

**NOMBRE Y APELLIDOS:**

FECHA: 12/02/2025

**CURSO:** 2º DAM

**MÓDULO:** Acceso a datos (AAD)

**UNIDADES DIDÁCTICAS:** UD05

**CALIFICACIÓN**

## Examen Parcial 2<sup>a</sup> Evaluación (120 min)

## Instrucciones

---

**No se permite el uso de ningún material de consulta ni el acceso a Internet.**

Antes de comenzar el examen, el alumno debe:

1. Leer el enunciado del examen completo.
2. Descargar los recursos del AV necesarios para la realización del examen.
3. Desconectar el latiguillo de red de su PC.
4. Cerrar todos los proyectos abiertos del Eclipse (desde la sección “package explorer” -> botón derecho sobre cada proyecto -> close Project)
5. Una vez cerrados todos los proyectos, se deberá aplicar un filtro para ocultar los proyectos cerrados (desde la sección “package explorer” -> View menu -> Filters -> seleccionar “closed projects”)

## Ejercicios

---

Se dispone de una BD **FCTs** desplegada en un SGBD relacional (MySQL) en la que encontramos información sobre alumnos, empresas, tutores, habilidades y las propias prácticas de formación profesional.

Crea un nuevo proyecto maven llamado **AAD\_EX03\_NomApe1** donde *Nom* sea tu nombre y *Ape1* tu primer apellido (evita acentos y eñes). Crea dentro un paquete llamado **es.NomApe1** donde desarrolles un programa en JAVA que conecte con la Base de Datos MySQL proporcionada, usando la API estándar de *Jakarta Persistence* (JPA) mediante el framework de *Hibernate* y que cumpla con los requisitos que se detallan a continuación:

- a. La aplicación deberá conectar con una Base de Datos de forma transparente para el usuario. Los datos de conexión deben configurarse en un fichero aparte (*persistence.xml*) con los siguientes parámetros:

- BD: **FCTs**
- Usuario: **examen**
- Pwd: **examen**

- b. La aplicación deberá ser *traceada* con un log (logback) donde además de las operativa *Hibernate*, se añadan las acciones más relevantes de forma que facilite la detección de errores, así como el añadido de futuros cambios. Debe hacerse uso de los distintos niveles de granularidad en función de la acción a tracear (**trace, debug, warn, error, info**) Dicho log debe llamarse **AAD\_EX03\_<NomApe1>.log** donde nom\_ape1 será el nombre y primer apellido del alumno.
- c. La aplicación debe usar las clases proporcionadas en el AV, añadiendo el paquete, interfaces, métodos o atributos que el usuario considere necesarios para poder cumplir con los requisitos funcionales que se indican a continuación.
- d. La aplicación debe cumplir con los siguientes requisitos funcionales. Cada uno de ellos debe ser gestionado en una sesión o contexto de persistencia.

### 1) Alta de alumno y sus datos de contacto: (4 puntos)

Se darán de alta alumnos, en caso de no existir, junto con sus datos de contacto usando operaciones en cascada. Al menos se deberá dar de alta los siguientes alumnos con sus correspondientes datos de contacto:

<b>Id</b>	<b>Nombre</b>	<b>Apellido</b>	<b>Fecha nac</b>	<b>email</b>	<b>teléfono</b>
101	Dario	Pazos	15/06/05	dario.pazos@instituto.es	912443213
102	Fermin	Sanz	21/09/02	fermin.sanz@instituto.es	6660006666

### 2) Consulta y modificación de una empresa: (3 puntos)

Se recuperará la empresa cuyo **id = 1** para modificar su dirección y posteriormente se deberá volcar en el LOG todos sus datos junto con los tutores que tenga asociados **sin usar HQL**.

La nueva dirección de la empresa 1 será: Carretera de Colmenar Km 12,800

### 3) Listar las empresas y sus tutores, pero únicamente los que no sean tutores de instituto: (3 puntos)

Se debe volcar la información en el LOG haciendo uso de HQL. Únicamente se desea visualizar el nombre de la empresa, el nombre del tutor y el email de contacto de la empresa.

## Tips & Tricks

---

Puedes apoyarte en los ficheros de ejemplo subidos en el AV. Los cuales tendrás que adaptar para cumplir con los requisitos del examen:

- Persistence.xml (Configuración de *Jakarta persistence*)
- Pom.xml (dependencias Maven)
- Logback.xml (Configuración del log)
- Ficheros JAVA correspondientes a las clases POJO.

## Entrega:

---

Cada alumno/a entregará en el Aula Virtual de la asignatura dos ficheros: un fichero comprimido **en formato jar** donde se incluya el código fuente. Un segundo fichero en formato “**Runnable JAR file**” que incluirá todas las librerías necesarias para la ejecución correcta de la aplicación.

El primer fichero deberá nombrarse con el nombre original de la tarea, indicando a continuación un guion bajo y el nombre de usuario del/la alumno/a. Con extensión **jar**. El segundo fichero se nombrará igual que el primero, pero añadiendo “**runnable**” para diferenciarlo del primero. Con extensión **jar**.

Ejemplo:

- En este caso la tarea se llama *AAD\_EXAMEN\_ORM\_Hibernate*.
- Si el usuario **jose.sala** es quien realiza la entrega, los ficheros deberán llamarse de la siguiente forma:
  - *AAD\_EXAMEN\_ORM\_Hibernate\_jose\_sala.jar*
  - *AAD\_EXAMEN\_ORM\_Hibernate\_runnable\_jose\_sala.jar*

Para generar los ficheros JAR debes adaptar tu pom.xml de Maven de acuerdo al fichero pom.xml que tienes disponible en el AV.

Actualiza el proyecto.

Desde **Maven Build**, indica como **Goals** *clean package* para generar los ficheros jar. Dichos ficheros los encontrarás en el directorio target del proyecto.

## Criterios de evaluación:

---

- No hay errores de compilación.
- Cumple con los requisitos exigidos en cada ejercicio.
- Se adelanta y gestiona posibles errores provocados por el usuario controlando número de parámetros, tipos, etc.
- Se entrega en el Aula virtual toda la información requerida, en el plazo establecido y con la nomenclatura y formato exigido.
- Se hace uso de las **buenas prácticas de programación** (evitando hardcodes, parametrizando la aplicación, optimizando el uso de recursos, modularizando funcionalidades en métodos, reutilizando los recursos siempre que sea posible, organizando en clases la información y funcionalidades...)
- Se gestionan correctamente las posibles **excepciones** que se puedan disparar en el programa evitando su propagación desde el método main.
- Se añade **trazas** (con niveles **debug, trace, warn, error, info**) a sus programas para facilitar futuras correcciones o cambios de alcance.

La calificación de dicha práctica se tendrá en consideración para la evaluación de los siguientes **Resultados de Aprendizaje y Criterios de Evaluación**:

RA3. Gestiona la persistencia de los datos identificando herramientas de mapeo objeto relacional (ORM) y desarrollando aplicaciones que las utilizan.

- a) Se ha instalado la herramienta ORM.
- b) Se ha configurado la herramienta ORM.
- c) Se han definido configuraciones de mapeo.
- d) Se han aplicado mecanismos de persistencia a los objetos.
- e) Se han desarrollado aplicaciones que modifican y recuperan objetos persistentes.
- f) Se han desarrollado aplicaciones que realizan consultas usando el lenguaje SQL.
- g) Se han gestionado las transacciones.