
ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
DISEÑO DE SOFTWARE
TALLER DE PATRONES CREACIONALES Y UML

Objetivos Específicos

1. Aplicar patrones de diseño creacionales en la elaboración de diagramas utilizando el lenguaje de modelado unificado para la resolución de los problemas propuestos.
2. Diseñar un producto de software aplicando principios de diseño orientado a objetos para que sea robusto, mantenible y escalable.

Resultado de Aprendizaje

1. Habilidad para aplicar teoría de ciencias computacionales y fundamentos de desarrollo de software para producir soluciones basadas en computación.

Descripción

En equipos de trabajo, conformados por hasta cuatro estudiantes, identifique patrones de diseño aplicables dentro del escenario que se presenta a continuación y elabore un diseño utilizando diagrama de clases y secuencia en UML e implemente el código mínimo java correspondiente a los patrones identificados. Justifique su respuesta y modele solo la parte relevante a los patrones de diseño.

Especificaciones

Considere el sistema para administrar los productos de **una tienda que comercializa tanto artículos físicos como productos digitales**. Permite registrar categorías, crear nuevos productos con sus atributos específicos, consultar información de inventario y precios, y almacenar toda la información en una base de datos central. El sistema busca mantener la coherencia de los datos y ofrecer una estructura flexible que permita añadir nuevos tipos de productos sin modificar el resto del código.

Para asegurar que todas las operaciones trabajen sobre la misma fuente de información, se define un componente encargado de gestionar la conexión con la base de datos y atender todas las consultas y actualizaciones. Dicho componente garantiza que solo exista una instancia activa en todo el sistema, evitando duplicación de recursos y posibles conflictos de acceso a los datos.

Por otro lado, cuando se necesita crear un producto, el sistema emplea un mecanismo que decide dinámicamente qué tipo de objeto construir según el tipo solicitado. Así, si el producto es físico, incluirá peso y dimensiones; si es digital, contendrá tamaño y formato. Este mecanismo facilita la extensión del sistema para soportar nuevos tipos de productos, manteniendo el código organizado y fácil de mantener.

Sección A

Elabore un **reporte** en el que **identifique los patrones aplicables al sistema**. Para cada patrón: objetivo, motivación, consecuencias y relación con SOLID. Declare supuestos. Mantenga el foco en dos piezas: acceso a datos y creación de productos (físico, digital, extensibles). **[30%]**

Sección B

Entregue diagramas que muestren cómo operan los patrones elegidos:

- Un diagrama de clases del sistema que muestre como está compuesto cada patrón en **paquete separado**¹, con interfaces, clases abstractas y concretas, herencias, visibilidad y multiplicidades.
- Diagramas de secuencias necesarios para evidenciar el funcionamiento de cada patrón; su objetivo es ilustrar el mecanismo del patrón. **[40%]**

¹ Paquete en UML: Debe investigar cómo se representan los paquetes en un diagrama de clases y separar de esta manera las clases de su propuesta.

Sección C

Implemente exclusivamente el código relativo a los patrones identificados:

- Componente único de acceso a datos para toda la aplicación.
- Punto de creación de productos que decida el subtipo en tiempo de ejecución (p. ej., físico con peso/dimensiones; digital con tamaño/formato). Debe permitir agregar nuevos tipos de productos sin cambiar el código existente.

[30%]

Entregables (en un archivo ZIP o RAR)

1. Un **documento (disponible en formatos docx o pdf)** que contenga el reporte y las imágenes correspondientes a los diagramas solicitados. Además, el documento debe tener una carátula en la que se liste a los integrantes del equipo, una tabla de contenido e indicar la herramienta utilizada para modelar los diagramas (por ejemplo, Visual Paradigm). Cada patrón debe estar en una página y la imagen debe tener buena resolución.
2. El código mínimo Java implementando los patrones identificados.