

Trabajo Práctico Final

Terminal Portuaria

Integrantes

- Miguel Laiun(laiunmiguel@hotmail.com)
- Maximiliano Souto(max.xies12@gmail.com)
- Luis Salas Castromonte (sc.luis93@gmail.com)

Patrones de diseño

State

Para representar las fases de un Buque utilizamos el patrón State.

Los roles que representan en el Buque son:

- **Context**: Buque
- **State**: IEstadoBuque
- **ConcreteState**: Outbound, Inbound, Arrived, Working, Departing

Strategy

Para representar el mejor circuito utilizamos el patrón Strategy.

Los roles que representan son:

- **Context**: TerminalGestionada
- **Strategy**: IStrategy
- **Concrete Strategy**: MenorTiempo, Menor Precio, MenoresParadas

Composite

Utilizamos el patron Composite para la búsqueda de rutas marítimas y así poder realizar búsquedas combinadas.

Los roles que representan son:

- **Component** : IBuscadorViajes
- **Composite** : BuscadorLogico
- **Leaf** : BuscadorPuertoDestino, BuscadorFechaDeSalida, BuscadorFechaDeLlegada, BuscadorAnd, BuscadorOr

Decisiones De Diseño

Se eligió el patrón State en representación del Buque para poder cambiar su comportamiento en tiempo de ejecución. El patrón Strategy se eligió porque permite cambiar dinámicamente la estrategia de búsqueda del mejor circuito en tiempo de ejecución, de acuerdo con los criterios especificados. Además, el modelo queda abierto para agregar nuevas estrategias sin agregar código existente. Por último, en el buscador se eligió el patrón Composite para poder realizar búsquedas compuestas.

Detalles De Implementación

Creamos la clase Turno para quitarle responsabilidad a la clase Orden y así no romper el principio de Solid de una sola responsabilidad por clase.

Tenemos una clase Coordenada que representamos los puntos x e y con el tipo Double.

Implementamos Servicio como una interfaz para en el futuro poder implementar nuevos servicios sin la necesidad de modificar los ya existentes.

Implementamos la interfaz "CompaniaDeEmail" para simular el envío de emails con las facturas y el desglose de servicios.

Para los test utilizamos mock para mockear clases que no son la clase principal a testear.