACION_ESRB`.
- Es deseable organizar el proyecto en capas diferenciadas: entidades, repositorios y, un paquete donde se ubicaran las clases que se soliciten.

Recordatorio

En el simulacro (y también en el parcial), además de inicializar la base con este esquema, será necesario:

1. **Cargar los datos desde un archivo CSV.**
2. **Obtener resultados específicos.**
3. **Validar los resultados mediante tests unitarios.**

La preparación realizada en esta simulación servirá para optimizar el tiempo disponible durante el examen.