

# Incursiones Navales

Autores : Levi Barros García & Raúl Meijide Couto

Logins : [levi.barrosg@udc.es](mailto:levi.barrosg@udc.es) & [r.meijide@udc.es](mailto:r.meijide@udc.es)

## Introducción

La clave para el éxito de este proyecto reside en la implementación cuidadosa de patrones de diseño que facilitan la gestión de la complejidad y promueven la modularidad del código. Un patrón de diseño crucial empleado en este contexto es el Patrón Composición. Este patrón permite tratar tanto a objetos individuales como a composiciones de objetos de manera uniforme, brindando una estructura jerárquica para representar la diversidad de nodos y eventos en la ruta.

A lo largo de este informe, exploraremos la aplicación práctica del Patrón Composición en el diseño de nodos de bifurcación, nodos de ruta fija y nodos finales. Analizaremos cómo este enfoque facilita la simulación de la flota a través de la ruta, permitiendo la adaptabilidad a cambios y la fácil extensión del sistema.

## Nuestros principios SOLID

- Principio de Responsabilidad Única (SRP):

En consonancia con el SRP, cada clase en el proyecto tiene una única razón para cambiar. Por ejemplo, la clase Flota se encarga exclusivamente de representar el estado de una flota y gestionar sus atributos, mientras que las clases de nodos se centran en la lógica específica de su comportamiento sin involucrarse en responsabilidades ajenas.

- Principio de Abierto/Cerrado (OCP):

El proyecto sigue el principio OCP al diseñar clases que son abiertas para la extensión pero cerradas para la modificación. La introducción de nuevos tipos de nodos, como nodos de batalla o de avistamiento, puede lograrse sin alterar las clases existentes, permitiendo así una fácil extensión del sistema.

- Principio de Sustitución de Liskov (LSP):

La herencia y la implementación de interfaces en las clases de nodos cumplen con el principio LSP. Cada clase derivada puede ser sustituida por su clase base (o interfaz) sin afectar la integridad del programa. Esto permite un manejo uniforme de los nodos en la simulación, facilitando la composición de rutas complejas.

- Principio de Segregación de Interfaces (ISP):

Las interfaces, como Nodos, se mantienen cohesivas y específicas para cada tipo de nodo, siguiendo el principio ISP. Esto permite que las clases implementen solo las interfaces que son relevantes para ellas, evitando la dependencia de métodos innecesarios.

- Principio de Inversión de Dependencia (DIP):

La implementación sigue el DIP al depender de abstracciones en lugar de implementaciones concretas. Las clases como Simulador interactúan con las interfaces, permitiendo una fácil sustitución de implementaciones concretas de nodos sin afectar el flujo general de la simulación.

## Uso del Patrón Composición

El Patrón Composición se erige como un pilar fundamental en la arquitectura de este proyecto, proporcionando una solución elegante para la representación jerárquica y compuesta de nodos en la ruta estratégica de la flota. Este patrón facilita la construcción de estructuras complejas al tratar tanto a objetos individuales como a sus composiciones de manera uniforme.

En este proyecto, la aplicación del Patrón Composición se evidencia en la jerarquía de clases que representan nodos. La interfaz Nodos actúa como el componente base, definiendo las operaciones comunes a todos los nodos, como la capacidad de continuar la ruta y obtener el nombre del nodo. Las clases concretas, como NodoBifurcacion, NodoRutaFija, y NodoFinal, actúan como hojas individuales o nodos compuestos, dependiendo de su jerarquía.

### Interfaz Nodos como Componente Base:

La interfaz Nodos define las operaciones comunes a todos los nodos. Cada clase concreta implementa esta interfaz, garantizando que todos los nodos, independientemente de su tipo, proporcionen una interfaz coherente para la simulación.

### Clases Concretas como Hojas y Nodos Compuestos:

Las clases concretas, como NodoBifurcacion, NodoRutaFija, y NodoFinal, actúan como hojas individuales o nodos compuestos, según su posición en la jerarquía. Por ejemplo, NodoBifurcacion puede tener nodos hijos (izquierdo y derecho), formando así una composición.

### Uso Recursivo para Componer Estructuras Complejas:

La simulación utiliza un enfoque recursivo al interactuar con los nodos. El método simular en la clase Simulador se adapta al Patrón Composición al tratar tanto a nodos hoja como a nodos compuestos. De esta manera, la ruta completa, compuesta por bifurcaciones, rutas fijas y nodos finales, se maneja de manera uniforme.

El Patrón Composición ha permitido la construcción de una estructura jerárquica de nodos de manera elegante y extensible. Proporciona un mecanismo eficaz para componer rutas estratégicas complejas, facilitando la simulación de incursiones de flotas a través de eventos

variados y decisiones en la ruta. La uniformidad en la manipulación de nodos individualmente y en grupos mejora la mantenibilidad y escalabilidad del sistema.