



Informe Proyecto Semestral

Jorge Slimming, Benjamin Henriquez y Daniel Támara

Grupo 8

Universidad de Concepción

Facultad de Ingeniería, Ingeniería Civil Informática

Desarrollo Orientado a Objeto

Profesor: Geoffrey Jean-Pierre Christophe Hecht

Concepción, Chile

8 de Julio de 2024

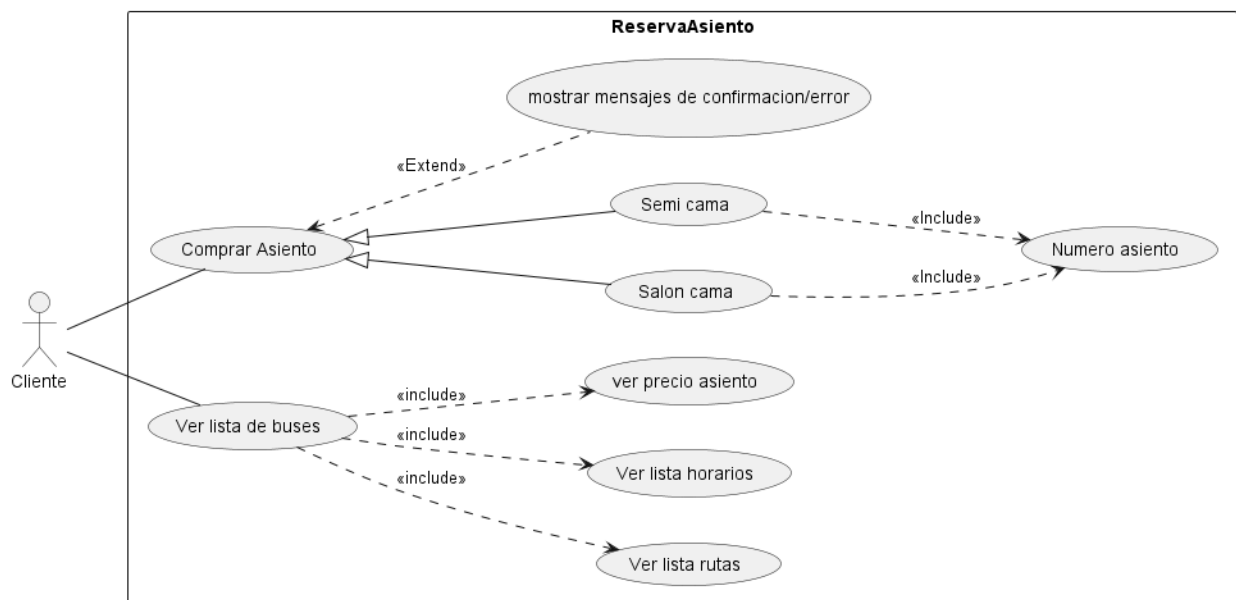
Enunciado

Tema 2: 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)

Diagrama de clases UML: [Subido al repositorio](#)

Diagrama de casos de uso:



Patrón de diseño utilizado:

El patrón de diseño utilizado para el desarrollo de este proyecto fue el patrón de diseño Decorator, pues, para este caso era conveniente su uso porque servía para agregar dos cosas a los buses: La primera era incrementar el precio de los asientos de cada bus dependiendo de su destino, ya que se debe pagar un precio mayor dependiendo de qué tan lejos el cliente desea viajar, y la segunda para calcular una hora aproximada de la llegada del autobús a su destino, pues de nuevo, se toma en cuenta la lejanía de cada destino para calcularla, así, haciendo decoradores para cada uno de los posibles destinos, podemos agregarle esos incrementos a cada bus **sin necesidad de hacerlo manualmente**.

Captura de pantalla de la interfaz:

Transportes papu

AGENCIA

SALIDA/LLEGADA

DESTINO

Nº PISOS

ASIENTOS DISPONIBLES

PISO 1

PISO 2

Papu Buses	06:00/09:00	Papus Ville	2	Piso 1: 16 Piso 2: 16
Tilin Express	12:00/17:00	Tilin Town	2	Piso 1: 14 Piso 2: 16
Sigma Service	18:00/01:00	Sigma City	1	Piso 1: 16
Skibidi Travel	21:00/05:00	Skibidi Springs	1	Piso 1: 16
Mewing Mobility	21:00/07:00	Mewing Metr.	1	Piso 1: 16

FRONT

BACK

Información de Asientos

Asiento Número=4, VIP=No, Disponibilidad=Disponible, Precio=26000

RESERVAR

Decisiones tomadas durante el desarrollo del proyecto:

Unas de las decisiones más importantes tomadas durante el desarrollo del proyecto fueron las siguientes:

- Cambiar en la parte Lógica la clase “Asiento”: En un inicio la clase Asiento era una superclase de la cual provenían dos subclases que sería “AsientoNormal” y “AsientoVIP”, sin embargo, se volvió incómodo trabajar con dos clases distintas que con una sola, por lo que optó por utilizar un valor booleano “esVIP”, en este caso, para distinguir si un asiento es Normal o es VIP, de esta forma se volvió más fácil la manipulación de los asientos a la hora de asignarlos a cada Autobús.
- Cambiar en la parte gráfica el uso de dos ArrayList de “PanelesAsiento”: En un principio la manera de trabajar los paneles que representan los asientos en la interfaz era con dos ArrayList’s de “PanelAsiento”, donde un ArrayList correspondería los asientos del piso 1 y el otro a los asientos del piso 2 si se da el caso de que el autobús posea piso 2, sin embargo al estar los asientos en el mismo lugar se superponen y el MouseListener aplicado a estos para realizar la acción de seleccionarlos para luego reservarlos actuaba para ambos paneles en lugar de solo uno, por lo que se tomó la decisión de solo usar un ArrayList en este caso llamado PanelAsientoPisoList cuyos elementos son objetos del tipo “PanelAsiento” y los cuales se van reutilizando para mostrar en pantalla la disponibilidad de cada asiento en cada piso de cada autobús.

Problemas encontrados:

El problema más significativo encontrado durante el desarrollo tiene que ver con el punto anterior pues corresponde al manejo de los dos ArrayList’s de “PanelAsiento” pues en un inicio pensamos que utilizar dos ArrayList’s sería más sencillo, sin embargo, a la hora de utilizar el MouseListener sobre los paneles, el evento ocurría simultáneamente para ambos lo que provocaba un comportamiento incorrecto del programa a la hora de seleccionar los asientos que se deseaban reservar.