

PRACTICA
M03 – PROGRAMACIÓN
UF 6 POO. INTRODUCCIÓN A LA PERSISTENCIA EN BASES DE DATOS
FECHA LÍMITE DE ENTREGA: Práctica entregada antes de la fecha 18/01/2018 23:55h -> Nota máxima 10 Práctica entregada antes de la fecha 01/02/2018 23:55h -> Nota máxima 7 Práctica entregada a partir de 02/02/2018 y hasta 13/05/2018 23:55h -> Nota máxima 5

## OBJETIVOS

Crear un proyecto JAVA SE 8 donde se apliquen los conceptos vistos en la asignatura y especialmente en la UF6: acceso a bases de datos, diagramas UML, definición de clases, utilización avanzada de clases, aplicación de estructuras de almacenamiento, control de excepciones, lectura/escritura de información, interfaces gráficas de usuario.

El proyecto se desarrollará por fases:



- **DEFINICIÓN:** Establecer los requisitos funcionales de la aplicación y casos de uso
- **MODELADO:** Definir E-R Diagram, Definir UML Use Case Diagram, UML Activity Diagram, UML Class Diagram, sketches de las distintas pantallas de la aplicación (si aplica)
- **IMPLEMENTACIÓN:** Crear base de datos. Programar la aplicación en JAVA
- **TESTING:** Realizar las pruebas necesarias para comprobar que la aplicación funciona correctamente y cumple requisitos. Se registrarán los detalles de las pruebas realizadas (funcionalidad bajo test, valores de entrada, valores de salida esperados, etc) en la memoria.
- **DOCUMENTACIÓN:** Redactar memoria

## REQUISITOS MÍNIMOS

A continuación se explican los requisitos mínimos que debe cumplir el proyecto:

- El proyecto se desarrollará individualmente o en equipo de máximo 2 personas
- La aplicación se denominará M03\_UF5\_Practica\_[NApellido]\_[NApellido]
- JDK 8
- La aplicación Java tendrá sus datos fuente en una base de datos relacional. Se deberá entregar en formato pdf o imagen el diagrama E-R que muestre la estructura de la base de datos.
- La base de datos relacional tendrá más de una tabla y debe disponer de al menos:
  - Una relación 1:1
  - Una relación 1:N
- El diseño de la base de datos debe definir las restricciones necesarias para mantener la integridad de los datos (no duplicidades, no valores null cuando el dato es obligatorio, etc)
- Utilización de Java APIs: java.lang, java.time, etc.
- Utilización programación avanzada de clases
- Control de excepciones
- Cierre y liberación de recursos de la base de datos y JDBC
- No permitir SQL Injection
- Debe permitir interacción con el usuario
- Debe realizar operaciones de insertar, actualizar, borrar, leer registros de la base de datos.
- Cada clase o interface se definirá en un fichero .java independiente. En la aplicación existirá la clase Main que contendrá el punto de arranque (método public void main... ) del programa.

**La correcta implementación de los requerimientos mínimos citados y su defensa y respuesta a cuestiones técnicas en entrevista individual con profesor es condición necesaria para poder aprobar la práctica.**

## BAREMO DE EVALUACIÓN

- Hasta 5 puntos: implementar correctamente los requerimientos mínimos y saber defender su desarrollo y cuestiones técnicas en entrevista individual con profesor.
- Hasta 1 punto por la creatividad de la app y su aplicación a un entorno real y complejidad de la base de datos (múltiples tablas, relaciones M:N).
- Hasta 1 punto diagramas UML.
- Hasta 1 punto por la calidad del código, su correcta estructura y documentación. Se valorará:
  - la correcta división en packages/clases/interfaces
  - uso de Design Patterns: Singleton, DAO, Factory, MVC (si aplica)
  - código limpio
  - programación solid: lo más atómica posible para permitir reaprovechamiento
  - Comentarios de código
- Hasta 1 punto por Memoria explicativa con:
  - Portada: Nombre de la aplicación (icono si aplica), nombre y apellidos de los participantes en el proyecto
  - Project overview: describe en qué consiste la aplicación, cuáles son sus casos de uso. Incluye imágenes de demo de su funcionamiento.
  - Project organization: describe cómo se ha organizado el equipo para desarrollar la aplicación. Indica qué tareas ha realizado cada uno, cuál ha sido el calendario.
  - Project design: explica el diseño utilizado apoyándote en los diagramas UML. Estructura del proyecto en NetBeans (packages, clases, etc)
  - Project testing: indica qué funcionalidades has testeado de la aplicación. De cada prueba indica: valores de entrada, valores de salida esperados, valores de salida obtenidos y resultado (PASS/FAIL)
  - Project extras: si has decidido incorporar extras a tu proyecto, indica cuales y explícalos brevemente.
  - Conclusiones: qué has aprendido con el desarrollo de este proyecto
- Hasta 1 punto por extras a los requerimientos mínimos previamente validados con el profesor:
  - Uso de transacciones
  - Uso de herramientas de mapeo como Hibernate
  - Uso de interfaz gráfica de usuario
  - Uso de sistemas de control de versiones
  - Aplicación de conceptos de programación de Java no vistos en el aula: concurrencia, localización, etc
  - Creación de un video “promocional” que muestre las funcionalidades e interacciones más interesantes de la aplicación.
  - ....

### **¿QUÉ DEBES ENTREGAR?**

- Memoria en pdf
- Diagramas UML en pdf o formato imagen: E-R Diagram, UML Use Case Diagram, UML Activity Diagram, UML Class Diagram
- Código fuente Java Application Project