



Tabla de contenido

1 Introducción	3
2 Requisitos del proyecto.....	3
3 Desarrollo.....	4
4 Temporalización.....	5
5 Entrega y defensa	5
6 Evaluación	6
7 Rúbrica de corrección.....	7

1 Introducción

A lo largo del final del tercer trimestre se realiza un proyecto personal e individual donde se realiza el diseño y desarrollo de una aplicación “standalone” escrita en Java con interfaz gráfica y que se conecta a bases de datos.

Este proyecto supone la finalización del curso y su enfoque práctico proporciona:

- Un excelente mecanismo de consolidación de contenidos.
- Una eficaz herramienta de valoración del nivel y de evaluación de su consecución.
- Una eficiente estructura de conexión entre el primer y el segundo curso del Ciclo Formativo.

2 Requisitos del proyecto

Se debe realizar un diseño de la solución, documentando esta fase en al menos:

- Modelo Entidad/Relación
- Diagrama de clases.
- Diseño de pantallas / Casos de uso.

En el diseño de la base de datos se solicita como requisito mínimo:

- Al menos dos relaciones 1:N o una relación N:M, con un mínimo de tres tablas relacionadas.
- Implementación en MySQL / MariaDB.
- Configuración de la conexión a la base de datos almacenado en XML (se valora uso de JAXB).

En el diseño de clases se solicita:

- Empleo de interfaces.
- Empleo de clases abstractas.
- Sobrecarga y sobreescritura de métodos.
- Herencia optimizada.
- Desarrollo de DAOS, de forma eficaz, implementando aproximaciones lazy y eager.
- Empleo de colecciones adecuadas a cada situación.

El desarrollo se realizará en el IDE que más convenga con los siguientes requisitos:

- JDK > 8.

- Emplear Maven para la estructura del proyecto.
- Emplear Github para control de repositorios.
- Emplear JavaFX para la interfaz gráfica.

En las funcionalidades básicas se solicita:

- Carga de conexión y control de la misma.
- CRUD con base de datos, además de las operaciones básicas.
- Control de impedancia Relación - Objeto.

En cuanto a la solución y temática, se valora:

- Utilidad real de la aplicación.
- Diseño intuitivo de la interfaz.
- Edición avanzada de datos: multimedia...
- Uso de stream y expresiones lambda.
- Log de errores en ficheros.
- Herramientas de copias de seguridad.
- Uso de datos avanzados: fechas, expresiones regulares...

3 Desarrollo

El desarrollo del proyecto se hará de forma individual empleando el entorno de desarrollo antes indicado. Todos los contenidos y manuales necesarios se encuentran en la plataforma del módulo (Moodle). Periódicamente, se realizarán revisiones para aclarar dudas y controlar el proceso de desarrollo.

Es importante fijar un calendario de desarrollo y cumplir los sprints.

Es vital emplear sistemas de copias de seguridad y control de versiones.

El desarrollo se realizará pensando en un entorno real de producción.

Ha de buscarse la calidad en el código, su reutilización, su optimización y, por supuesto, SU DOCUMENTACIÓN.

El código y los comentarios han de desarrollarse en idioma inglés.
(Recomendable, pero no obligatorio)

No se pueden utilizar asistentes de inteligencia artificial para el desarrollo del código.

Tal como hemos visto, debemos cuidar el diseño y cumplir el patrón MVC.

La asistencia a clase es obligatoria durante el desarrollo. (durante las 3 semanas de prácticas, el alumnado que esté en la empresa, no tiene obligación)

¿Por dónde empezar?

Una vez hemos diseñado el modelo (base de datos y diagrama de clases) y la lógica de negocio. Una vez hemos diseñado las pantallas (JavaFX con SceneBuilder), podemos comenzar implementando el modelo y la conexión con BBDD.

A partir de aquí ir conectando el modelo con la vista (FXML) mediante sus controladores.

4 Temporalización

Cada desarrollador podrá establecer sus sprints de forma particular, pero existen unas fechas que debemos cumplir sí o sí.

- Inicio del proyecto: última semana de abril.
- Sprint intermedio: revisión del desarrollo. Debe estar documentado el diseño, implementado el modelo y diseñado las vistas.
- **Entrega final: miércoles 21 de mayo.**
- **Presentaciones: a partir del miércoles 22 de mayo.**

Todo proyecto que no esté entregado ANTES de la fecha límite establecida quedará rechazado y, por tanto, anulado.

5 Entrega y defensa

La entrega se realizará en la tarea que estará habilitada en la Moodle. Y debe realizarse ANTES de la fecha límite establecida.

La entrega consistirá en un **ÚNICO documento PDF** (adecuadamente maquetado) con los siguientes contenidos mínimos:

- Descripción: explicación del proyecto. En qué consiste la solución elegida.
- Fase de diseño: modelo E-R, diagrama de clases, diseño de pantallas, archivo SQL que explicita el diseño de la base de datos, así como el control de integridad y consistencia (Índices, claves, integridad referencial, restricciones...)
- Fase de desarrollo: temporalización, explicación de los puntos fuertes del diseño, **URL del repositorio Github que contiene el código fuente** (comprobar que está público).
- Lista de comprobación: incluye, cómo incluye tu proyecto los siguientes requisitos:
 1. Cómo se usa la herencia, las restricciones con abstract y la llamada a constructores de la clase padre.

2. Qué relaciones se utilizan, 1:N o N:M. o ambas.
 3. Cómo has construido los DAO, quién tiene el control en las relaciones N:M.
 4. Cómo funciona el mostrar, insertar y eliminar.
 5. Cómo funciona el actualizar.
- Fase de presentación: URL al JAR (en Github) para poder probarlo.
 - Fase de defensa: Exposición en clase a partir de la fecha establecida, por un tiempo no superior a 10 minutos, donde se presente la solución, puedes emplear los recursos audiovisuales que veas oportunos, se comentarán los aspectos a destacar y se realizará la demo: duración aprox. 10'.