



# **Informe proyecto DOO: Sistema de reserva de asientos de autobús**

Grupo 9

Martin Tapia, Benjamín Martínez

Universidad de Concepción

Facultad de Ingeniería, Ingeniería Civil Informática

501250-1 DESARROLLO ORIENTADO A OBJETO

Profesor: Geoffrey Hecht

Concepción, Chile

7 de julio de 2024

#### 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)

Diagrama UML de clases:

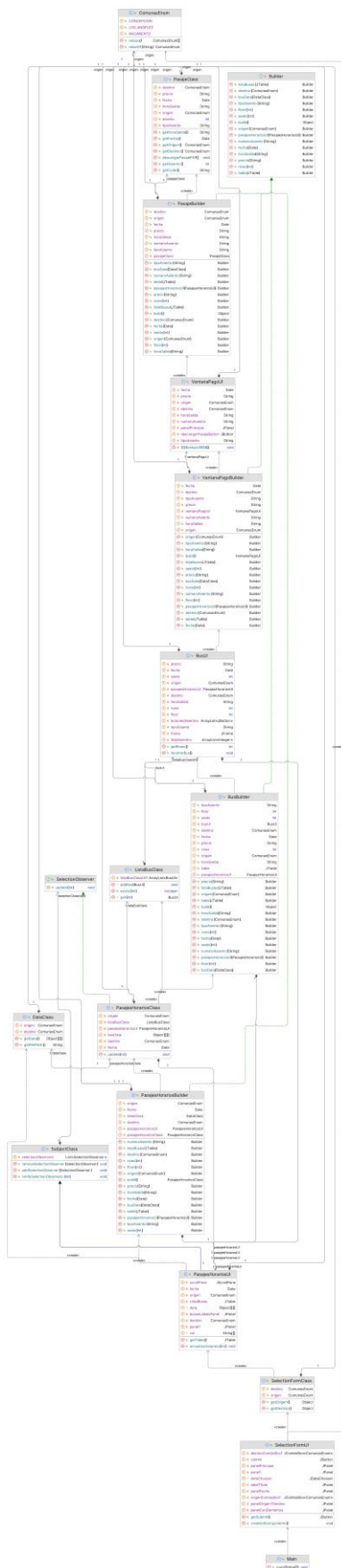
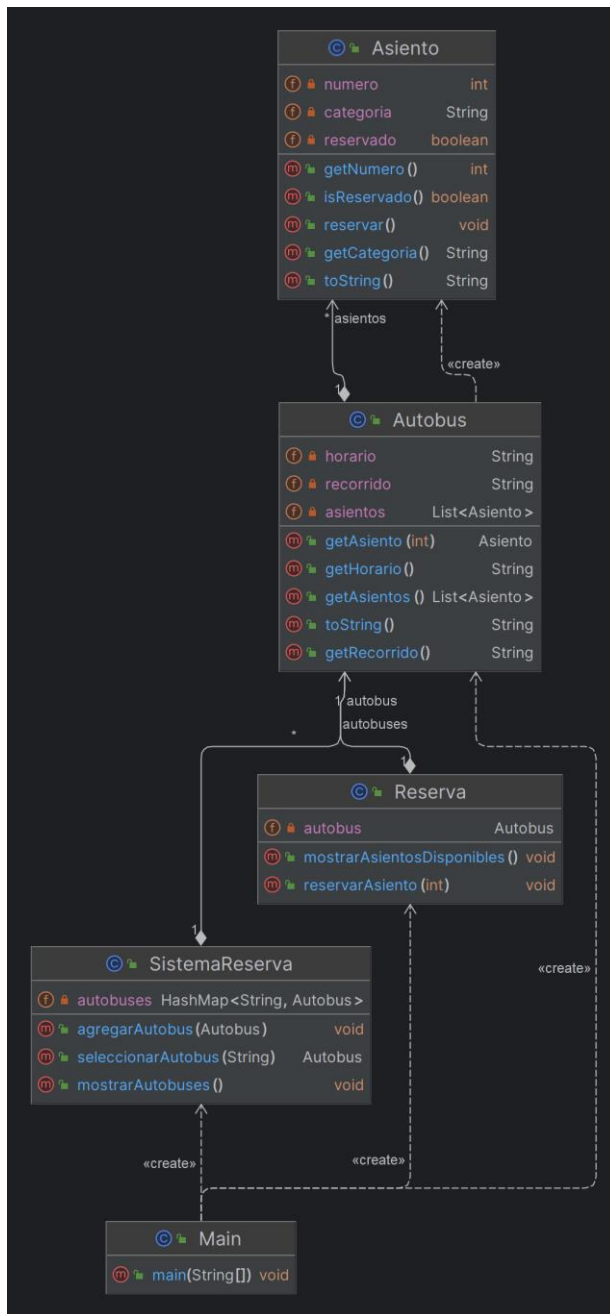


Diagrama UML de casos de uso:



Esta es una representación abstracta de nuestro programa para presentarse, básicamente nos dice que iniciando el código tenemos un sistema de reservas que permite mostrar los buses y seleccionar los buses, y que tienen funciones para mostrar los asientos de los buses y reservar sus asientos, las clases contienen a los buses o su información, los buses un horario, recorrido y una lista de asientos, esta lista con información asociada a ellos siendo su número, categoría y si ya están reservados o no.

Lista y justificación de los patrones utilizados:

A lo largo del proyecto en las ocasiones que usamos patrones usamos primero el patrón Builder debido a que este se asemejaba más a como sabíamos programar y ordenar nuestro código además de que según lo que encontramos en internet este patrón es recomendable para clases con muchos parámetros de entrada y el patrón Observer que nos ayudó con la separación lógica y grafica del programa

## Clases que usan patrón Builder

- Bus
- VentanaPago
- Pasaje

## Clases que usan el patrón Observer

- PasajesHorariosClass

## Capturas de pantalla de la interfaz:

El siguiente es un prototipo generado al ejecutarse el proyecto

des cómo hacer y que se debe incluir con indicando su avance (resuelto, por consultar, u otro) en li ones posterio

jo de la interf en ser dibujados de mu upadas por s elementos del Panel cen

arrollar grad nes y solicitando la opi

f a subir en C

bres

f. de la usen

no hacer y que se debe incluir con indicando su avance (resuelto, por consu

osterio

i interf en ser: por s element

r grad nes y si

ir en C

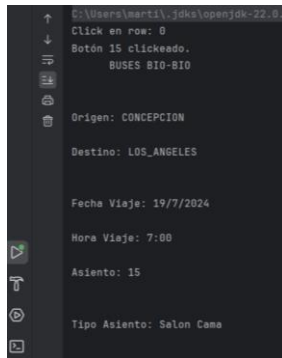
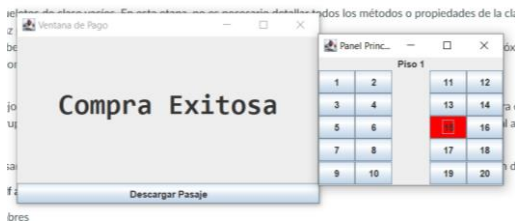
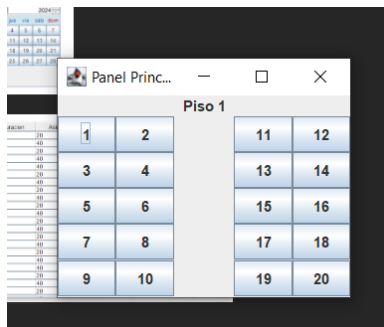
Bus

nes utilizados (pueden usar este si


man parte del patrón

in el contexto de su aplicación

Empresa	Salida	Llegada	Duración	Asientos	Precio	Piso	Tipo
Buses Bio Bio	7:00	8:30	1:30	20	\$6.000	1	Salon Cama
Buses Bio Bio	7:00	8:30	1:30	40	\$5.000	2	Semi Cama
Buses Bio Bio	8:00	9:30	1:30	20	\$6.000	1	Salon Cama
Buses Bio Bio	8:00	9:30	1:30	40	\$5.000	2	Semi Cama
Buses Bio Bio	8:30	10:00	1:30	40	\$4000	1	Clasico
Buses Bio Bio	9:00	10:30	1:30	20	\$6.000	1	Salon Cama
Buses Bio Bio	9:00	10:30	1:30	40	\$5.000	2	Semi Cama
Buses Bio Bio	10:00	11:30	1:30	20	\$6.000	1	Salon Cama
Buses Bio Bio	10:00	11:30	1:30	40	\$5.000	2	Semi Cama
Buses Bio Bio	11:00	12:30	1:30	20	\$6.000	1	Salon Cama
Buses Bio Bio	11:00	12:30	1:30	40	\$5.000	2	Semi Cama
Buses Bio Bio	12:00	13:30	1:30	20	\$6.000	1	Salon Cama
Buses Bio Bio	12:00	13:30	1:30	40	\$5.000	2	Semi Cama
Buses Bio Bio	13:00	14:30	1:30	20	\$6.000	1	Salon Cama
Buses Bio Bio	13:00	14:30	1:30	40	\$5.000	2	Semi Cama
Buses Bio Bio	14:00	15:30	1:30	20	\$6.000	1	Salon Cama
Buses Bio Bio	14:00	15:30	1:30	40	\$5.000	2	Semi Cama
Buses Bio Bio	14:30	16:00	1:30	40	\$4000	1	Clasico
Buses Bio Bio	15:00	16:30	1:30	20	\$6.000	1	Salon Cama
Buses Bio Bio	15:00	16:30	1:30	40	\$5.000	2	Semi Cama
Buses Bio Bio	16:00	17:30	1:30	20	\$6.000	1	Salon Cama



PASAJE BUS	
Origen:	CONCEPCION
Destino:	LOS_ANGELES
Fecha:	19/7/2024
Hora de Salida:	8:00
Asiento:	2
Tipo de Asiento:	Salon Cama
Precio:	\$6.000

  
33493481

Decisiones que se Tomaron a lo largo del proyecto:

Tuvimos que decidir el diseño de la aplicación esto no es solo el aspecto sino también la información presentada y recibida por el usuario y el cómo interactúan las distintas pantallas entre sí. No solo esto sino también tuvimos que pensar en el manejo de la información recibida y el cómo hacerla conectar entre todas las clases para tener un resultado coherente y exento de fallos que empeoren la experiencia de quien use el programa, un ejemplo de esto fue la decisión de que la fecha de viaje pueda ser entregada a través de un calendario interactivo en lugar de texto esto evitando errores de que se ingrese equivocadamente el formato en que se escribe la fecha (dd-mm-yyyy) además de hacerlo más intuitivo.

Problemas encontrados y que sirve de autocrítica:

El desconocimiento que ambos miembros teníamos sobre los patrones de diseño y el tema de la creación de los buses el cual solucionamos creando una 'pseudo' base de datos que nos permite almacenar la información de los viajes y que después siguió siendo usada en la creación de los asientos y pisos de los buses facilitándonos el trabajo.

## Cambios

Se mejoro la usabilidad del proyecto con mensajes de error y permite múltiples reservas (en diferentes autobuses).

Se le aplico el patrón de diseño Builder a varias clases del proyecto, estas siendo Bus, Pasajes, VentanaPago y PasajesHorariosClass

Se hizo que al descargar pasaje se cree un pdf, los cuales se pueden acumular.

Se cambiaron nombres a las clases y se separaron en varios package

Se hizo que los mensajes de error salgan como Pop-Ups

Se agrego el patrón Obsever a la clase PasajesHorariosClass

Se completo la separación lógica-gráfica