

# *Informe Programación II*

Grupo 4(Ivonne Flores)

**Estudiantes:** Juan Carvajal

Sebastián Rosas

**Docente:** Geoffrey Hecht

**Asignatura:** Programación II

## Tema 3: Sistema de reserva de asientos de autobús

Diagrama Use Case

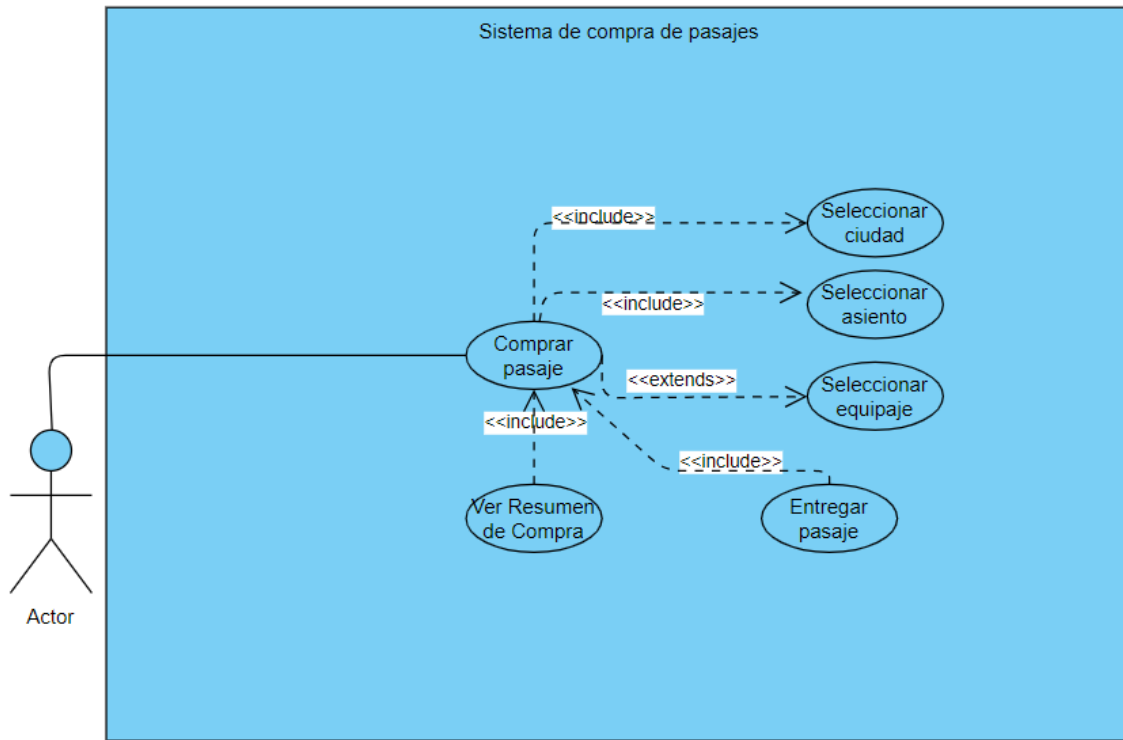
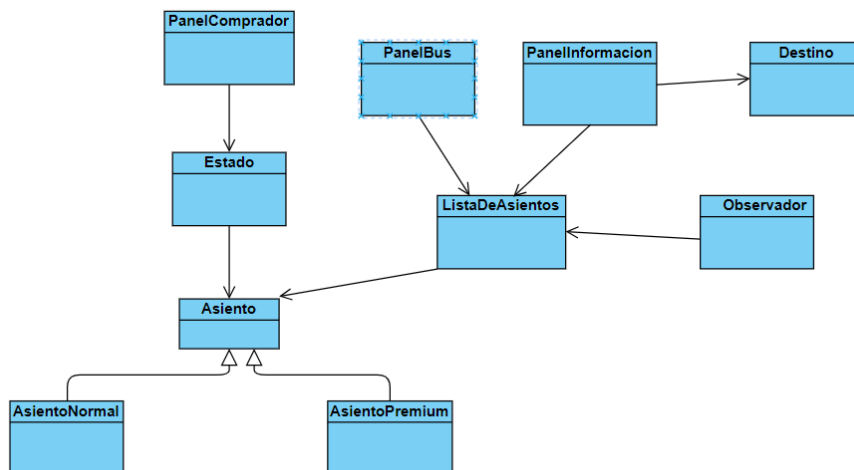


Diagrama UML



Uso de patrones:

Patrón Observer: Este patrón se usó para mantener a todas las partes informadas acerca de los cambios en el estado de la aplicación. El Estado es el sujeto y PanelComprador es el observador. Se notifica al observador a través de este patron cuando un asiento es agregado o eliminado para actualizar su vista.

Patrón Composite: Este patrón se usa en la creación y manejo de los paneles y componentes de swing, los cuales son agregados a paneles más grandes para formar la interfaz de usuario. Este patrón permite que un grupo de objetos sea tratado de la misma manera que una instancia única de un objeto.

Patrón State: Este patrón se implementa a través del manejo de los botones de los asientos en la clase ListaDeAsientos. Los botones tienen dos estados posibles: seleccionado o no seleccionado, que cambian su color y su estado de reserva.

Nuestro modelo se basó en una interfaz simple pero clara, un panel en el panel superior que muestra los destinos, un panel inferior que reacciona a los eventos del mouse y un panel lateral derecho en donde se desarrolla un resumen de la petición del cliente.

El bus tiene dos tipos de asientos del 1-8 son premium y cuestan \$1000 más que los asientos del 9-16. Cuando un asiento se clickea entra en un estado marcado para su compra, el precio también varía con los distintos itinerarios seleccionados.

En el panel lateral se muestra el destino, los números de asientos, qué tipo de asientos son, la opción de equipaje (no totalmente implementada) y el precio total. Luego de eso se confirma la compra.

Algunos otros elementos que no pudimos implementar o que implementamos parcialmente fueron: El método para dejar reservado el asiento, los horarios y los tipos de buses.

## Captura de pantalla de la Interfaz



## Conclusión y reflexiones

Al comienzo del proyecto teníamos muchas ideas sobre como íbamos a implementar nuestras herramientas, las cuales a medida progresábamos tuvimos que cambiar, una de las primeras ideas que desechamos fue la creación de un segundo piso para el bus , ya que tuvimos varios problemas al tratar de implementarlo ,lo principal fue que se nos repetía el piso 1 con el piso 2 y no se guardaban las selecciones que se les hacían, tratamos de varias maneras de solucionarlo pero finalmente decidimos enfocarnos en que se efectuara bien la compra de los asientos en un solo piso. Otro problema que tuvimos fue ocupar el patron Observer, principalmente fue debido a la poca experiencia que teníamos ocupando dicho patrón, pero con los ejemplos encontrados en [refactoring.guru](http://refactoring.guru) se hizo más a meno ocupar dicha herramienta y finalmente se agrego a nuestro catalogo de patrones sin problemas extra.

Como autocrítica podemos decir que no seguimos una metodología de trabajo organizada y nos faltó constancia, nos dimos cuenta la gran capacidad de coordinación, planificación y estructura que requiere un proyecto de Java. Es bastante fácil ir por el camino equivocado cuando no se tiene una clara imagen de lo que se quiere hacer y cómo se hará. Siempre tuvimos las ideas generales pero fuimos desarrollándolas a medida que se presentaban. Nos queda como lección que el diseño de un proyecto es tan importante como el desarrollo del mismo.