EJERCICIO 2 PRACTICA DE DISEÑO

Principios de Diseño.

En este ejercicio utilizaremos el principio de responsabilidad única, en el que cada objeto debe tener una responsabilidad única que este enteramente encapsulada en la clase. Todos los servicios que provee el objeto están estrechamente alineados con dicha responsabilidad. En nuestro código las clases que cumplen este principio son las clases ClienteSencillo, ClienteDetallado y la clase OtrosClientes, que tienen como único propósito mantenerse actualizados.

Tambien utilizamos el principio de sustitución de Liskov, en el cual al redefinir un método en una clase derivada, solo podemos sustituir una precondición por otra precondición mas débil y una postcondición mas severa. En el código cuando se pide que se pase un observador podemos pasar tres tipos distintos (ClienteSencillo, ClienteDetallado y OtrosClientes).

Patron de Diseño Usado.

Para realizar este ejercicio utilizamos el patron Observador, que nos permite definir una dependencia uno a muchos entre objetos de tal forma que, cuando el objeto cambie de estado, todos sus objetos dependientes sean notificados y actualizados automáticamente. Traduciendo esto a nuestro ejercicio quiere decir que cuando una acción cambia su estado avisaremos del cambio a diferentes clientes, que serán responsables de mantener actualizada esa información. En este caso utilizamos el modelo push, es decir, el observable le indica a los observers de lo que ha cambiado de manera que estos podrán actuar en consecuencia.

Diagrama de Clases

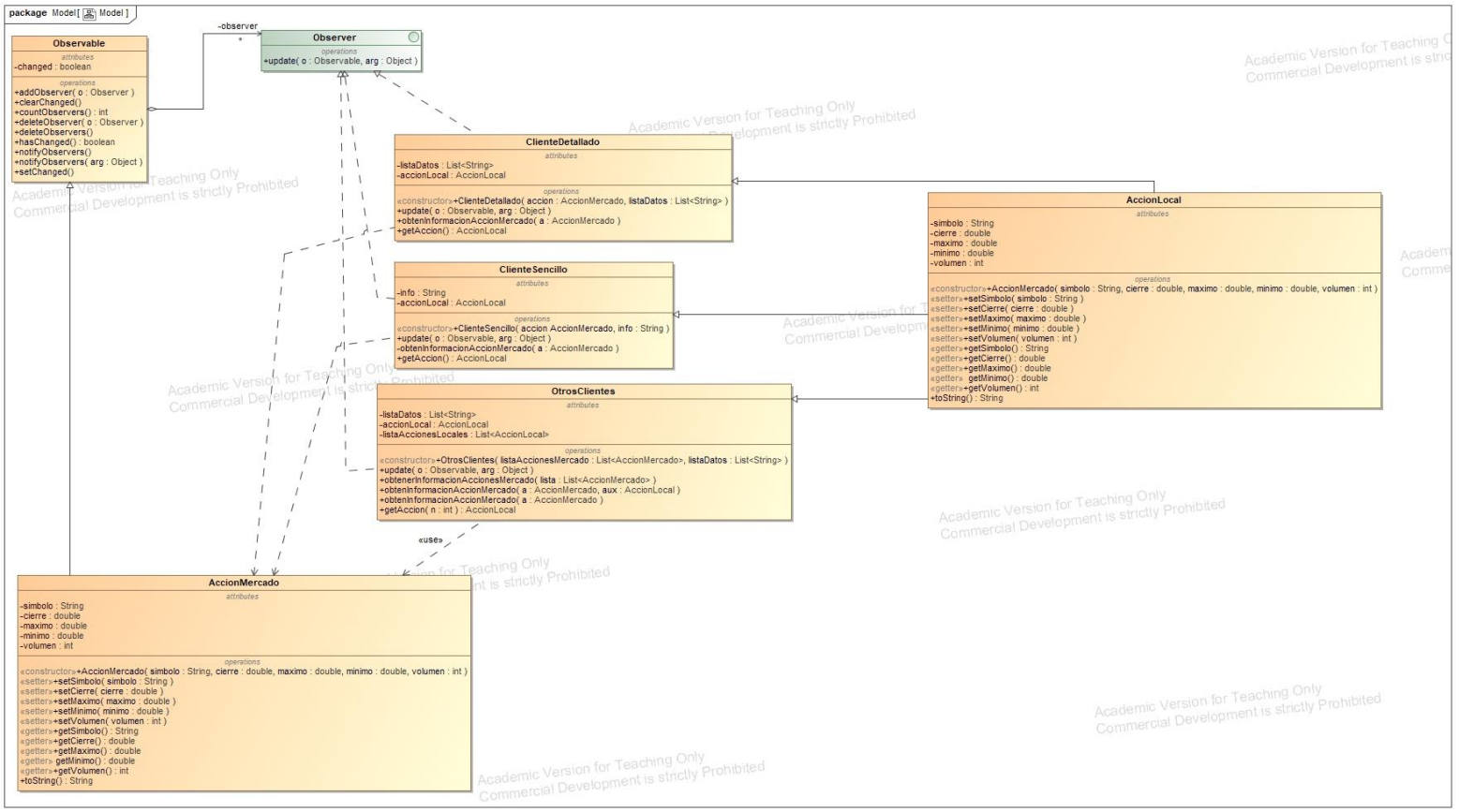

Diagrama Dinamico