**Ejercicio Patrones Comportamiento**

Jarvic Balanta Santacruz  
Ejercicio Patrones Estructurales: Facade, Proxy y Flyweight  
Facultad de Ingeniería  
Patrones de Diseño Orientado a Objetos

# Resumen

Esta guía explora los **patrones de comportamiento**, los cuales definen cómo interactúan los objetos dentro de un sistema orientado a objetos. Se enfocan en la asignación de responsabilidades, flujo de control y colaboración entre clases, promoviendo principios como bajo acoplamiento, alta cohesión, y cumplimiento de SOLID. La guía incluye los patrones: Strategy, Observer, Command, Chain of Responsibility, State, Mediator, Memento, Template Method y Visitor, con ejemplos y casos de uso prácticos.

# Introducción

Los patrones de comportamiento resuelven problemas relacionados con la dinámica de interacción entre objetos. Estos patrones permiten separar el "qué se hace" del "cómo se hace", desacoplando la lógica de ejecución y facilitando el mantenimiento, la trazabilidad y la extensión del software. Son útiles en entornos donde los procesos cambian con frecuencia, se requiere flexibilidad o múltiples componentes colaboran entre sí.

Se incluyen en esta guía los 9 patrones más representativos del grupo, presentados con su explicación, cuándo usarlos, antipatrones comunes y una historia de usuario con código de ejemplo

# Resultados

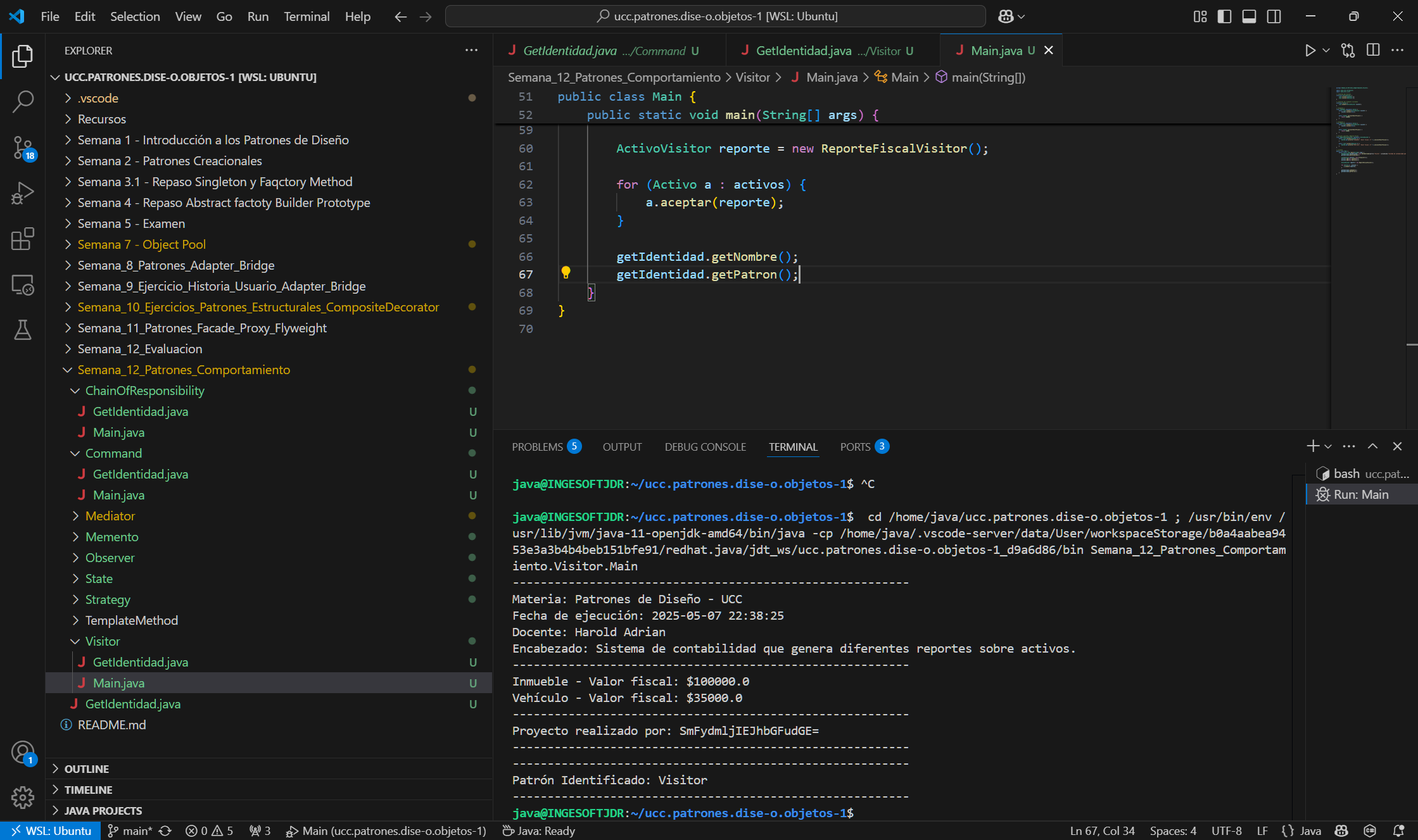


Ilustración 1. Patron Visitor.

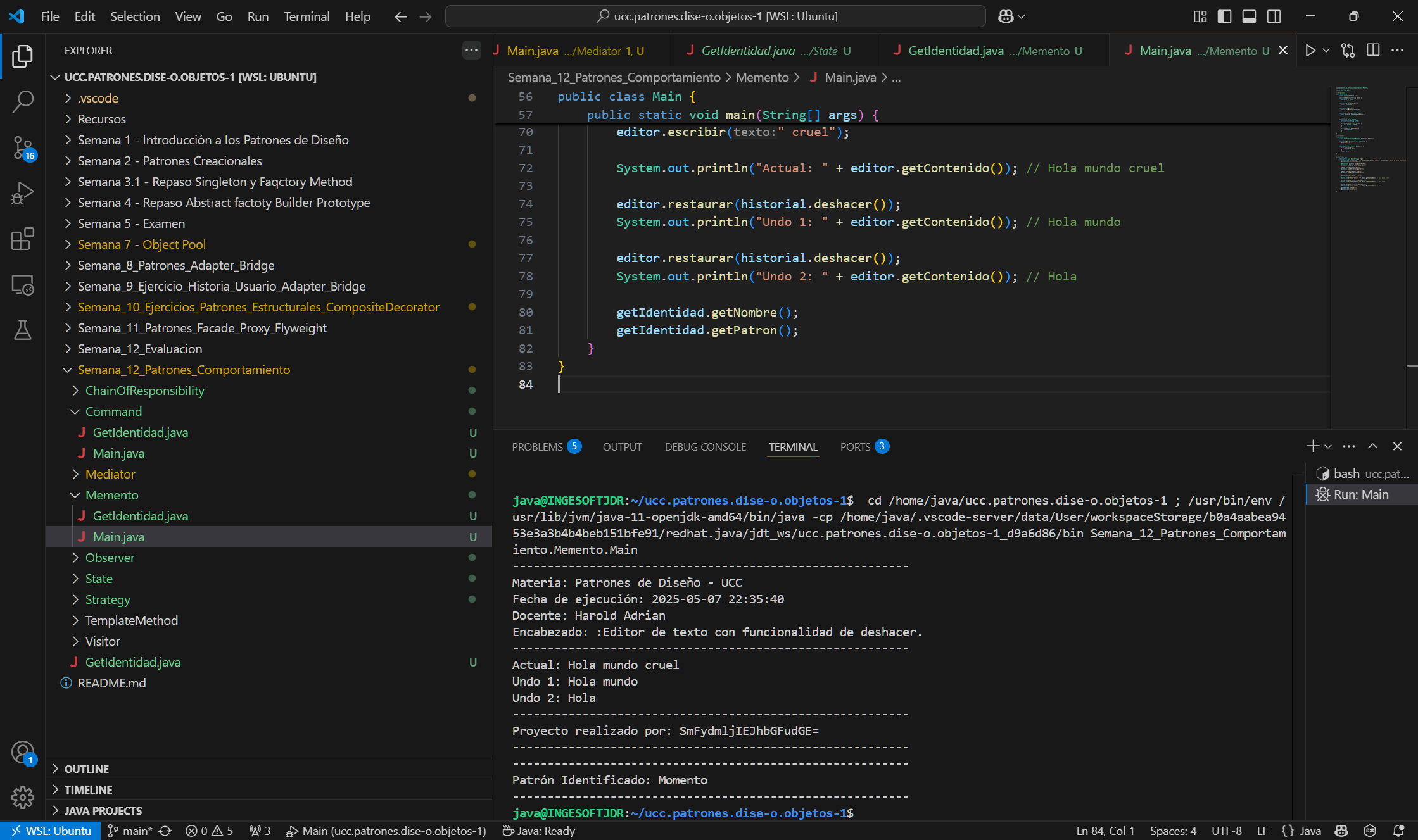


Ilustración 2. Patron Momento.

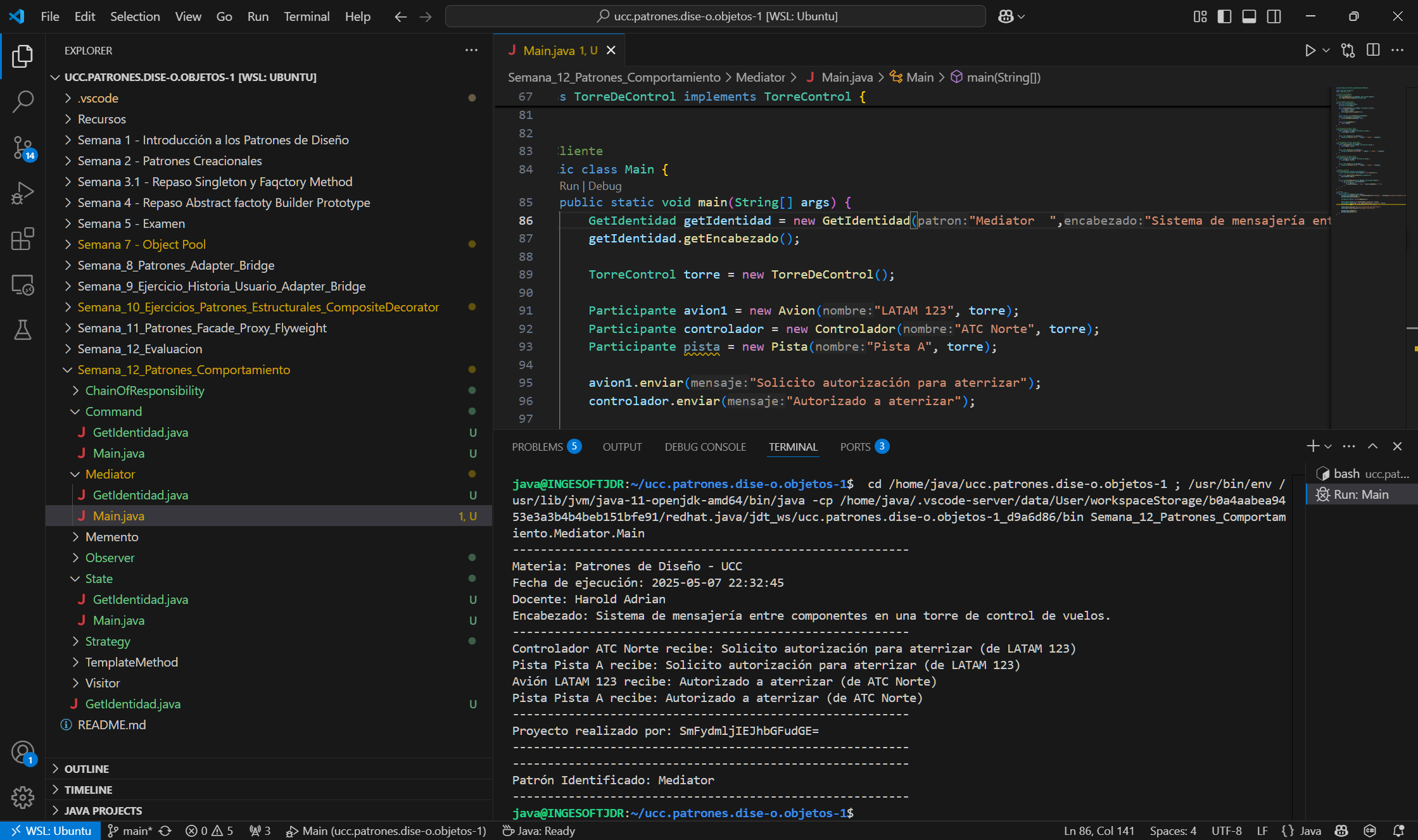


Ilustración 3. Patron Mediador.

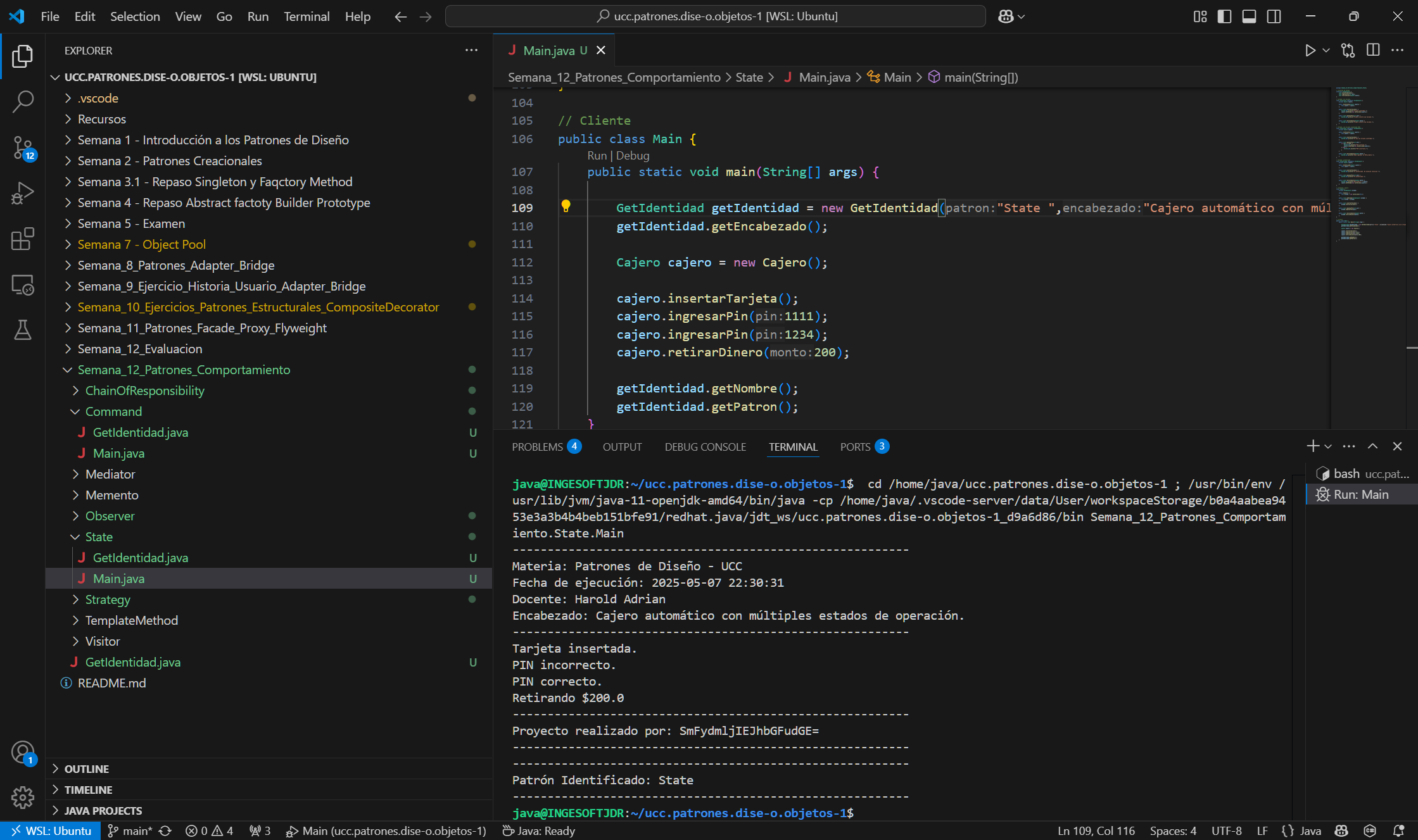


Ilustración 4. Patron State.

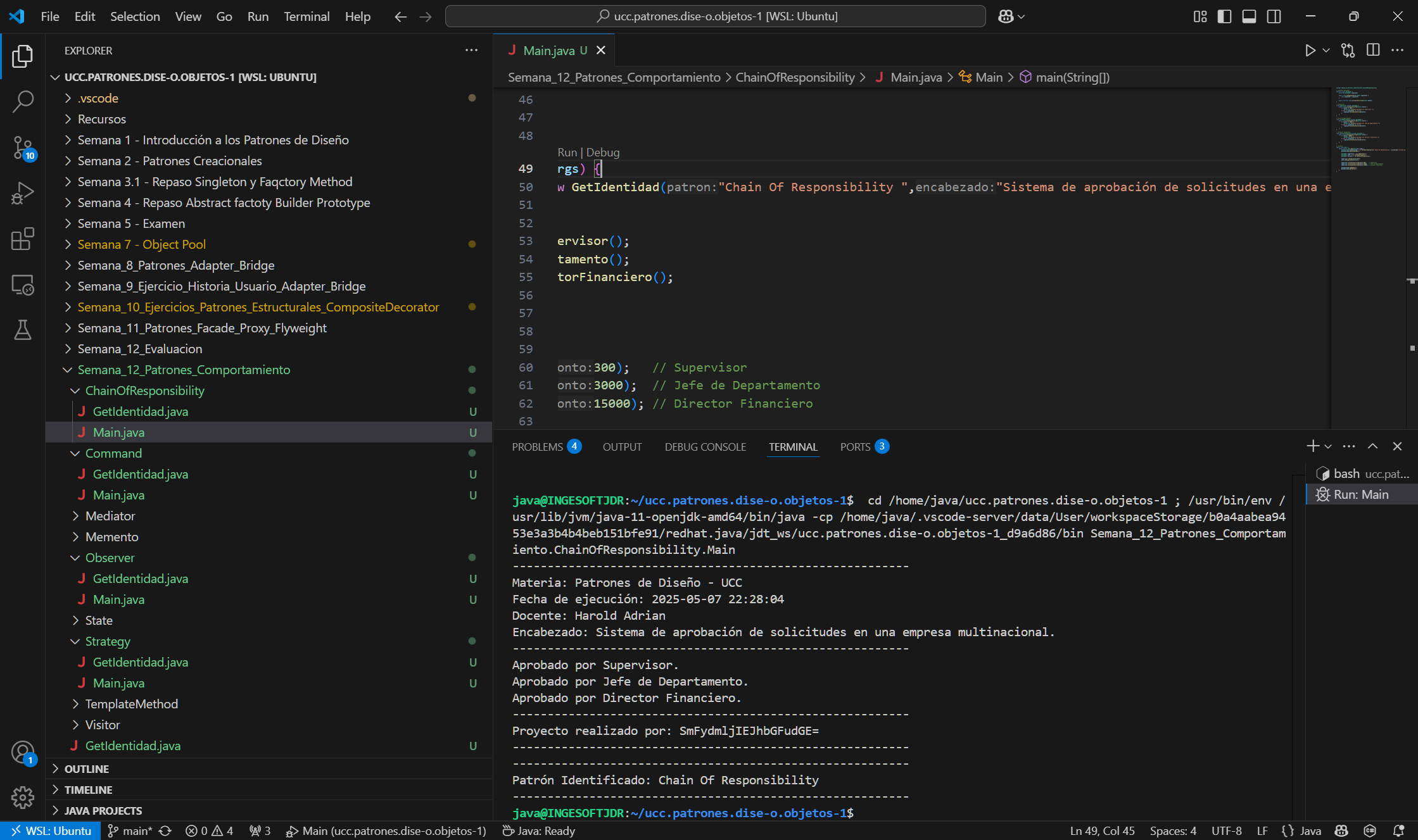


Ilustración 5. Patron Chain of Responsability.

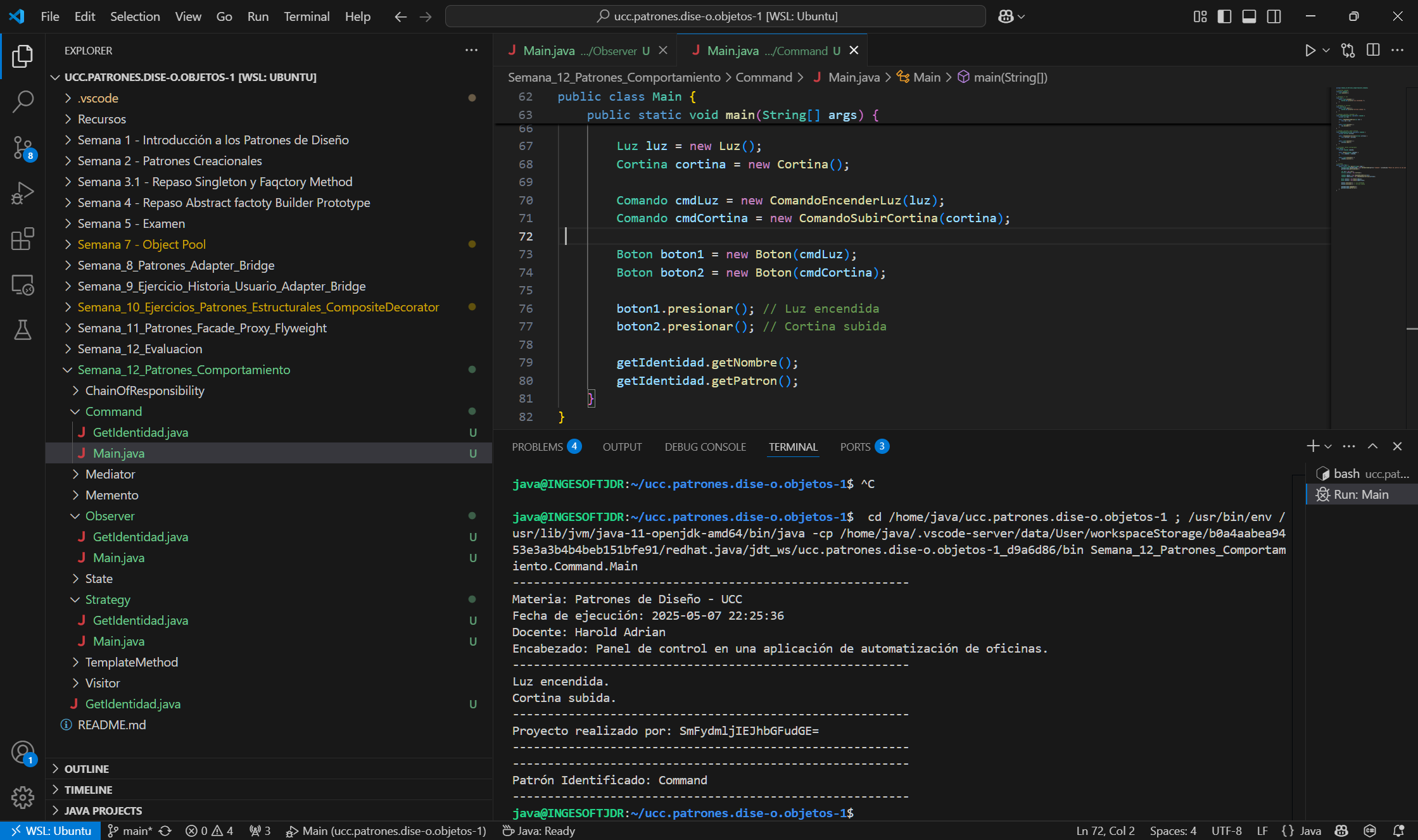


Ilustración 6. Patron Command.

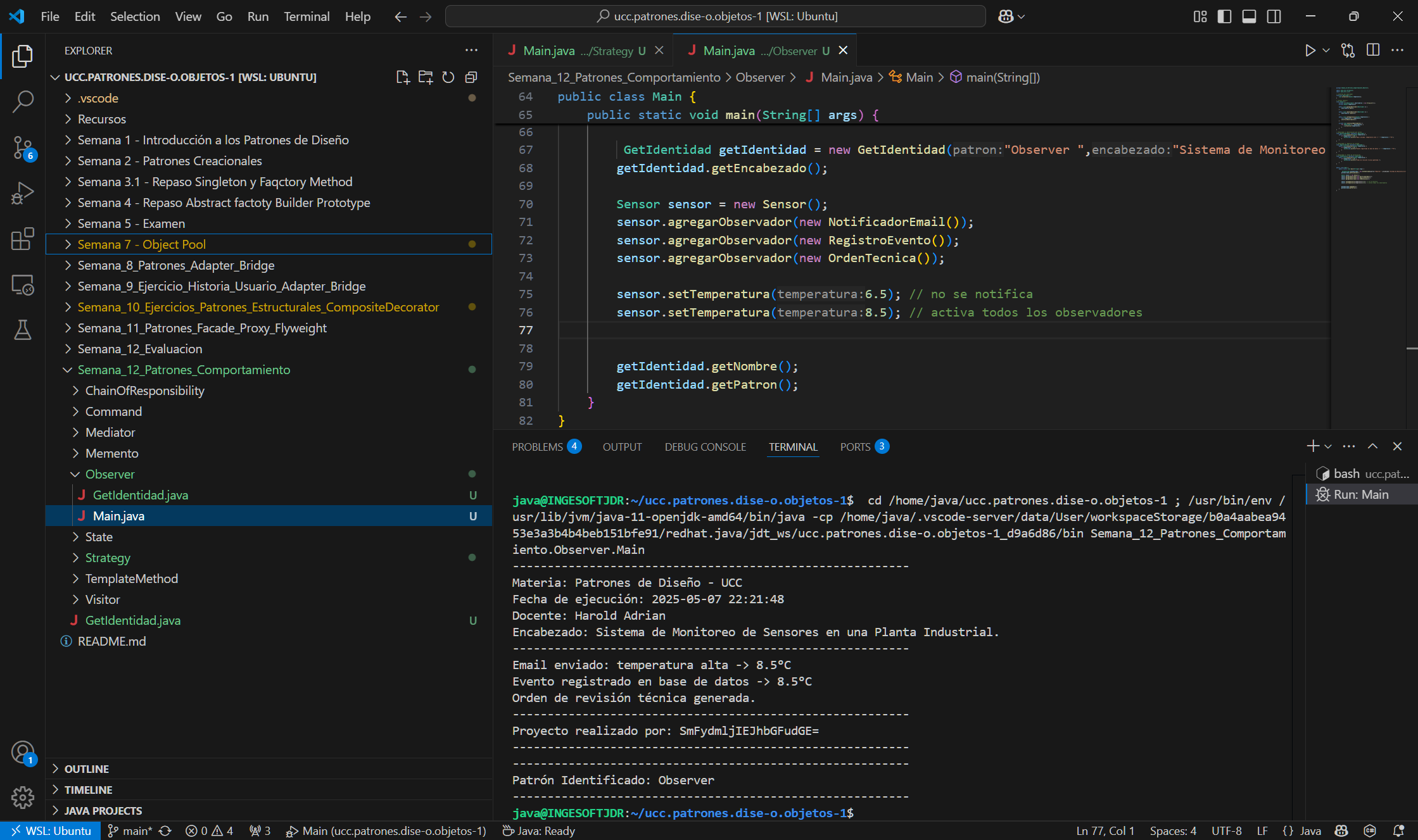


Ilustración 7. Patron Observer.

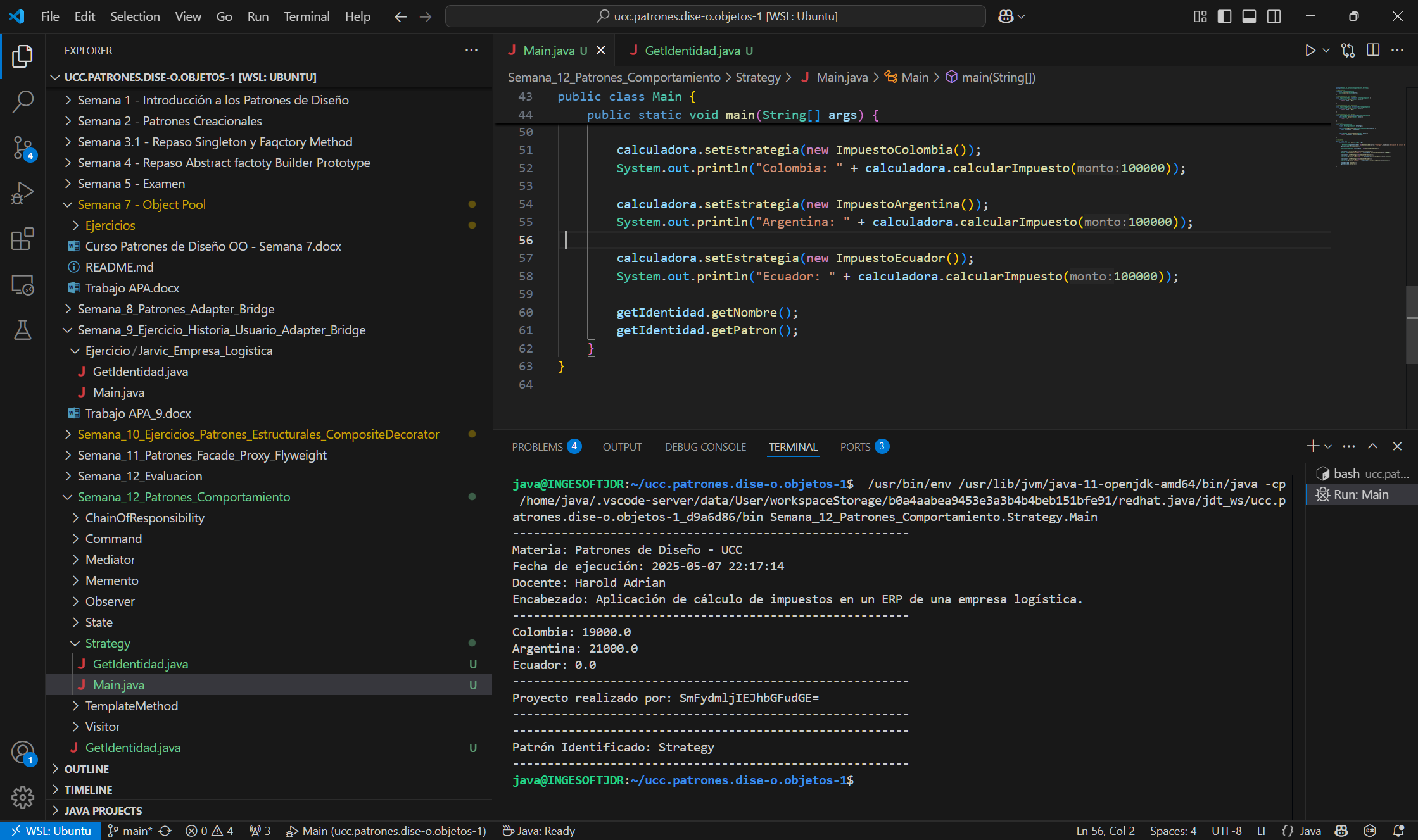


Ilustración 8. Patron Strategy.

# Conclusiones

Los patrones de comportamiento son esenciales para la construcción de sistemas robustos, modulares y adaptables. Aplicados correctamente, permiten implementar flujos complejos, responder a eventos, gestionar estados y extender funcionalidades sin modificar la estructura del sistema. Su uso inteligente promueve principios SOLID, mejora la mantenibilidad del código y facilita el trabajo en entornos colaborativos y cambiantes.

# Referencias

Bolaños Rodríguez, H. A. (2025). Curso Patrones de Diseño OO - Semana 12. Universidad Cooperativa de Colombia.

.