

PRÁCTICA DE DISEÑO:

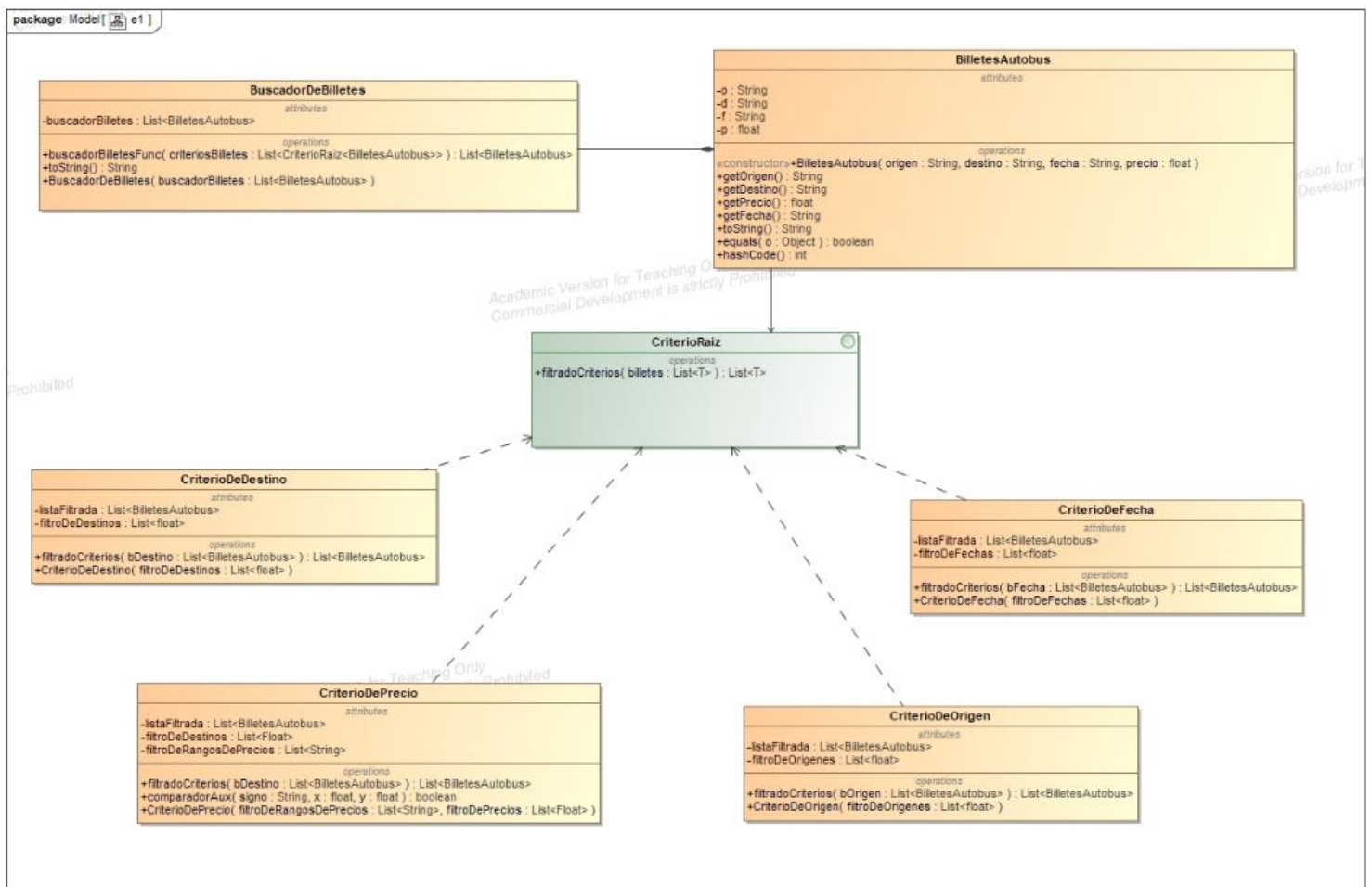
1. Gestión de Billetes

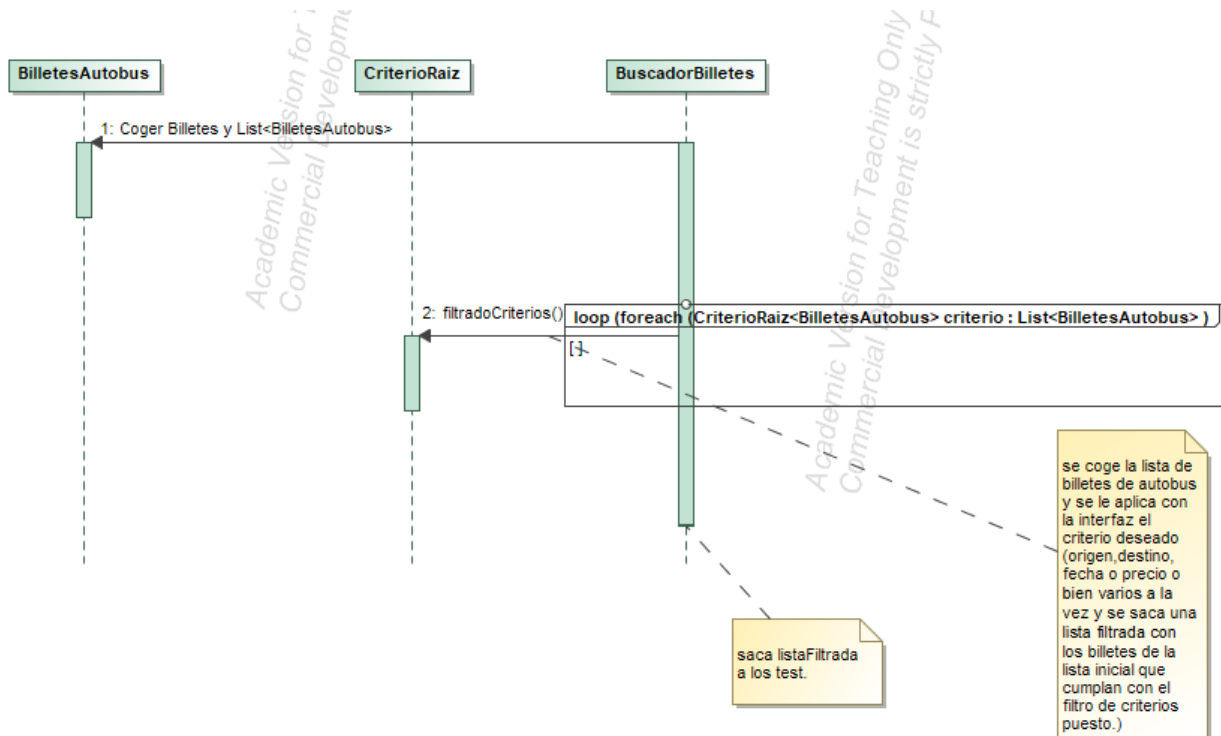
1.1. Principios de diseño utilizados son el **principio abierto-cerrado** ya que en el desarrollo del ejercicio queremos que en un futuro se puedan expandir a nuevas funcionalidades de las que ya están puestas como los criterios de filtrado pero que se puedan poner otras funcionalidades de filtrado. Es decir programamos abierto a la expansión pero cerrado a la modificación. Se puede ver mediante la interfaz CriterioRaiz que nos permite meter más criterios de filtrado de billetes en un futuro. También empleamos, ya que se ha mencionado, el **principio de la inversión de la dependencia** ya que mediante una interfaz hacemos que el código dependa de un criterio abstracto que se le meta por parámetro y no de un criterio en concreto.

1.2. Patrones de diseño utilizados:

En este ejercicio hemos utilizado el **patrón composición**. Es un patrón estructural que permite crear objetos a partir de una interfaz haciendo que todos sean similares y permitiendo que todos sean tratados de la misma manera.

1.3. Diagramas UML





2. Planificador de Tareas

2.1. Principios de diseño utilizados son el **principio abierto-cerrado** ya que en un futuro es posible que aparezcan nuevos ordenes de ejecución a petición del Gestor de proyectos y de esta manera que se integren fácilmente en el código sin complicación. Con esto conseguimos que el código sea abierto a la expansión y cerrado a la modificación de lo ya hecho. También empleamos el **principio de la inversión de la dependencia** ya que mediante una interfaz hacemos que el código dependa de unas funciones fijas las cuales actuarán de un modo u otro según el Gestor de proyectos desee. En este caso escogerá si el ordenFinal de la ejecución será con dependencia fuerte, débil u orden jerárquico.

2.2. Patrones de diseño utilizados

En este ejercicio hemos utilizado el **patrón de estrategia**. Se trata de un patrón que se utiliza para definir una familia de algoritmos, encapsularlos y hacerlos intercambiables.

En este ejercicio, la familia de algoritmos son las distintas dependencias (dependencia fuerte, débil y jerárquica). La clase ClasificadorDeTareas será nuestro clasificador el cual buscará las dependencias primarias, es decir, se encarga de

AUTORES DE LA PRÁCTICA:
DAVID TWINE PADÍN Y LUCÍA DOCAMPO RODRÍGUEZ.

buscar aquellas dependencias que en ningún lado dependan de otra dependencia. La interfaz `DepRaizInterfaz`, es nuestra interfaz común a todos los algoritmos la cual posteriormente es la encargada de darnos la flexibilidad de usar un algoritmo de dependencia u otro.

Usamos el patrón estrategia porque nos permite la flexibilidad, según la utilidad que quiera en este caso el Gestor de proyectos, de utilizar la dependencia que más nos sea de interés en el preciso momento de elección.

2.3. Diagramas UML

