

PROYECTO MULTIMODULAR APLICACIÓN JAVA CON BD Y EDS

El proyecto consiste en diseñar y construir una aplicación de gestión que utilice, mediante JDBC, una base de datos en MySQL.

El programa debe hacer uso del lenguaje de programación Java junto con sentencias SQL y almacenar toda la información en una base de datos relacional en MySQL.

Para desarrollar la aplicación se pide:

1. Escribir en lenguaje natural el sistema a modelar. Esta descripción debe ser lo más detallada posible del objetivo y principales funcionalidades (requisitos) del proyecto.

Respecto a BD:

2. **Diseño conceptual.** A partir de la descripción anterior, construir el modelo Entidad/Relación extendido. Donde aparecen el nombre de todas las entidades, relaciones y atributos; así como todas las cardinalidades.

3. **Diseño Lógico.** Utilizar la documentación generada en el punto anterior para obtener el modelo relacional normalizado hasta tercera forma normal. Es necesario especificar las claves foráneas y a que tablas pertenecen, claves alternativas,...
En el caso, que el modelo relacional sea demasiado simple: con un número menor o igual a 3 tablas, es necesario volver a realizar el primer punto y ampliar el sistema.

4. **Diseño Físico.** Script con la Creación de la base de datos, directamente en MySQL, tomando como referencia la documentación obtenida en el punto anterior.
Script con la Carga de datos individual o masiva en las tablas generadas.

5. **Script con las operaciones de consulta y manipulación sobre la base de datos** creada (no tienen que ser los requeridos en el CRUD ya que se valorará la dificultad de las mismas):

- 8 Sentencias multitablas (5 consultas, 2 actualizaciones y 1 borrado): incluir Join, Subconsultas, Consultas que utilicen funciones de agregados y agrupación por campos...
- 2 Vistas. Las consultas más complejas se convertirán en vistas.
- 3 Procedimientos almacenados y/o funciones.
- 2 Disparadores (1 Before y 1 After)

Respecto a Java:

6. Codificar en **Java con JDBC**, una aplicación que sirva para gestionar el sistema: altas, bajas, modificaciones, control de acceso, informes, búsquedas, etc.

NOTA: Todo lo que es necesario incluir en la aplicación debe tener un uso lógico, con respecto a la aplicación que se está construyendo, y representar cierta dificultad.

Por ejemplo, no se puede usar una vista para realizar la consulta: `SELECT * FROM Tabla.`

7. En la aplicación hay que incluir lógica que demuestre la adquisición de los aprendizajes elementales del módulo:

- Conceptos fundamentales de Programación
- Concepto de clases, objetos, métodos, atributos.
- Relación entre clases.
- Estructuras de gestión de objetos.
- Manejo de excepciones.
- Clases abstractas e interfaces.
- Estructuras dinámicas: Colecciones, Maps.
- Interfaces de la API y Stream.
- Bases de datos Conexión con JDBC.
- Realización de operaciones CRUD.

Respecto a Entorno de desarrollo:

8. Creación del Diagrama de clases UML.
9. Uso de Git/GitHub para el control de versiones.
10. Generación de la documentación técnica con Javadoc.
11. Generación de la documentación sobre los casos de prueba.
12. Generación de algún test unitario de prueba con JUnit.
13. Creación de la documentación funcional: Manual del usuario.

El proyecto se realizará de forma individual. El desarrollo del mismo se hará en clase de forma guiada por el profesor y se complementará en casa. **Ningún proyecto que no haya sido supervisada su ejecución en clase o parcialmente, en el caso de los alumnos de prácticas, será admitido.**

La entrega del proyecto debe incluir:

- Descripción del sistema (Objetivo general y principales funcionalidades).
- Diseño conceptual: Modelo Entidad/Relación extendido.
- Diseño lógico: Modelo Relacional.
- Diseño físico: Script SQL en el que se creen las tablas con las constraints adecuadas.
- Script SQL para la carga de datos. Será más fácil probar la aplicación con datos que desde cero. Se pueden realizar la carga masiva de datos a tablas usando herramientas de tipo kaggle o Generate Data.
- Script SQL con las consultas, vistas, procedimientos, funciones y disparadores.
- Diagrama de clases UML
- Proyecto Java, con el código fuente de la aplicación.
- Repositorio GitHub del proyecto bien estructurado.
- Manual técnico. Generación con Javadoc de la documentación técnica del proyecto.

- Pruebas unitarias. Documentación de los casos de uso y demostración del uso de JUnit para la generación de test unitarios de pruebas.
 - Tabla con la definición de los casos de prueba
 - Test unitarios en JUnit
- Manual de usuario. Un conjunto de explicaciones para que una persona que no conoce la aplicación sea capaz de utilizarla.

SOBRE LAS ENTREGAS

Recuerdo la **obligatoriedad de usar GitHub** para todo lo que no sea documentación.

Toda la documentación estará en formato electrónico, incluidos los diagramas, así como el enlace al repositorio de GitHub. Se entregará en la plataforma moodle.

Calendario Entregas:

- Entrega BASE del Proyecto con fecha límite el domingo 27 de Abril a las 23:59
- Entrega FINAL del proyecto con fecha límite el domingo 25 de Mayo a las 23:59
- La **defensa**, será en principio el **27 de Mayo, en horario de clase**, donde se demostrará el funcionamiento y la autoría del mismo.

ANOTACIÓN COMPLEMENTARIA

Posteriormente, durante el mes de Junio, los alumnos que quieran subir nota podrán añadir a su proyecto una interfaz visual con Java FX.

INFORMACIÓN COMPLEMENTARIA: Sugerencias de posibles temáticas del proyecto

En el mundo empresarial, social, familiar, etc. se utilizan las bases de datos de forma directa o a través de las aplicaciones en las diferentes plataformas de acceso a las mismas en los diferentes sectores y gestiones realizadas con las mismas. Por ello el modelo o parte de un modelo a poder realizar de:

Sectores económicos	Primario	Agricultura, minería, ganadería, pesca, reciclaje, etc.
	Secundario	Calzado, electrónica, muebles, plásticos, alimentos procesados, ropa, automóviles, maquinaria, tecnología, etc.
	Terciario	Públicos o Privados: Educación, salud, banca, transporte, comunicaciones, entretenimiento, comercio, servicios legales, viajes, etc.
	Cuaternario	Es una subdivisión del sector terciario. Labores intelectuales o la economía del conocimiento: Investigación, desarrollo, tecnologías de la información, consultorías, planificación financiera, etc.
	Quinario	Es una subdivisión del sector terciario. Actividades de creación, reordenación e interpretación de ideas y proyectos. Uso de datos y nueva tecnología para la toma de decisiones. (Toma de decisiones de los directivos de empresas y ONG, labores oficiales de gobierno, etc.)

De los sectores anteriores podemos tener modelos de base de datos relacional para la gestión de:

Aeropuertos, Distribución, Ventas, Pedidos, Almacenes, Bancos, Clubs deportivos, Transporte, Farmacéuticas, Laboratorios, Clínicas médicas, Vacunación, Redes Sociales, Blogs, Periodismo, Financieras, Agencias de viajes, Seguros, Gimnasios, Energía (Eléctricas, Combustibles, etc.), almacenes de datos, software, hardware, plataformas tecnológicas, servicios de internet, etc.