Facultad de Ingeniería
Universidad de Concepción
El origen de grandes soluciones

Implementación de interfaz para sistema de reserva de asientos de autobús

Profesor Geoffrey Hecht
Asignatura: Programación II 2023-2

Integrantes grupo 8 :

Martin Figueroa ~ Ingeniería civil informática

Ignacio Diaz ~ Ingeniería civil informática

Objetivo del proyecto

Para más completitud, se comentará el enunciado del proyecto así como también su objetivo.

Para el proyecto se nos pide crear un sistema de reservas de asientos, que permita a una empresa la venta de los asientos, así como también la reserva de los mismos. De esta manera se necesita visualizar una representación gráfica que simula una página web de compra de pasajes de autobús, pudiendo seleccionar el destino y origen del viaje. Dependiendo de si es un viaje solo de ida o de ida y vuelta, se permite seleccionar las fechas, horarios y tipo de bus tanto para el viaje de ida como para el de vuelta. Luego de esto se debe mostrar en pantalla la representación del bus seleccionado junto con sus asientos disponibles y sus distintos tipos para cada bus en particular, en este caso de 1 piso y de 2 pisos. Al momento de seleccionar un asiento este debe verificar su disponibilidad (de no estar disponible no se debe poder reservar) y mostrar en pantalla el precio a pagar. El pago no debe ser gestionado.

Diagrama de Casos de Uso

Aquí se puede ver el diagrama de los casos de uso del proyecto:

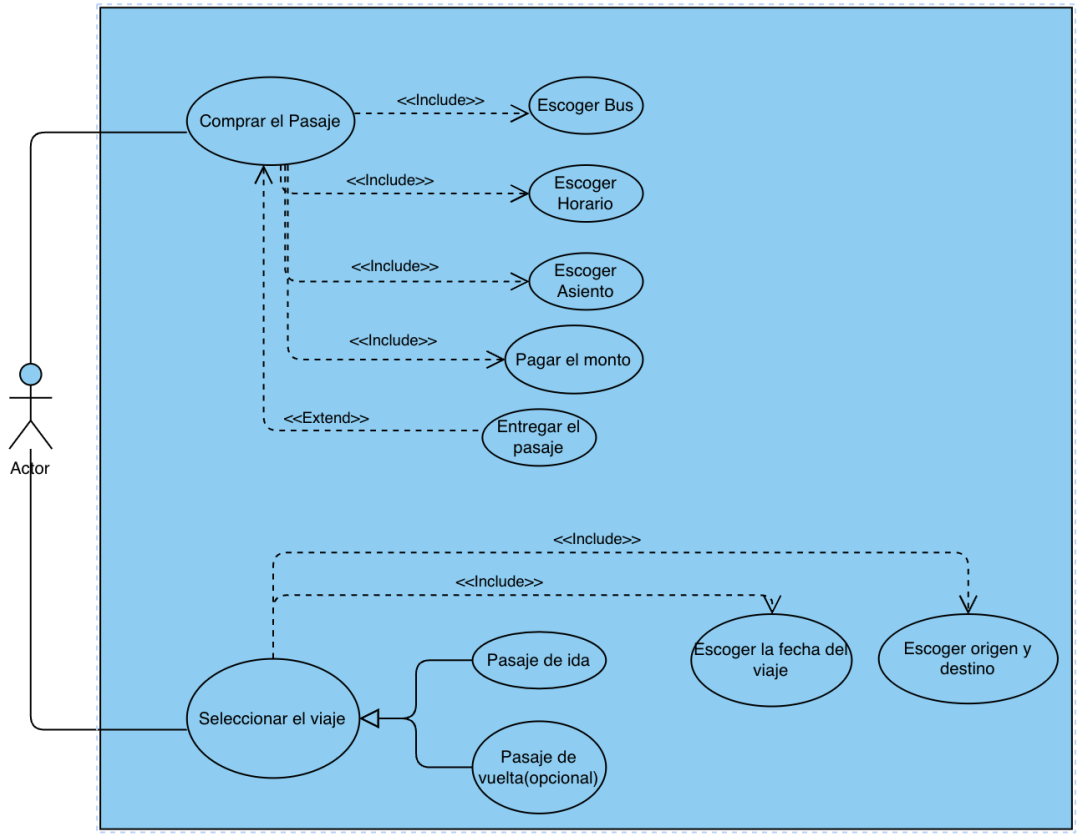
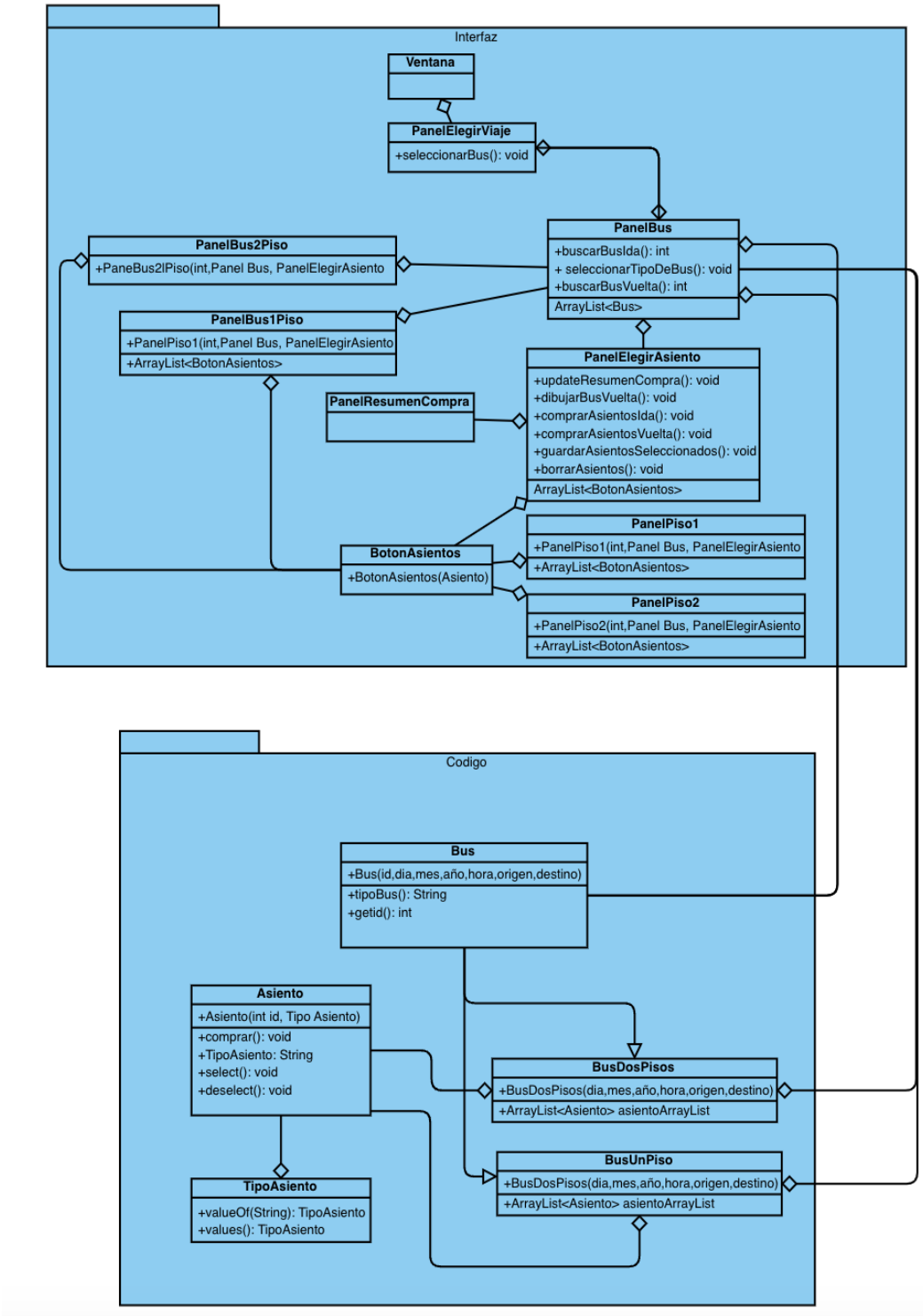


Diagrama UML



Captura de pantalla de la interfaz

Decisiones para la creación del sistema

Dentro de lo que fue la creación del proyecto se nos presentaron varias dificultades al momento de definir y pasar a código todo lo que teníamos en mente así que al principio la ayuda del referente fue necesaria.

Al momento de solo tener una interfaz sin funcionalidades, fuimos con la referente de nuestro proyecto comentando nuestras ideas y pidiéndole ayuda con lo que teníamos en mente para el proyecto. Por ejemplo en la creación de nuestro panel de botones que representan los asientos, en donde nos pudo guiar proponiendo ejemplos de cómo hacerlo. Algunos ejemplos de esto son: otorgarle atributos a la clase “Asiento” de seleccionado o comprado, crear un ArrayList de asientos y de buses. Esto nos fue de gran ayuda para entender el uso y generación de un panel de botones que luego pudimos implementar por nuestra cuenta.

En cuanto a la selección de nuestro patrón de diseño, al principio pensamos en usar el patrón Observer, ya que al momento de marcar un asiento que ya no está disponible, podríamos notificar al cliente el estado del asiento que quiere ser seleccionado. Sin embargo, por falta de información y

dificultad, decidimos usar el patrón Singleton para ocuparlo en el panel que muestra en pantalla el precio a pagar del asiento.

Luego de tener una idea de cómo se podría implementar todo esto, tuvimos que decidir cómo dividir el sistema en distintos paneles de manera coherente. Finalmente, decidimos tener 4 paneles principales. En el primer panel se seleccionará el recorrido del viaje y la fecha, además de si es de ida y de vuelta o solo de ida. Una vez obtenida esta información, se pasa al siguiente panel en el cúal se seleccionará el tipo de bus y el horario en que se desea viajar. Luego, el bus seleccionado es desplegado en el tercer panel. Este panel varía en función si es un bus de dos pisos y de uno solo. Decidimos diferenciar los distintos tipos de asientos según el color (Naranjo para los Semi-Cama y Rosado para los Sillón Cama). Además, los asientos ya comprados se visualizan en rojo . El panel final es el que muestra el resumen de la compra, siendo organizado por tipo de asiento y por viaje, mostrando tanto la información el bus/viaje como el precio unitario de cada tipo de asiento, el número de asientos seleccionado de cada tipo y el precio total.

Al presionar el botón “Comprar” en este último panel, los asientos seleccionados pasan a estar comprados. Se puede verificar el correcto funcionamiento de esta mecánica al seleccionar dos veces el mismo bus, donde la segunda vez ya no estarán disponibles los asientos que fueron comprados en la primera vez.

Sin embargo, tuvimos problemas a la hora de reiniciar los paneles luego de hacer una compra, en especial en el caso del bus de 2 pisos. Si bien se guardan los asientos comprados, no se logró el objetivo de visualizar los botones de los dos pisos para poder efectuar una nueva compra. Sabemos que esto es probablemente un error en el método refresh() de alguno de estos paneles.

La mayor parte del proceso de diseño fue realizado en Windows. En esta plataforma, como se puede apreciar en la captura de pantalla de la interfaz, no tuvimos inconveniente en cambiar los colores de los botones. Sin embargo, al probar la interfaz en el sistema operativo Mac, esto no funcionaba de la manera esperada. No tenemos del todo claro porque esto se produce, por lo que de ser posible, considere la plataforma de desarrollo.

Respecto al Patrón de Diseño

El patrón de diseño a utilizar como se nombró anteriormente es el Singleton, el cual consiste en la creación de una clase que garantice tener una única instancia y que se pueda acceder a esta desde cualquier parte del programa. Esto puede ser útil para proporcionar un punto de acceso global a una clase, que a su vez pueda ser actualizada de manera global.

A pesar de intentarlo desde un inicio, no logramos comprender bien cómo utilizarlo en esta aplicación en particular. Por lo mismo, decidimos por falta de tiempo no utilizarlo y usar el PanelResumenCompra sin este patrón de diseño, nos hacemos responsables de la falta del patrón de diseño pero también nos ayudó a poder terminar la implementación del resumen de la compra el cual era parte del objetivo del proyecto final.

Conclusión final y Autocrítica

A pesar de tener un fin de semestre un tanto difícil, logramos sacar adelante gran parte del proyecto gracias a el trabajo previo y las consultas respectivas tanto con el referente y el profesor.

Los problemas que tuvimos que enfrentar fueron en su totalidad el como crear un sistema de código, que partía desde las necesidades de nuestra interfaz primitiva creada las primeras semanas de trabajo, el tener gran parte de los botones pero no su lógica, que fue lo contrario a la tarea anterior, en donde la parte de la lógica ya estaba completa en su totalidad y luego era crear la interfaz, gracias a esto pudimos aprender bastante en el cómo trabajar en una situacion asi, por ello también nuestro diseño de interfaz es simple y deja de lado el atractivo visual que puede generar en el usuario, ya que tuvimos que enfocarnos más en la parte del código, más que en la parte visual, el cual pudo tener una mejoría si le hubiéramos dado más tiempo de estudio y trabajo, como también en la implementación de un patrón de diseño útil y funcional para el código, junto con las carencias en completar el objetivo del proyecto.