

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA

FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

SEDE BOCA DEL MONTE

Programación II

CAT. ING. Luis Alvarado



Primer Proyecto

Jonathan Joel Chan Cuellar 7690-22-1805

# **Documentación del Programa de Fabricación y Registro de Ventas de sillas y muebles**

## **Introducción**

Este programa en Java tiene como objetivo simular un sistema de fabricación y venta de muebles, donde se pueden crear sillas y mesas personalizadas, registrar ventas de estos productos y mostrar detalles de las ventas realizadas. El programa se basa en el patrón de diseño de fábrica abstracta para la creación de muebles.

## **Resumen del Programa**

Este programa Java permite a los usuarios crear sillas y mesas personalizadas, registrar ventas y mostrar información detallada de las ventas. Utiliza el patrón de diseño Abstract Factory para crear instancias de sillas y mesas de acuerdo con las especificaciones del usuario.

## **Cómo Usar el Programa**

- Ejecuta el programa y sigue las instrucciones en la consola para crear sillas y mesas personalizadas y registrar ventas.
- Proporciona detalles como estilo, color, material, precio, fecha de entrega, nombre del cliente, NIT y número de venta para cada producto.
- El programa creará las instancias de sillas y mesas, mostrará un mensaje de confirmación y registrará la venta.
- Al final, se mostrará la información detallada de todas las ventas registradas.
- También preguntará la cantidad de madera y metal que usa la fábrica para hacer las sillas y mesas

## **Patrón de Diseño Abstract Factory**

Este programa utiliza el patrón de diseño Abstract Factory para permitir la creación de productos (sillas y mesas) sin especificar sus clases concretas. Las fábricas

concretas (ChairFactory y TableFactory) implementan la interfaz FurnitureFactory y proporcionan la lógica para crear productos específicos.

## **Ejemplo de Uso**

El programa es un ejemplo básico de cómo se pueden aplicar los conceptos de la orientación a objetos y el patrón de diseño en Java para crear y gestionar objetos de manera eficiente.

## **Requisitos del Sistema**

El programa está escrito en Java y requiere una máquina virtual de Java (JVM) para ejecutarse. Asegúrate de tener Java instalado en tu sistema antes de ejecutar el programa.

Este documento proporciona una descripción general del programa y su funcionalidad. Para detalles específicos sobre la implementación de las clases y los métodos, puedes consultar el código fuente proporcionado.

¡Disfruta explorando y aprendiendo más sobre la programación orientada a objetos y los patrones de diseño con este programa de fabricación y registro de ventas de muebles!

## **Método main en la clase Fabrica**

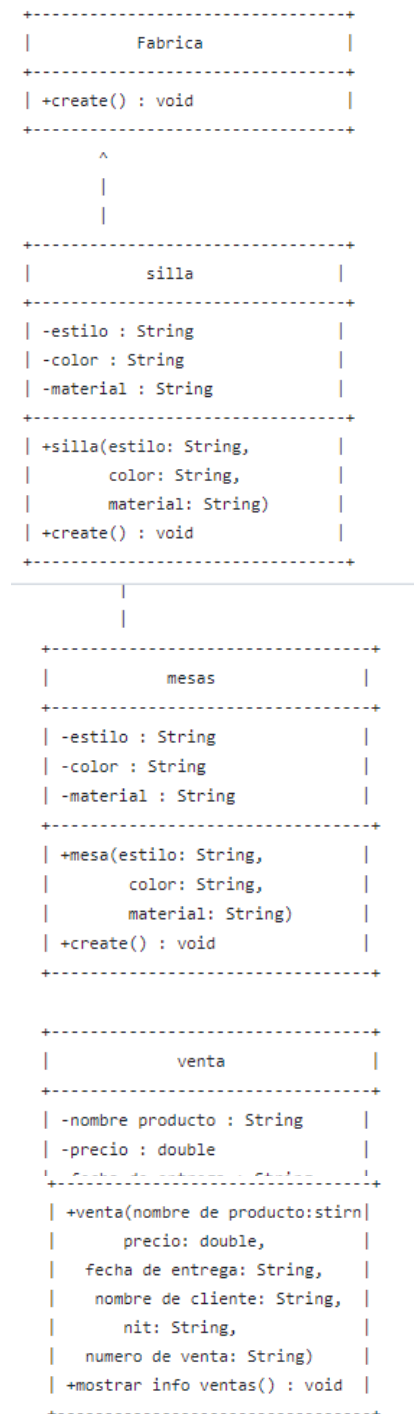
El método main en la clase Fabrica es el punto de entrada del programa. Permite al usuario ingresar información para crear sillas y mesas personalizadas, así como detalles de ventas. Utiliza instancias de las fábricas concretas ChairFactory y TableFactory para crear los objetos de muebles correspondientes y luego registra las ventas en una lista.

## **Conclusiones**

Este programa ilustra el uso del patrón de diseño de fábrica abstracta para crear objetos de diferentes tipos (sillas y mesas en este caso) de manera uniforme.

También muestra cómo registrar y mostrar detalles de las ventas. Puede ser útil como base para un sistema de gestión de ventas de muebles más completo.

## Diagrama UML



```

+-----+
|   fabrica de muebles   |
+-----+
| +crear muebles(estilo: String, |
|               color: String, |
|               material: String)|
+-----+
|
|
|
+-----+
|   silla fabrica   |
+-----+
| +crear muebles (estilo: String, |
|               color: String, |
|               material: String)|
+-----+
|
|
|
+-----+
|   mesa fabrica   |
+-----+
| +crearmuebles(estilo : String, |
|               color: String, |
|               material: String)|
+-----+

```

```

+-----+
|   Material   |
+-----+
| - nombre: String |
| - cantidad: double |
+-----+
| + Material(nombre: String) |
| + setCantidad(cantidad: double): void |
| + getCantidad(): double |
| + getNombre(): String |
+-----+
| + main(args: String[]): void |
+-----+

```

## Interfaz en modo texto

- Crear una silla:
  - Estilo: [Ingrese el estilo de la silla]
  - Color: [Ingrese el color de la silla]
  - Material: [Ingrese el material de la silla]
  - Precio: [Ingrese el precio de la silla]
  - Fecha de Entrega: [Ingrese la fecha de entrega de la silla]
  - Nombre del Cliente: [Ingrese el nombre del cliente]
  - NIT: [Ingrese el NIT del cliente]
  - Número de Venta: [Ingrese el número de venta]
- 
- [Se mostrará un mensaje indicando que se ha creado una silla con los detalles ingresados]
- 
- Crear una mesa:
  - Estilo: [Ingrese el estilo de la mesa]
  - Color: [Ingrese el color de la mesa]
  - Material: [Ingrese el material de la mesa]
  - Precio: [Ingrese el precio de la mesa]
  - Fecha de Entrega: [Ingrese la fecha de entrega de la mesa]
  - Nombre del Cliente: [Ingrese el nombre del cliente]
  - NIT: [Ingrese el NIT del cliente]

- Número de Venta: [Ingrese el número de venta]
- [Se mostrará un mensaje indicando que se ha creado una mesa con los detalles ingresados]
- Detalles de la venta:
  - Producto: [Nombre del producto (Silla o Mesa)]
  - Precio: [Precio del producto]
  - Fecha de Entrega: [Fecha de entrega del producto]
  - Nombre del Cliente: [Nombre del cliente]
  - NIT: [NIT del cliente]
  - Número de Venta: [Número de venta]
- [Se mostrarán los detalles de todas las ventas registradas]

1. El programa comienza importando las clases necesarias, como Scanner y List, y define una interfaz llamada Furniture que tiene un método create().
2. Luego, se definen dos clases concretas: Chair (Silla) y Table (Mesa). Estas clases implementan la interfaz Furniture y tienen propiedades como estilo, color y material. El método create() en estas clases muestra información sobre la creación de sillas y mesas.
3. La clase Sale representa una venta y almacena información sobre el producto vendido, el precio, la fecha de entrega, el nombre del cliente, el NIT y el número de venta. También tiene un método displaySaleInfo () para mostrar los detalles de la venta.

4. Se define una clase abstracta llamada FurnitureFactory que contiene un método abstracto createFurniture (String style, String color, String material). Las fábricas concretas implementarán este método para crear productos Furniture.
5. Se crean dos fábricas concretas: ChairFactory y TableFactory, que extienden FurnitureFactory. Cada una de estas fábricas implementa el método createFurniture para crear sillas y mesas respectivamente.
6. En el método main, se crea una instancia de Scanner para la entrada del usuario y una lista llamada sales para almacenar información de ventas.
7. El programa comienza solicitando información para crear una silla. Luego, crea una fábrica de sillas (chairFactory), utiliza esta fábrica para crear una silla y registra la información de la venta en un objeto chairSale, que se agrega a la lista de ventas.
8. Después, el programa solicita información para crear una mesa de manera similar y registra la información de la venta en un objeto tableSale, que también se agrega a la lista de ventas.
9. Finalmente, se muestra la información de todas las ventas registradas en la lista.
10. Después le pide que ingrese la cantidad de materiales usados en la fabricación de silla y mesas y le muestra al usuario cuanto material se usó.

.