

# **Sistema de reserva de asientos para autobuses**

---

Integrantes:

Joseph Matamala  
Felipe Tillería  
Darío Sepúlveda

Geoffrey Hecht  
Desarrollo Orientado a Objeto  
08/07/2024

# ÍNDICE

1.	SISTEMA DE RESERVA DE ASIENTOS DE AUTOBÚS.....	3
2.	CASOS DE USO.....	4
3.	PATRONES UTILIZADOS.....	5
4.	DIAGRAMA UML.....	6
5.	INTERFAZ GRÁFICA.....	7
6.	CONCLUSIONES.....	8

# Sistema de reserva de asientos de autobús

## ENUNCIADO

- 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).

El proyecto final se mantuvo fiel al enunciado original, por lo que no fue necesario agregar nada a este para adaptarlo al programa.

# CASOS DE USO

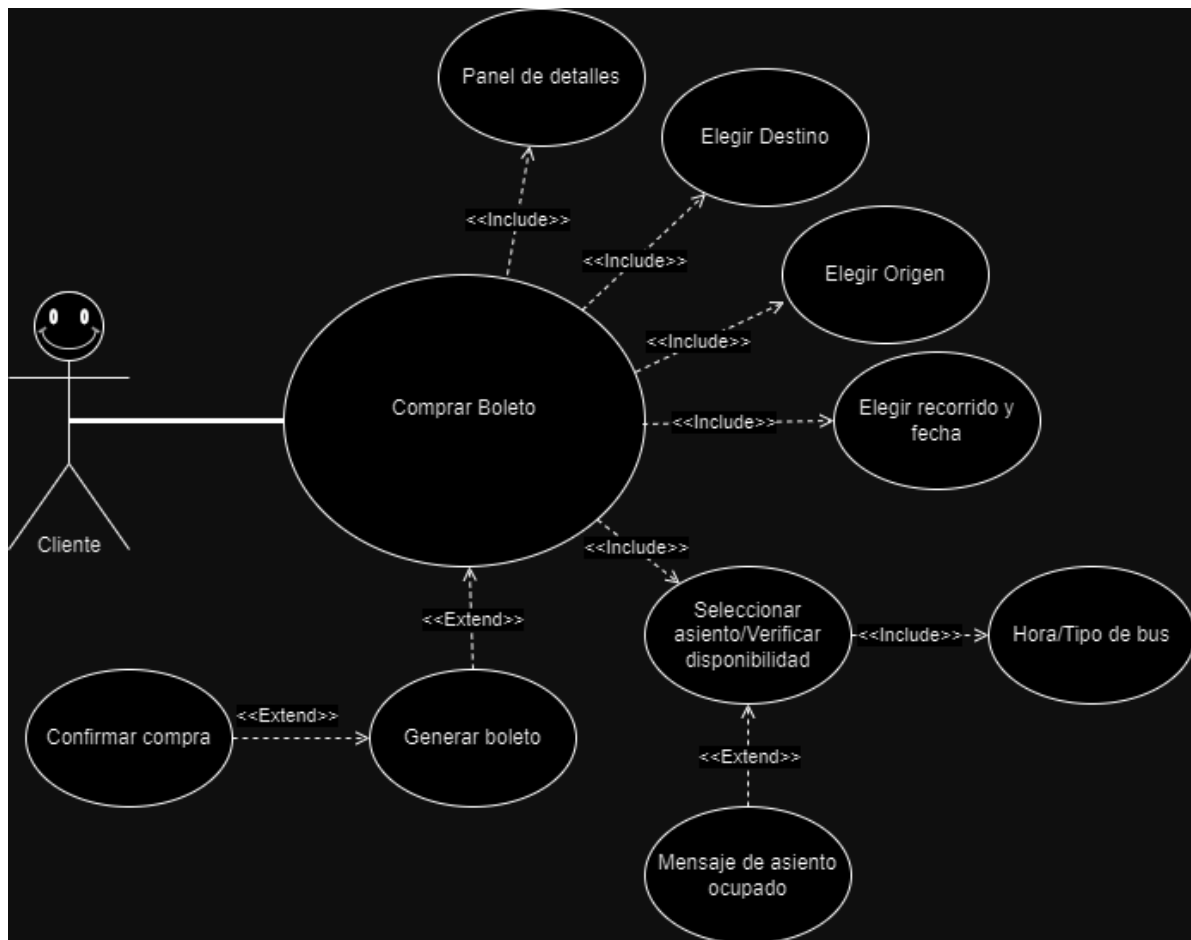


Imagen que representa los casos de uso del programa.  
(Se ha subido una copia al repositorio GitHub para facilitar la visualización)

Los casos de uso son las opciones que tiene el usuario del programa para elegir los boletos que él quiera, analizando todos los detalles de su compra, y verificando la disponibilidad y correcto formato del pasaje.

# PATRONES UTILIZADOS

## Patrón Singleton

Clase que lo utilizó:

- TipoAsientoEventManager

Este patrón fue utilizado porque así aseguramos que haya una única instancia que controle los listeners (añadir, eliminar).

## Patrón Observer

Clases que lo ocuparon:

- Asiento
- TipoAsientoEvent
- TipoAsientoListener

Decidimos utilizar este patrón para que varios componentes del sistema reaccionen a los cambios en el estado de los asientos (al ser seleccionados, comprados, etc).

## Patrón Factory Method

Clases que lo ocuparon:

- Asiento
- AsientoCama
- AsientoSemiCama
- AsientoEjecutivo

Usamos factory method ya que nos permite crear diferentes tipos de asientos (cama, ejecutivo, etc.) sin acoplar el código cliente a las clases concretas, esto simplifica la creación de objetos.

# DIAGRAMA UML

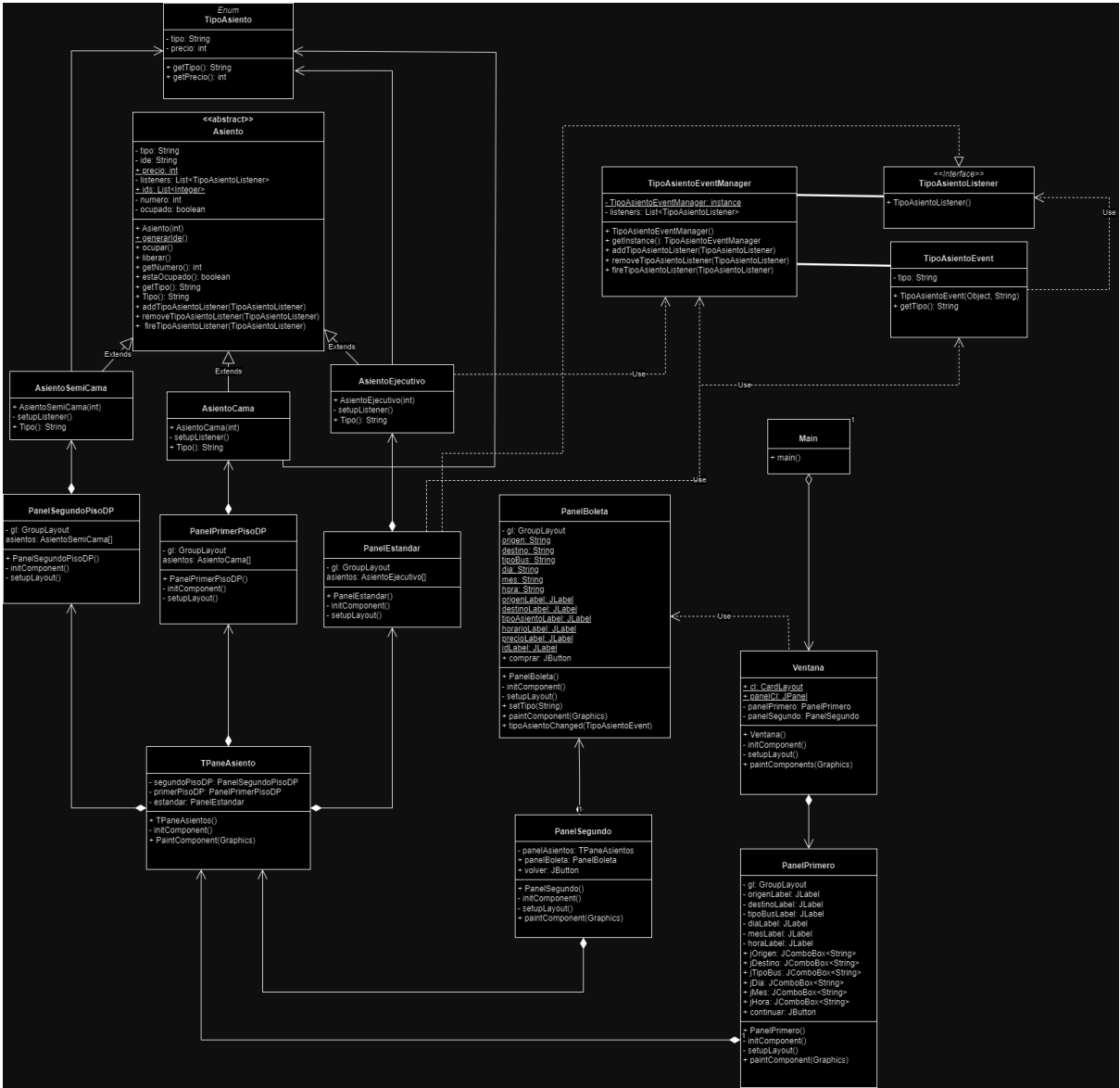


Diagrama UML del contenido **gráfico** del programa  
(Al igual que el diagrama casos de uso, en caso de no poder visualizarse correctamente debido al tamaño se ha subido una copia al repositorio GitHub)

# INTERFAZ GRÁFICA



The screenshot shows a window titled "Touhou Bus" with a yellow background and decorative black circles. The interface contains several dropdown menus for selecting travel details. The "Origen" (Origin) dropdown is set to "Concepcion", "Tipo de Bus" (Bus Type) is set to "Estandar" (Standard), and "Destino" (Destination) is set to "Linares". Below these, there are three more dropdowns for "Dia" (Day) set to "01", "Mes" (Month) set to "Enero" (January), and "Hora" (Hour) set to "00 : 00". A blue "Continuar" (Continue) button is positioned below the date and time selections. The bottom right of the window features a stylized illustration of a bus.

Origen	Tipo de Bus	Destino	Dia	Mes	Hora
Concepcion	Estandar	Linares	01	Enero	00 : 00

Continuar

Imagen del menú inicial del programa.

A la hora de abrir el programa este nos da la bienvenida con esta pantalla. Aquí es donde seleccionamos el lugar de origen, el destino al que vamos, el tipo de bus que vamos a elegir (De planta doble o singular), y la fecha en la que queremos viajar.



Imagen de la pantalla principal del programa

Acá encontramos un panel con *JToggleButton* simulando los asientos de bus, al seleccionarlos se mostrará el tipo de asiento, su ID, y se le sumará su valor al precio total. Se pueden comprar hasta 10 asientos.



# CONCLUSIONES

A la hora de iniciar el proyecto se creó un plan de acción para poder ejecutarlo, sin embargo, se tuvieron que realizar cambios sobre la marcha a la hora de hacer uso de este:

## Interfaz gráfica

La interfaz gráfica se pudo lograr satisfactoriamente, no obstante siempre hay espacio para mejorar.

Si bien su calidad pragmática es la indicada, el apartado estético de esta podría ser cambiada. En este caso se usaron colores sólidos, pero otras herramientas como más imágenes o animaciones pudieron haber sido utilizadas para una mejor apariencia para el usuario.

## Problemas encontrados

La comunicación entre clases fue el problema principal, especialmente con las encargadas del apartado gráfico. La clase **PanelBoleta** en concreto, ya que esta es la responsable de la información técnica del boleto.

El guardado de datos para los asientos ya comprados fue otro problema, el cual sólo es visible al momento de comprarlos, pero una vez fuera de la segunda pantalla ya no lo son. Esto en una futura ocasión se puede implementar añadiendo una tercera ventana con la información de los pasajes para tener una visión más gráfica de estos. Pero en nuestro caso, esto se resolvió generando un archivo **.txt** el cual representa nuestros boletos, simulando una página real de transporte la cual nos entrega un PDF con estos.