**Título del Trabajo**  
  
Jarvic Balanta Santacruz  
Universidad Cooperativa de Colombia  
Facultad de Ingeniería  
Patrones de Diseño Orientado a Objetos

# Resumen

Este taller está enfocado en la implementación de los patrones de diseño creacionales Abstract Factory, Builder y Prototype. A través de ejercicios prácticos, aprenderemos a aplicar estos patrones para mejorar la flexibilidad y escalabilidad en la creación de objetos dentro de un sistema de software.

# Introducción

Los patrones de diseño creacionales facilitan la gestión eficiente de la creación de objetos en el desarrollo de software.

* **Abstract Factory**: Permite la creación de familias de objetos relacionados sin acoplarse a implementaciones concretas.
* **Builder**: Facilita la construcción paso a paso de objetos complejos con múltiples configuraciones opcionales.
* **Prototype**: Permite la clonación de objetos para evitar su creación manual repetitiva.

A través de este taller, se analizarán casos de uso reales para entender cómo aplicar estos patrones en distintos escenarios.

# Resultados

Repositorio:

<https://github.com/Jarvicf/ucc.patrones.dise-o.objetos.git>

Ejercicio 1: <https://github.com/Jarvicf/ucc.patrones.dise-o.objetos/tree/7397fc60b953d61f3d1415ba73688a5ec3430f8e/Semana%204%20-%20Repaso%20Abstract%20factoty%20Builder%20Prototype/Ejercicios%20en%20clase/Abstract%20Factory/Abstract%20Factory>

Ejercicio 2: <https://github.com/Jarvicf/ucc.patrones.dise-o.objetos/tree/7397fc60b953d61f3d1415ba73688a5ec3430f8e/Semana%204%20-%20Repaso%20Abstract%20factoty%20Builder%20Prototype/Ejercicios%20en%20clase/Builder/Builder>

Ejercicio 3: <https://github.com/Jarvicf/ucc.patrones.dise-o.objetos/tree/7397fc60b953d61f3d1415ba73688a5ec3430f8e/Semana%204%20-%20Repaso%20Abstract%20factoty%20Builder%20Prototype/Ejercicios%20en%20clase/Prototype/Prototype>

* **Escenario**: Un sistema operativo permite cambiar el tema de su interfaz gráfica (Light Mode y Dark Mode). Cada tema debe proporcionar sus propios botones y ventanas con estilos coherentes

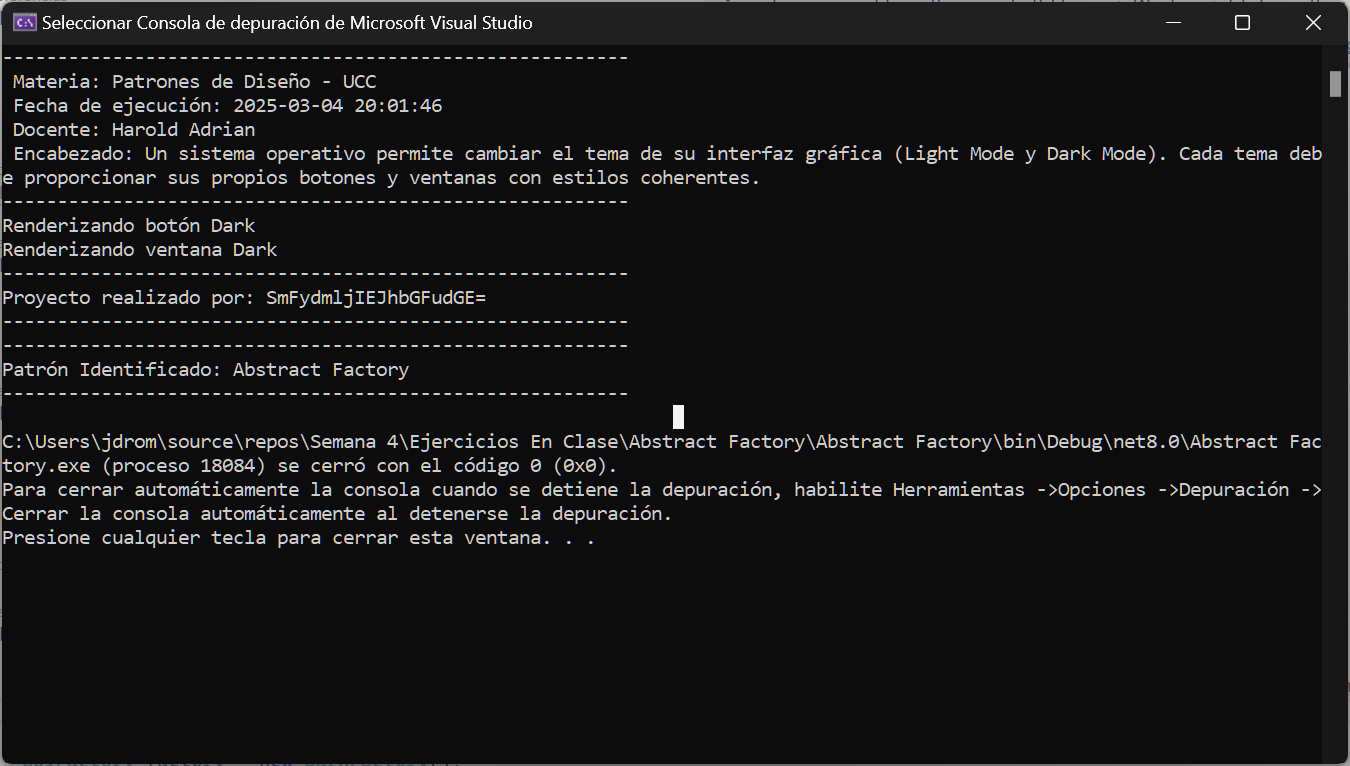


Ilustración . Abstract factory.

* **Escenario**: En una aplicación, los usuarios pueden configurarse con diferentes atributos como nombre, edad, dirección y número de teléfono. No todos los atributos son obligatorios, por lo que es conveniente usar Builder para evitar múltiples constructores.

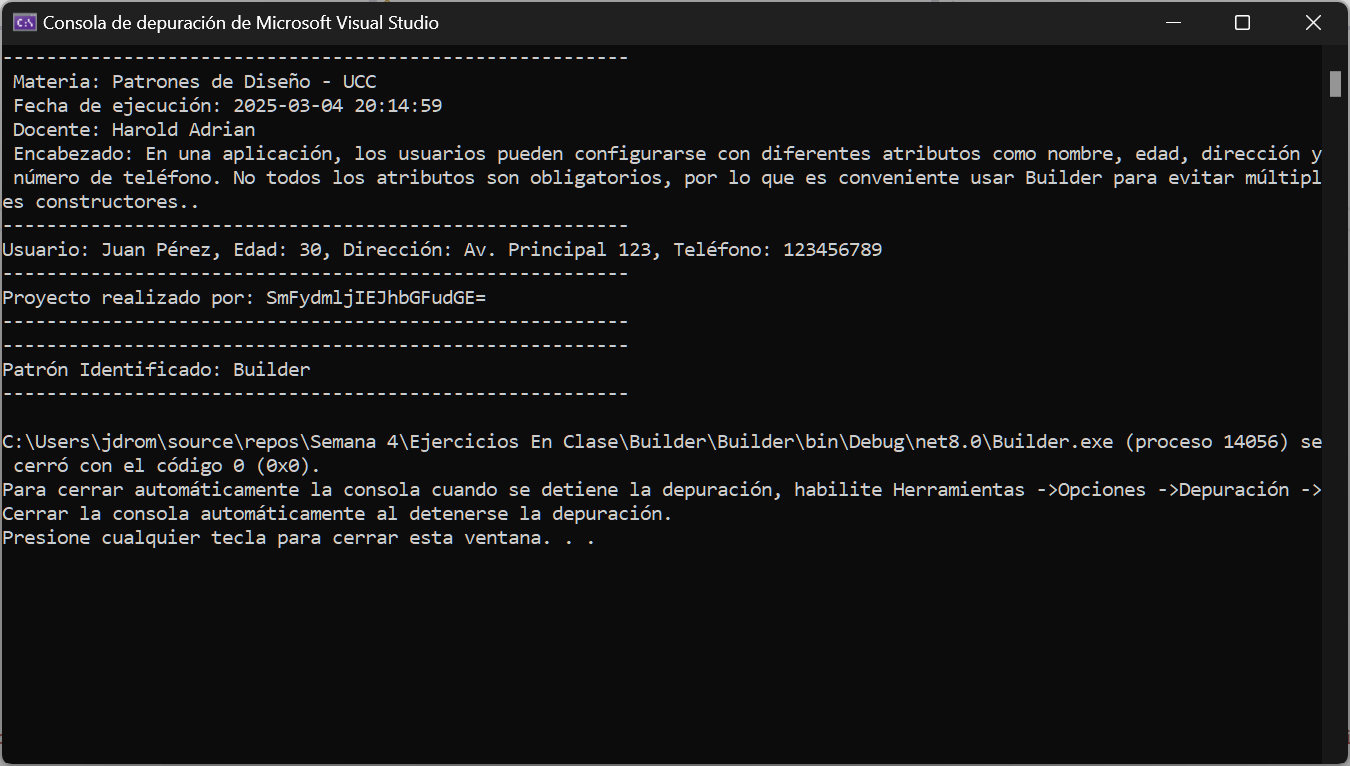


Ilustración . Builder.

* **Escenario**: Un sistema de gestión documental permite a los usuarios crear documentos a partir de una plantilla preexistente en lugar de configurarlos manualmente desde cero.

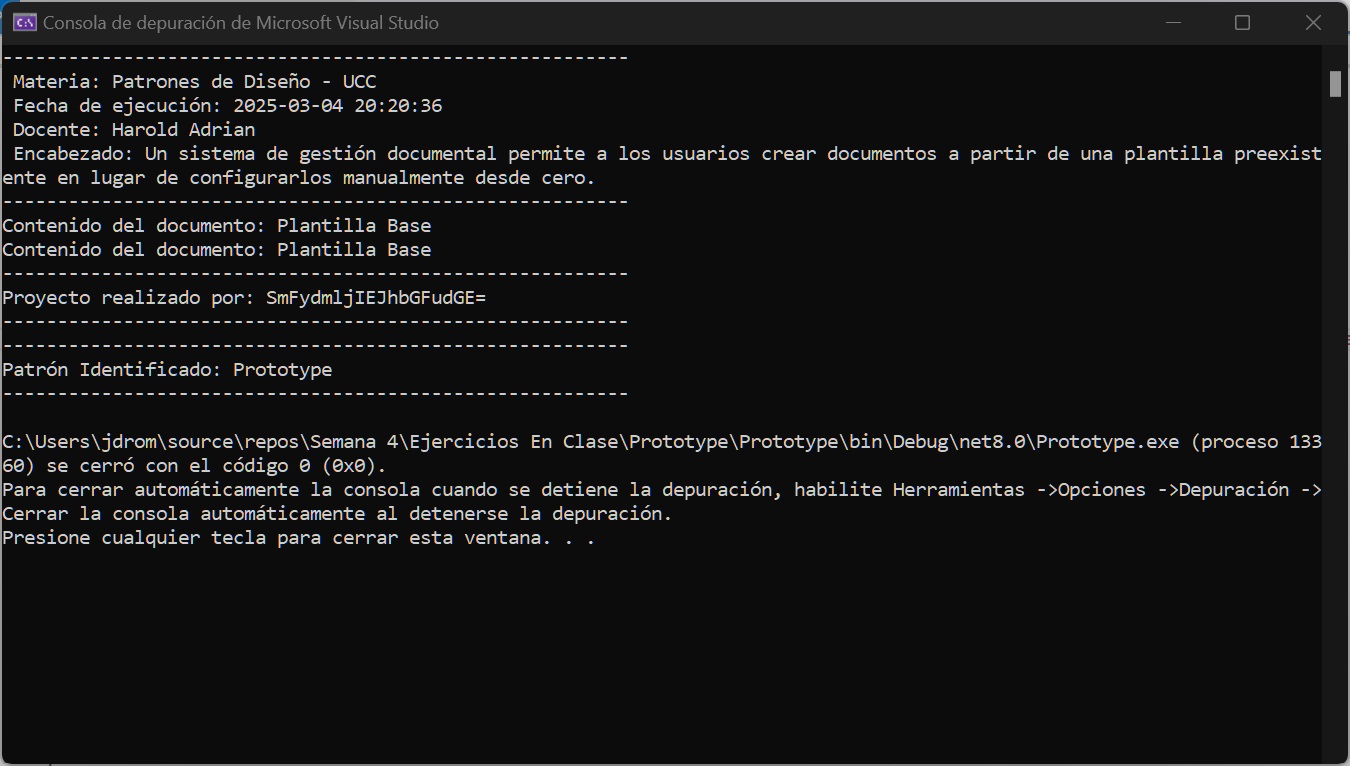


Ilustración . Prototype.

# Conclusiones

Los patrones de diseño creacionales son fundamentales en la estructuración de software modular y escalable.

* **Abstract Factory** facilita la creación de familias de objetos sin acoplarse a implementaciones concretas.
* **Builder** simplifica la construcción de objetos con múltiples configuraciones.
* **Prototype** permite la clonación eficiente de objetos, reduciendo la necesidad de creación manual.

Estos patrones permiten optimizar la mantenibilidad y escalabilidad de las aplicaciones, proporcionando soluciones estructuradas para la creación de objetos.

# Referencias

Bolaños Rodríguez, H. A. (2025). Curso Patrones de Diseño OO - Semana 5. Universidad Cooperativa de Colombia