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

# Resumen

Este ejercicio tiene como objetivo aplicar los patrones de diseño creacionales en el lenguaje C# a través de implementaciones prácticas. Se exploraron patrones como Singleton, Factory Method, Abstract Factory, Builder y Prototype, cada uno enfocado en la gestión eficiente de instancias y la reutilización del código. Los ejemplos demostraron cómo estos patrones pueden mejorar la organización y escalabilidad del software, permitiendo una mayor flexibilidad en la creación de objetos y su administración.

# Introducción

Los patrones de diseño creacionales son fundamentales en el desarrollo de software, ya que proporcionan soluciones estructuradas para la creación de objetos. En este ejercicio, se implementaron estos patrones en C# mediante programas que evidencian su aplicación en distintos escenarios. Se analizaron casos de uso concretos donde cada patrón resulta beneficioso, demostrando cómo la correcta selección de un patrón puede optimizar el código y mejorar su mantenibilidad.

# Resultados

Repositorio [**Aquí**](https://github.com/Jarvicf/ucc.patrones.dise-o.objetos/tree/2db0fa241731aaa6ca200e94077df573203e3144/Semana%202%20-%20Patrones%20Creacionales/Taller%20Patrones)

* Resultados del patrón **Singleton**:

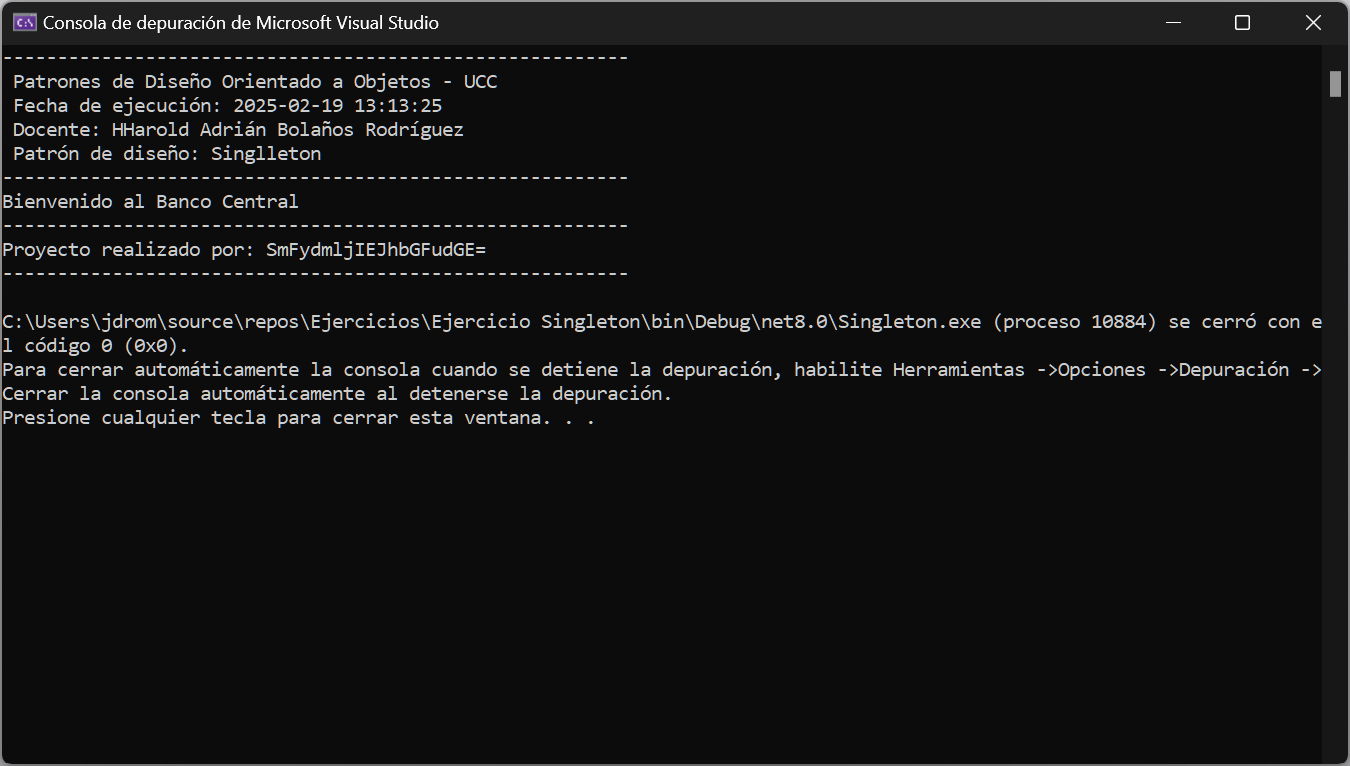


Ilustración 1: Patrón Singleton.

* Resultados del patrón **Factory Method**:

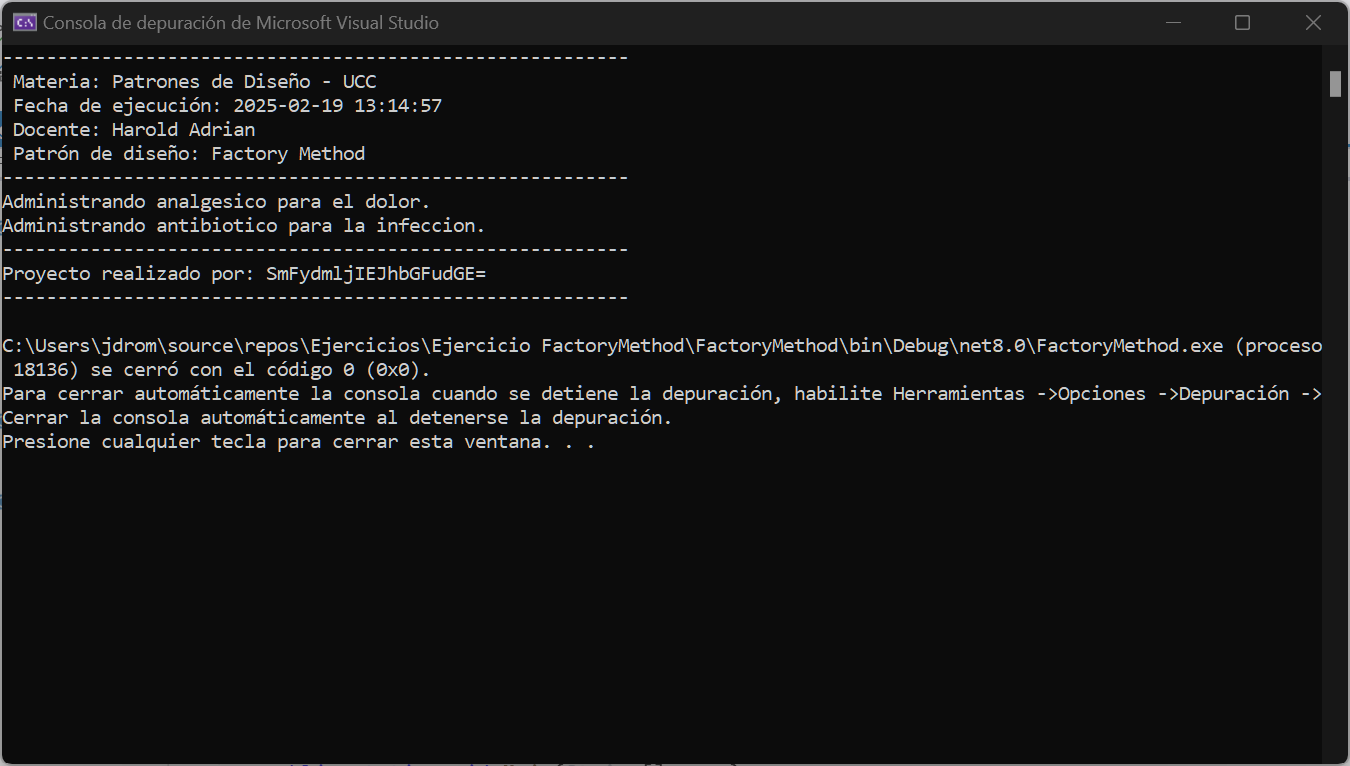


Ilustración 2: Patrón Factory Method.

* Resultados del patrón **Abstract Factory**:

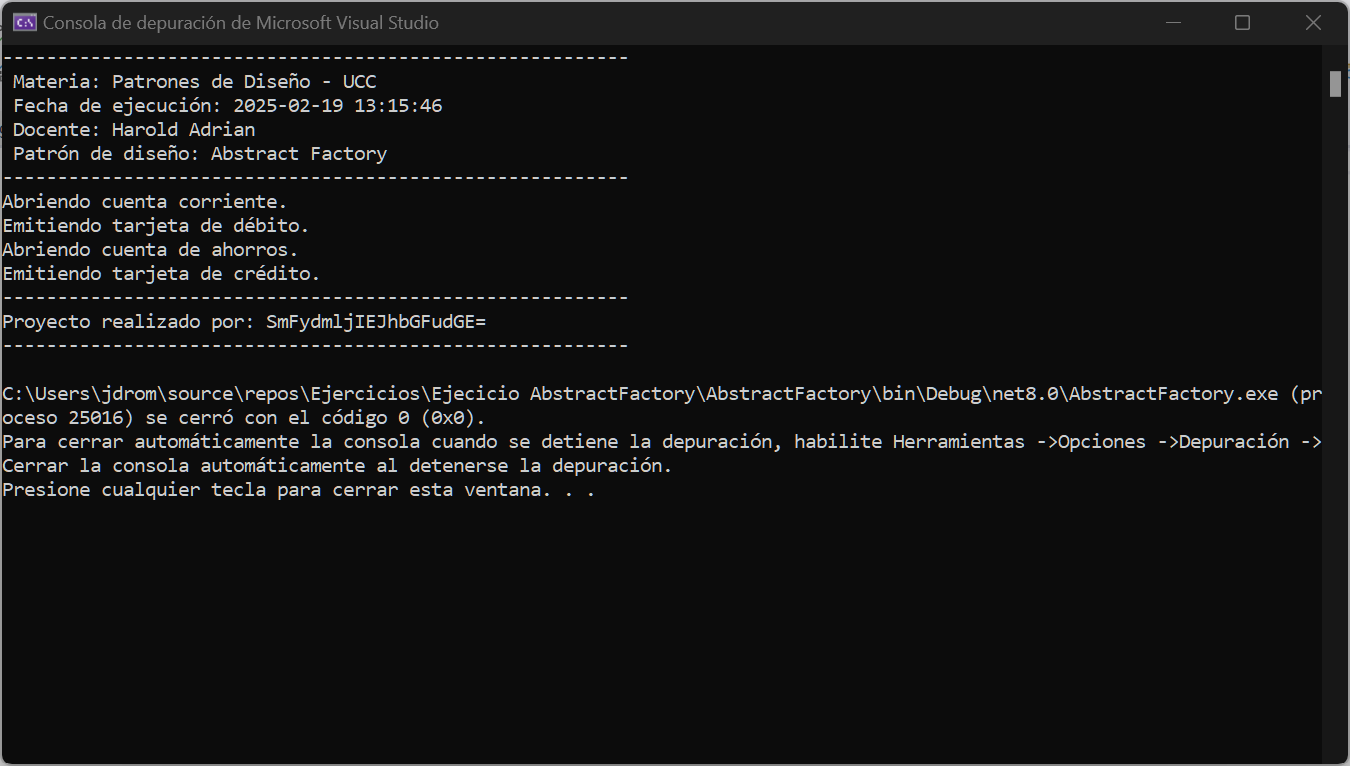


Ilustración 3: Patrón Abstract Factory.

* Resultados del patrón **Builder**:

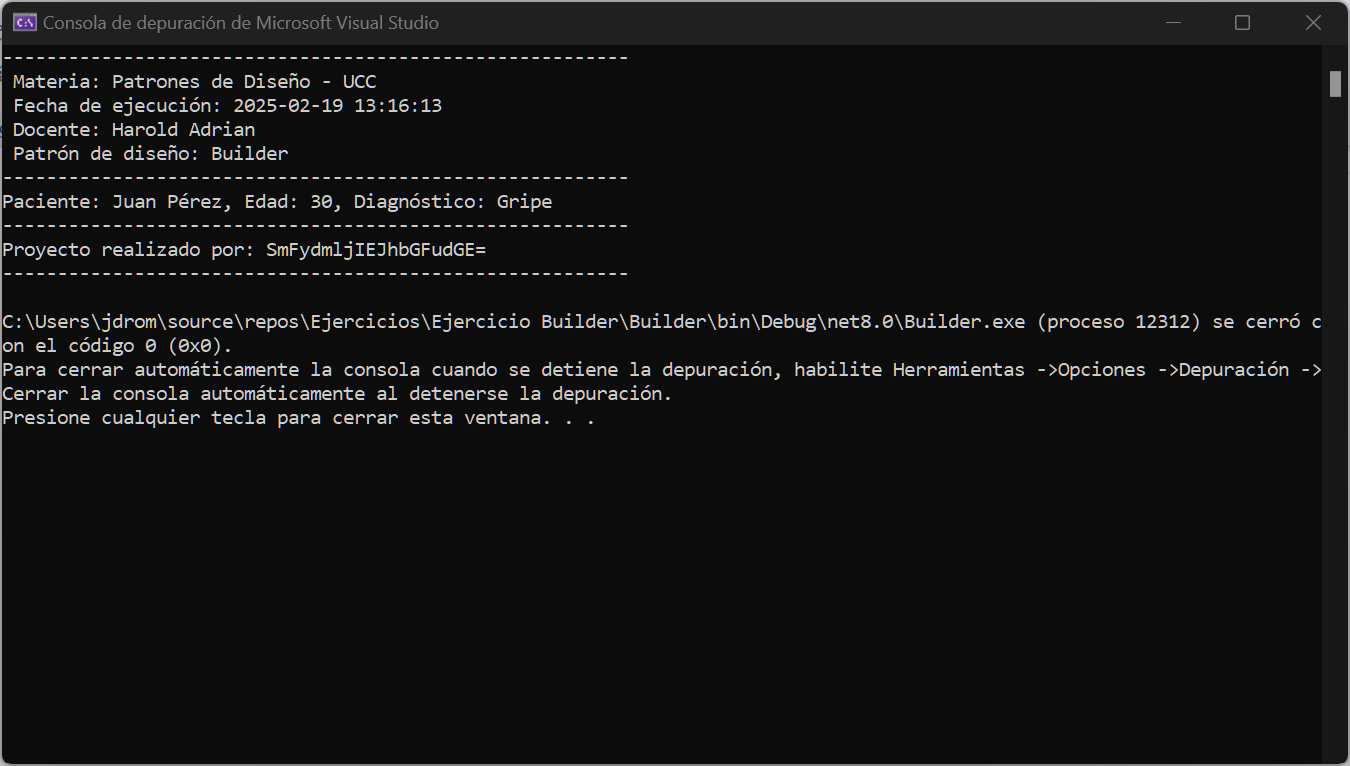


Ilustración 4: Patrón Builder.

* Resultados del patrón **Prototype**:

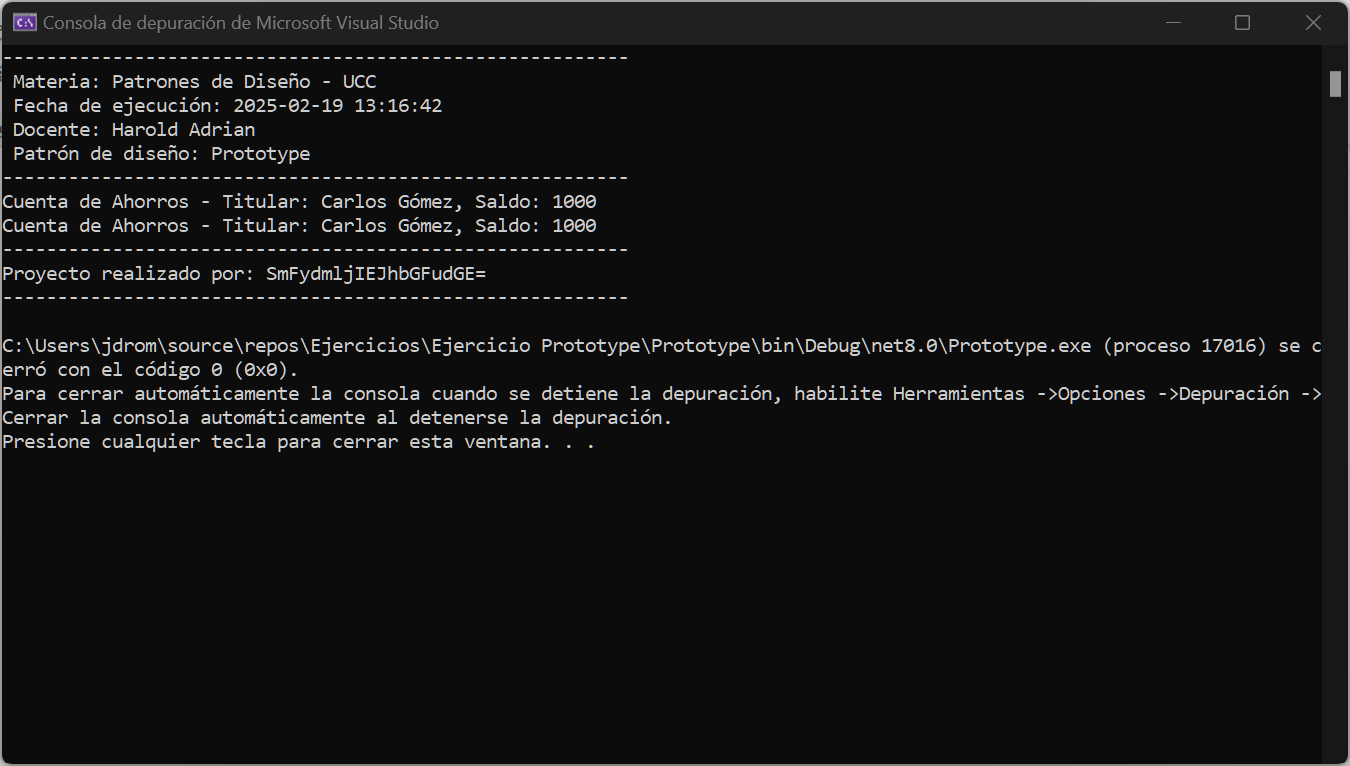


Ilustración 5: Patrón Prototype.

# Conclusiones

La implementación de los patrones de diseño creacionales en C# permitió comprender su impacto en la organización del código y en la optimización de la creación de objetos. Se evidenció que patrones como **Singleton, Factory Method, Abstract Factory, Builder y Prototype** no solo mejoran la reutilización del código, sino que también facilitan la escalabilidad y mantenibilidad de las aplicaciones. A través de los ejercicios, se demostró la importancia de seleccionar el patrón adecuado según el contexto del problema, lo que resulta en un software más estructurado y flexible.

Este ejercicio reforzó la capacidad de aplicar soluciones de diseño eficientes, permitiendo a los desarrolladores mejorar la calidad del código y optimizar el uso de recursos en la construcción de software orientado a objetos.

# Referencias

Bolaños Rodríguez, H. A. (2025). *Curso Patrones de Diseño OO - Semana 2*. Universidad Cooperativa de Colombia.

.