

NOMBRE DE LA ASIGNATURA	Patrones De Diseño de Software						
NOMBRE DE LA ACTIVIDAD	Patrones de diseño creacionales						
TIPO DE ACTIVIDAD	Sincrónica		Asincrónica	x	Individual	x	Grupal
TEMÁTICA REQUERIDA PARA LA ACTIVIDAD			OBJETIVOS				
Unidad 1. Introducción a UML y diseño orientado a objetos Unidad 2. Introducción a los patrones de diseño Unidad 3. Patrones creacionales			Evaluar la capacidad para modelar una solución de software a una problemática planteada, mediante la filosofía del diseño orientado objetos, utilizando UML. Implementar una solución de software utilizando incorporando patrones de diseño creacionales.				
COMPETENCIAS			INSUMOS PARA EL DESARROLLO DE LA ACTIVIDAD / REFERENCIAS BIBLIOGRÁFICAS				
<ul style="list-style-type: none"> Identificación de escenarios de uso de patrones de diseño creacional Identificación de componentes de patrones de diseño creacional Implementación de patrones de diseño creacional 			<ul style="list-style-type: none"> Material educativo y material complementario de la asignatura “Unidad 1, Unidad2 , Unidad 3” Fuentes bibliográficas del módulo. 				
CONOCIMIENTOS PREVIOS REQUERIDOS							
Conceptos fundamentales de POO y UML – Patrones de diseño creacionales							
ESPECIFICACIONES DE LA ACTIVIDAD							
Requerimientos de la evaluación: <ul style="list-style-type: none"> Diseño de diagrama de clases (obligatorio) Implementación de código (obligatorio) 							
1. Problema planteado: Tienda en línea "CustomYourStyle"							
La tienda tiene problemas para gestionar pedidos personalizados de manera eficiente. El equipo de desarrollo debe diseñar un sistema que cumpla con los siguientes requerimientos:							
a. Productos básicos: Ropa: camisetas, sudaderas, gorras. Accesorios: Cadenas, pulseras, anillos. Calzado: tenis, casuales, mocasines							
b. Opciones de personalización: Ropa: color, tela (nombre, material principal, % de uso), talla (S, M, L, XL), marca (nombre, país). Accesorios: diseñador (nombre, apellido), fecha elaboración, material (nombre, natural/sintetico/reciclado).							

Calzado: capellada (material sintético o textil), suela (antideslizante, con tacón, caucho), cierre (broche, corredera, cordón)

c. Flujo esperado:

El cliente selecciona un producto básico.

Personaliza los atributos del producto usando un configurador.

Puede guardar la configuración para crear variaciones rápidamente.

El sistema centralizado almacena todos los pedidos, asigna un ID único y gestiona su estado (en proceso, procesado, enviado, entregado)

Problemas Actuales

- El proceso de creación de pedidos es manual, lo que introduce errores humanos.
- No existe un sistema eficiente para clonar productos personalizados.
- No hay control centralizado de los pedidos realizados, lo que genera problemas la gestión de sus estados y sus identificadores únicos.

Patrones a Implementar

- **Factory Method:**
Gestionará la creación de productos básicos (camisetas, tazas, cuadros). Este patrón abstraerá el proceso de instanciación para cada categoría de producto.
- **Builder:**
Permitirá construir productos personalizados paso a paso, agregando atributos como color, material o tamaño.
- **Prototype:**
Implementará la clonación de productos ya configurados, permitiendo generar copias rápidas con pequeñas modificaciones.
- **Singleton:**
Garantizará que exista un único administrador de pedidos, centralizando el control de identificadores únicos y almacenando los pedidos en un repositorio compartido.

Interacciones Clave:

1. El cliente crea un pedido
2. El cliente decide agregar un producto básico a través del Factory Method, personalizarlo con el Builder, o a partir de un catálogo de productos preconfigurados clonándolo con el prototype.
3. El Administrador de Pedidos (Singleton) almacena y gestiona los pedidos.

**RECOMENDACIONES /
OBSERVACIONES**

Para el diseño UML del Diagrama de clases se sugiere utilizar cualquiera de las siguientes herramientas: StartUML, PlantUML, Draw.io, GenMyModel o Visual Paradigm.

Puede utilizar el lenguaje de programación de su preferencia, preferiblemente Java, es opcional el desarrollo de interfaces Gráficas de Usuario.