

PRÁCTICA DE DISEÑO

1. Introducción

En este ejercicio se desarrolla un diseño del software de un gestor de billetes de autobús, en el que los usuarios pueden buscar billetes en base a una serie de características.

Este diseño contiene una breve explicación de los principios y patrones usados para conseguir una correcta implementación, junto con dos diagramas UML, uno de clases y otro dinámico.

2. Principios de Diseño

Los principios respetados en nuestro trabajo son:

- **Principio de responsabilidad única:** Cada objeto tiene una responsabilidad única y todos sus servicios están alineados con dicha responsabilidad. En nuestro trabajo se puede ver en el diagrama de secuencia realizado.

- **Principio de inversión de la dependencia:** La inversión de la dependencia es la estrategia de depender de interfaces o clases y funciones abstractas, en vez de depender de clases y funciones concretas. Está presente tanto en la declaración de todas las listas como List en lugar de ArrayList y en la interfaz "Componente".

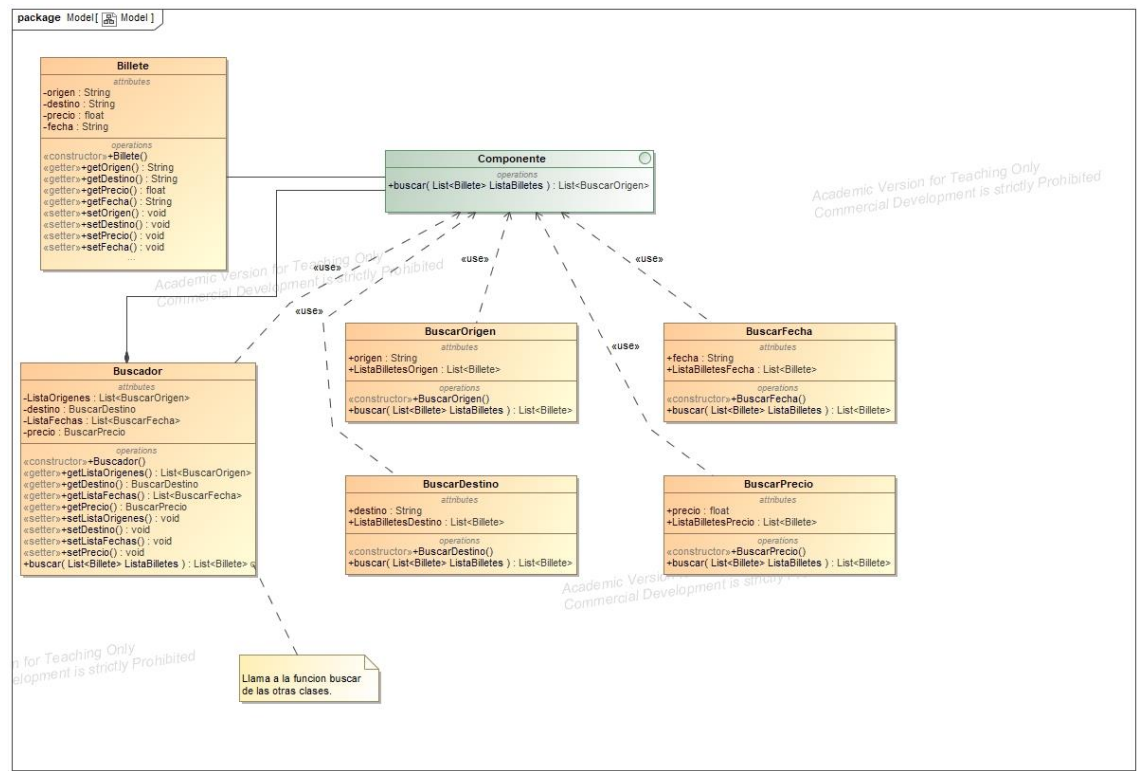
- **Principio de segregación de interfaces:** Muchos interfaces específicos para cada cliente son mejores que un único interfaz de propósito general. En nuestro código se puede observar que la interfaz "Componente" solo contiene una instrucción, lo que consigue que se cumpla este principio.

3. Patrones de Diseño

El patrón utilizado en nuestro trabajo se trata del Patrón Composición:

- **Patrón Composición:** Es un patrón estructural que se utiliza para componer objetos en estructuras de árbol que representan jerarquías todo-parte. Este patrón permite tratar uniformemente a los objetos y a las composiciones. Se puede observar claramente en la interacción entre la interfaz "Componente" y las clases "Buscador" y "BuscarOrigen", "BuscarDestino", "BuscarFecha" y "BuscarPrecio". Siendo el primero la composición y los demás los componentes concretos.

4. Diagrama de Clases:



5. Diagrama de secuencias

