

# (P) Programación orientada a objetos con acceso a base de datos

## Producto 2. Implementación de clases en Java

### Descripción

A partir de los diagramas desarrollados en el producto anterior, realizaremos un programa en Java que implemente estos diagramas almacenando la información en estructuras dinámicas de datos y ficheros XML aplicando patrones de diseño.

### Objetivo

El objetivo principal de la actividad es:

- Implementar estructuras dinámicas de datos y el modelo estático de clases en lenguaje Java, aplicar el patrón de diseño DAO, almacenar datos en estructuras XML y realizar pruebas unitarias utilizando *JUnit*, todo ello utilizando un sistema de control de versiones.

### Pasos a seguir

Los pasos a seguir para llevar a cabo el producto son:

1. Leer detenidamente estas instrucciones e identificar los requerimientos de la actividad.
2. Revisar detenidamente la rúbrica de evaluación.
3. Consultar los recursos necesarios facilitados en el aula.
4. Registrar y configurar GitLab JE UOC.
  - a. En el desarrollo del proyecto deberá utilizarse el sistema de control de versiones Git, en este caso se usará la herramienta online GitLab.
5. Implementar el modelo estático de clases en lenguaje Java.
  - a. En la implementación deberán utilizarse como mínimo las Clases Genéricas o Java Generics.
  - b. Elegir el Tipo de Colecciones en Java óptimo para cada caso en función de la necesidad o requisito a implementar.

- c. Se debe tener en cuenta la correcta gestión de Excepciones, así como implementar Excepciones personalizadas.
- 6. Realizar un programa en Java en modo de consola que almacene la información en estructuras dinámicas de datos y la guarde en ficheros XML.
  - a. Hay que tener en cuenta que en esta actividad del proyecto los datos se almacenarán en ficheros XML pero que posteriormente en este proyecto los mismos datos se almacenarán en una base de datos relacional.
  - b. Para separar la persistencia de los datos del resto de funcionalidades del sistema se deberá utilizar el patrón de diseño DAO (Data Access Object). Uno de los beneficios de DAO es la independencia del almacén de datos, así pues, el cambio de usar ficheros XML a una base de datos relacional o el cambio de motor de base de datos solo afectará al DAO y no a las clases encargadas de la lógica de negocio o de presentación. o el cambio de motor de base de datos.
  - c. Con el objeto de conseguir la independencia del almacén de datos se usará el patrón Factory para instanciar los DAOs.
  - d. Se debe tener en cuenta que estos ficheros XML se utilizarán posteriormente para la carga inicial de datos en la base de datos, así pues, su estructura debe ser adecuada para que sea posible su importación en el modelo relacional que se implantará en la base de datos MySQL.
- 4. Realizar pruebas unitarias utilizando el entorno JUnit. Elegir dos métodos que apliquen lógica de negocio y realizar las pruebas unitarias con *JUnit* para evaluar su correcto funcionamiento.

## Se requiere

Los requisitos indispensables para realizar el producto son:

- 1. Registrarse en la herramienta online GitLab *JE UOC*.
- 2. Disponer del entorno de programación en Java.
- 3. Disponer del entorno *JUnit*.
- 4. Investigar otras fuentes de información para ampliar los recursos facilitados en el aula virtual.

## Recursos

Para llevar a cabo la actividad puedes apoyarte en los materiales y fuentes de información disponibles en el espacio “Recursos” del aula virtual.



X



JESUÏTES  
educació

fp.uoc.edu

Formación  
profesional

## Criterios de evaluación

La puntuación máxima del producto es un 10.

La puntuación mínima para superarlo es de 5 sobre 10.

## Rúbrica de evaluación

La rúbrica de evaluación de la actividad es la siguiente:

Criterios	4	3	2	1
Control de versiones con GitHub	Implementa el control de versiones usando GitLab correctamente y de forma optimizada.	Implementa el control de versiones usando GitLab correctamente.	Implementa el control de versiones usando GitLab con algunos errores.	Presenta dificultades en su conjunto en la implementación del control de versiones usando GitHub.
Generación de código a partir de un diagrama de clases	Genera el código adecuado a partir de un diagrama de clases de forma optimizada.	Genera el código adecuado a partir de un diagrama de clases.	Genera código a partir de un diagrama de clases con algún error.	Presenta dificultades en su conjunto en la generación del código a partir de un diagrama de clases.
Estructuras dinámicas de datos	Implementa las Estructuras dinámicas de datos correctamente y de forma optimizada.	Implementa las Estructuras dinámicas de datos correctamente.	Implementa las Estructuras dinámicas de datos con algún error.	Presenta dificultades globales en implementación de las Estructuras dinámicas de datos.
Colecciones en Java	Implementa las colecciones en Java óptimas en función de los requerimientos correctamente y de forma optimizada.	Implementa las colecciones en Java óptimas en función de los requerimientos correctamente.	Implementa las colecciones en Java óptimas en función de los requerimientos con algún error.	Presenta dificultades en su conjunto en la implementación de las colecciones en Java óptimas en función de los requerimientos.
Patrón de diseño DAO	Implementa el patrón de diseño DAO correctamente y de forma optimizada.	Implementa el patrón de diseño DAO correctamente.	Implementa el patrón de diseño DAO con algún error.	Presenta dificultades globales en la implementación del patrón de diseño DAO.
Almacenar datos en	Almacena datos en estructuras XML	Almacena datos en estructuras	Almacena datos en estructuras	Presenta dificultades globales para

estructuras XML	correctamente y de forma optimizada.	XML correctamente.	XML con algún error.	almacenar datos en estructuras que XML
Pruebas unitarias utilizando JUnit	Realiza pruebas unitarias utilizando JUnit correctamente y de forma optimizada.	Realiza pruebas unitarias utilizando JUnit correctamente.	Realiza pruebas unitarias utilizando JUnit con algún error.	Presenta dificultades globales para realizar pruebas unitarias utilizando JUnit.

## Indicaciones para la entrega de la actividad

Para la entrega de la actividad deberás:

- Hacerlo por el canal indicado en el aula virtual.
- Formato: zip o rar, en el contenido se deberá incluir el documento PDF con la resolución de los ejercicios y los ficheros del programa en java así como el resto de ficheros relacionados en este producto.
- Nombre del archivo: producto2(P5)\_Apellido1Apellido2\_Nombre
- Extensión recomendada: 5-7 páginas.