

Venta de pasajes de Bus

## Grupo 15

503212-1 Programación II

Profesor: Geoffrey Jean-Pierre Christophe  
Hecht

Referente: Ivonne Flores Roa

Gabriel Huerta, Martin Rubilar y Benjamín  
González

---



## **Índice**

Introducción.....	(3)
Tema.....	(3)
Diagramas.....	(4-5)
Patrones Utilizados.....	(6)
Interfaz.....	(7)
Conclusión.....	(8)





## **Introducción**

En el dinámico mundo de los servicios de transporte, la calidad y el confort son factores clave que aseguran una experiencia satisfactoria para los empleados y clientes de la empresa. En este contexto, desarrollamos un sistema de reserva de asientos de autobús, una herramienta diseñada para simplificar y mejorar el proceso de selección y reserva.

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

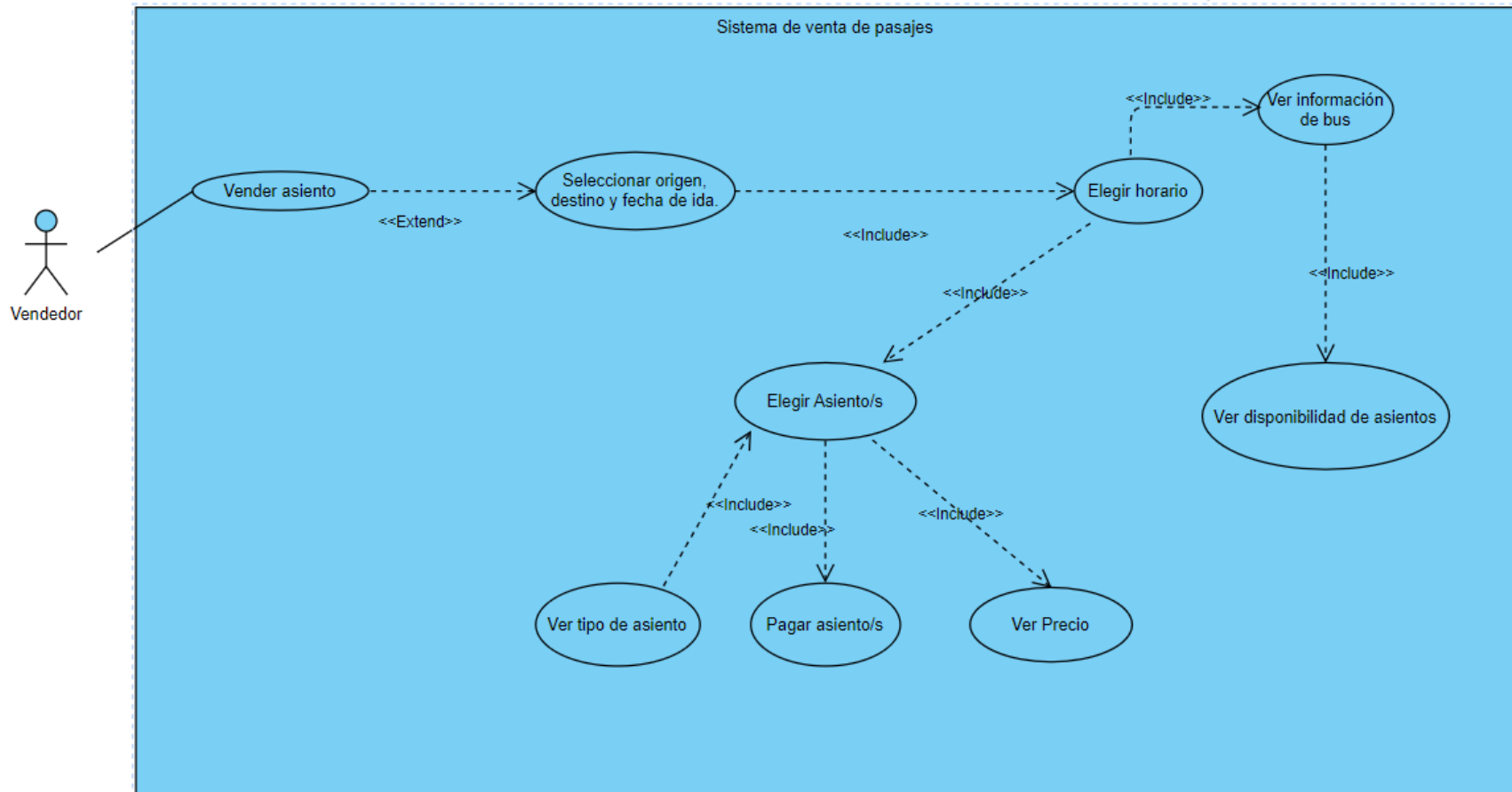
- El sistema de reserva de asientos de autobús permite al personal de una empresa de autobús elegir y reservar asientos de forma conveniente por su cliente. Los usuarios pueden visualizar una representación gráfica de los asientos disponibles en el autobús y seleccionar los que deseen ocupar. El sistema muestra información detallada sobre cada asiento, como su ubicación, número y categoría (por ejemplo, semi cama, Salón Cama).
- Una vez que los usuarios seleccionan los asientos deseados, el sistema verifica la disponibilidad y permite confirmar la reserva mostrando el precio a pagar. En caso de que algún asiento ya esté reservado por otro pasajero, se informa al usuario para que pueda elegir otro asiento disponible. El personal confirma el pago (no gestionado por el sistema) lo que reserva los asientos.
- El sistema debe gestionar varios tipos de autobuses (por ejemplo, con diferente número de plazas, o de 1 o 2 pisos...).
- El sistema debe mostrar un menú que permita seleccionar el autobús en función de su horario y recorrido (se supone que estos datos están disponibles con los autobuses vacíos cuando se lanza el software).



[illegible]



## Diagrama de Casos de Uso





## **Patrones Utilizados**

En nuestro código, hemos aplicado el patrón de diseño llamado "Observer". Este patrón se refleja en la interfaz denominada "AsientoClickListener". En esta implementación, la clase "VistasAsientos" actúa como el "sujeto", mientras que los objetos que implementan la interfaz "AsientoClickListener" actúan como observadores. El método "onAsientoClick()" se invoca al hacer clic en un asiento, y se utiliza para informar a otros componentes que un asiento ha sido seleccionado o deseleccionado.

Un ejemplo claro de la aplicación de este patrón se encuentra en la clase "PanelDatosBus", la cual es un observador notificado cada vez que un asiento es seleccionado o deseleccionado. En este contexto, el patrón Observer facilita la actualización dinámica de información, como el cálculo del precio total a pagar, en respuesta a eventos específicos, como la selección o deselección de asientos. Este enfoque proporciona una estructura flexible y desacoplada para gestionar interacciones entre componentes, mejorando la modularidad y mantenibilidad del código.





# Interfaz

## **Ventana Comprar:**

### Descripción:

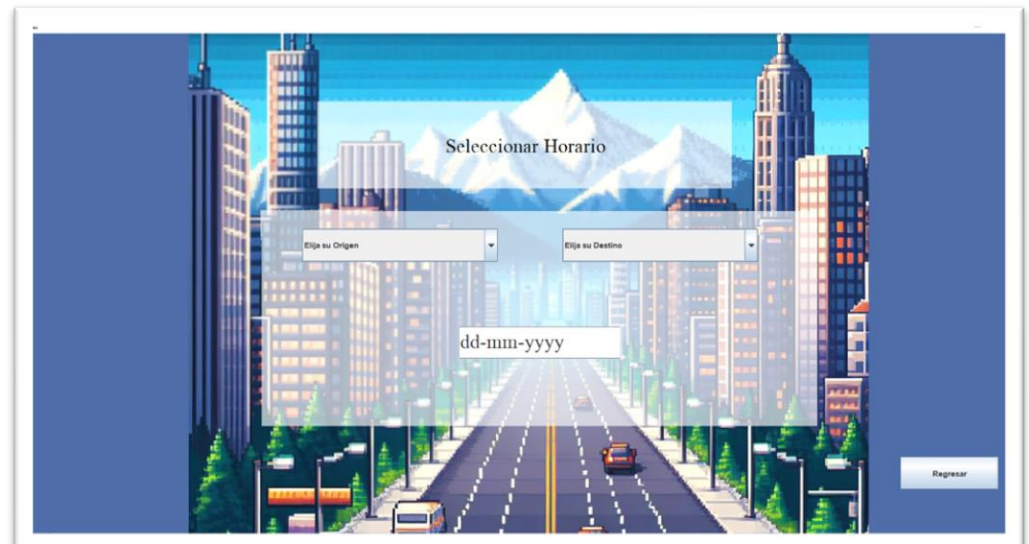
Primera ventana que se muestra al inicializar el programa. Es el botón de inicio y estado 0 del programa para comenzar con la venta.



## **Ventana Destino:**

### Descripción:

Segundo panel que se muestra, se debe ingresar el origen y destino además de la fecha para hacer la venta efectiva del pasaje. No se puede seleccionar la misma ciudad de origen en el destino y viceversa.





### Ventana Selección Horario:

#### Descripción:

En la tercera ventana se despliegan los diferentes horarios con sus respectivos buses según los datos introducidos anteriormente.



### Ventana Selección Asiento:

Descripción: Cuarta ventana en aparecer, se seleccionan todos los asientos que se planean comprar. Se suma el precio con cada elección y se puede pasar del piso 1 al piso 2 en caso de que el bus cuente con segundo piso. Al presionar el botón de Comprar quedan marcados como "Asiento Ocupado". Se calcula el Precio y IVA a tiempo real.







## **Conclusión**

El proceso de desarrollo del programa estuvo fuertemente marcado por la toma de **decisiones** importantes que influyeron en el rumbo del proyecto. Nos vimos obligados a abandonar algunas ideas por cuestiones de tiempo y complejidad de implementación. Por otra parte, la toma de decisiones frente al apartado gráfico fue un tema bastante repetido, pues la mejora de este, continuamente nos llevó a siempre cuestionar el actual y si debiese ser mejorado, llegando a un resultado que nos complace a los 3.

Por otra parte, uno de los mayores desafíos que enfrentamos fue la diferencia de ideas sobre la lógica del programa, resultando en un gran **problema** a afrontar. Si bien pudimos sacar el proyecto adelante, la constante acción de primero “descifrar” el código del compañero para luego recién comenzar a implementar nuevas funciones, fue una gran pérdida de tiempo que se hubiera podido solucionar fácilmente si tan solo hubiéramos contado con una mejor documentación y explicación de cada cosa.

En conclusión, la constante mejora del proyecto por c/u sin tanta comunicación fue lo más complejo, pero en general nos sentimos complacidos con el resultado final entregado.

Link: [https://github.com/benjag22/Proyecto\\_Final](https://github.com/benjag22/Proyecto_Final)

