

Proyecto final

Programación II

Integrantes:	Mauro Muñoz (Grupo 11)
Fecha:	10/12/23
Tarea:	Proyecto final
Ramo:	Programación II
Profesor:	Geoffrey Hecht

Introducción

Este trabajo nos ayudara a ver que la optimización de procesos y la mejora continua de servicios son elementos clave para proporcionar experiencias eficientes y cómodas a los usuarios. En este contexto, se presenta el informe sobre el sistema de reserva de asientos de autobús, una solución tecnológica diseñada para facilitar la planificación y gestión de viajes en autobús, ofreciendo a los pasajeros la posibilidad de seleccionar asientos de manera personalizada.

Este sistema aborda la diversidad de necesidades de los usuarios, brindando opciones de asientos como "asiento normal", "asiento premium" y "asiento vip", el programa busca no solo simplificar la reserva de viajes, sino también elevar la experiencia del pasajero al proporcionar un servicio diferenciado y adaptado a sus preferencias.

La flexibilidad del sistema permite a los usuarios elegir entre distintas categorías de asientos, adaptándose a sus necesidades específicas de comodidad y presupuesto. Desde opciones más asequibles hasta experiencias de viaje de lujo, el sistema ofrece una gama completa de elecciones para satisfacer los diversos perfiles de pasajeros.

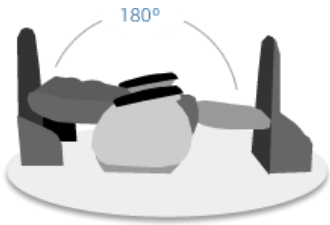
A lo largo de este informe, exploraremos en detalle las características clave del sistema, su arquitectura técnica, y cómo aborda los retos y oportunidades asociados con la reserva de asientos en el contexto de viajes en autobús. Además, analizaremos los beneficios que esta solución aporta tanto a los pasajeros como a las empresas de transporte, destacando la eficiencia operativa, la mejora en la calidad del servicio y la adaptabilidad a las demandas cambiantes del mercado.

Este informe proporcionará una visión completa del sistema de reserva de asientos de autobús, destacando su contribución a la modernización y mejora de la experiencia de viaje en el ámbito del transporte terrestre.

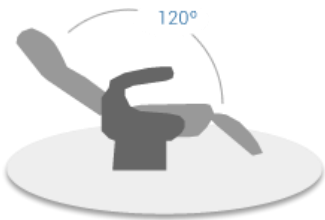
Imágenes utilizadas



Para asientos normales



Para asientos vip



Para asientos premium



Para asientos ocupados

UML

GUI

Panel (extend JPanel)

- PanelRecorrido pr
- SelectorBus bus
- BusDisponible busDisponible
- + panel()
- + void paint()

Ventana (extends JFrame)

- Panel P
- int escala
- Comprador comprador
- BusDisponible buses
- + Ventana()
- + void crearGUI()

PanelAsientosBus (extends JPanel)

- PanelAsientos asientos1
- PanelAsientos asientos2
- JButton confirmar
- class ConfirmarListener implements ActionListener
- + PanelAsientosBus()

PanelAsientos (Extend JPanel)

- JButton[] botones
- Bus busElegido
- Comprador comprador
- class BotonListener implements ActionListener
- + PanelAsientos()
- + void paint()

PanelRecorrido (extend JPanel implements ActionListener)

- JComboBox<Recorridos> origenComboBox
- JComboBox<Recorridos> destinoComboBox
- JComboBox<String> horarioComboBox
- JButton buscar
- Comprador comprador
- BusDisponible busDisponible
- SelectorBus selectorBus
- + PanelRecorrido()
- + void actionPerformed()

SelectorBus (extend JPanel implements ActionListener)

- JComboBox<String> comboBox
- JButton button
- BusDisponible busDisponible
- Comprador comprador
- Bus busElegido
- + SelectorBus()
- + void actionPerformed()
- + void actualizarBuses()

Exceptions

AsientoNoDisponibleExcepcion (extends Exception)

- + AsientoNoDisponibleExcepcion()

NoHayTicketExcepcion (extends Exception)

- + NoHayTicketExcepcion()

PagoInsuficienteExcepcion (extends Exception)

- + PagoInsuficienteExcepcion()

PagoIncorrectoExcepcion (extends Exception)

- + PagoIncorrectoExcepcion()

YaComproExcepcion (extends Exception)

- + YaComproExcepcion()

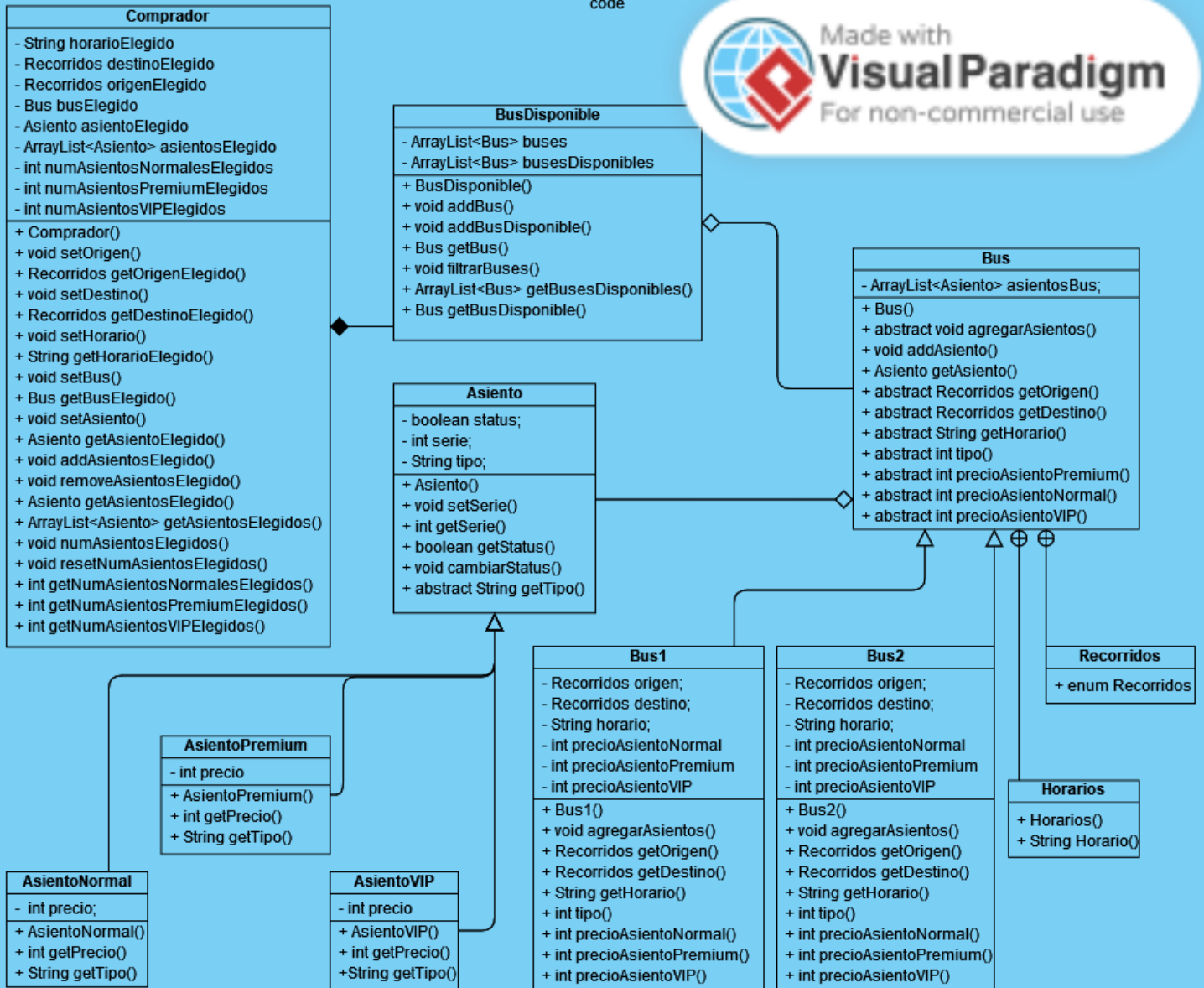
NoHayAsientoExcepcion (extends Exception)

- + NoHayAsientoExcepcion()

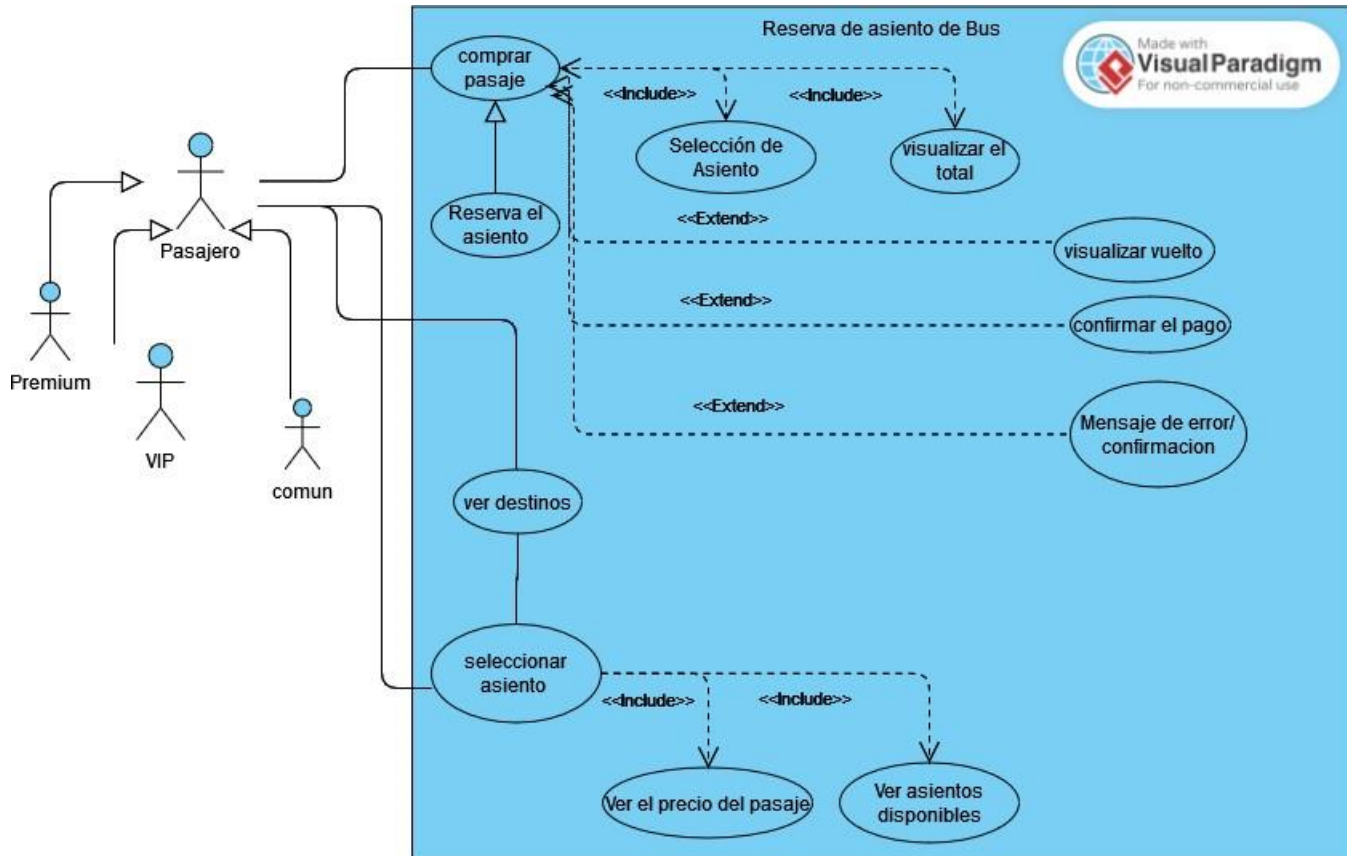
code



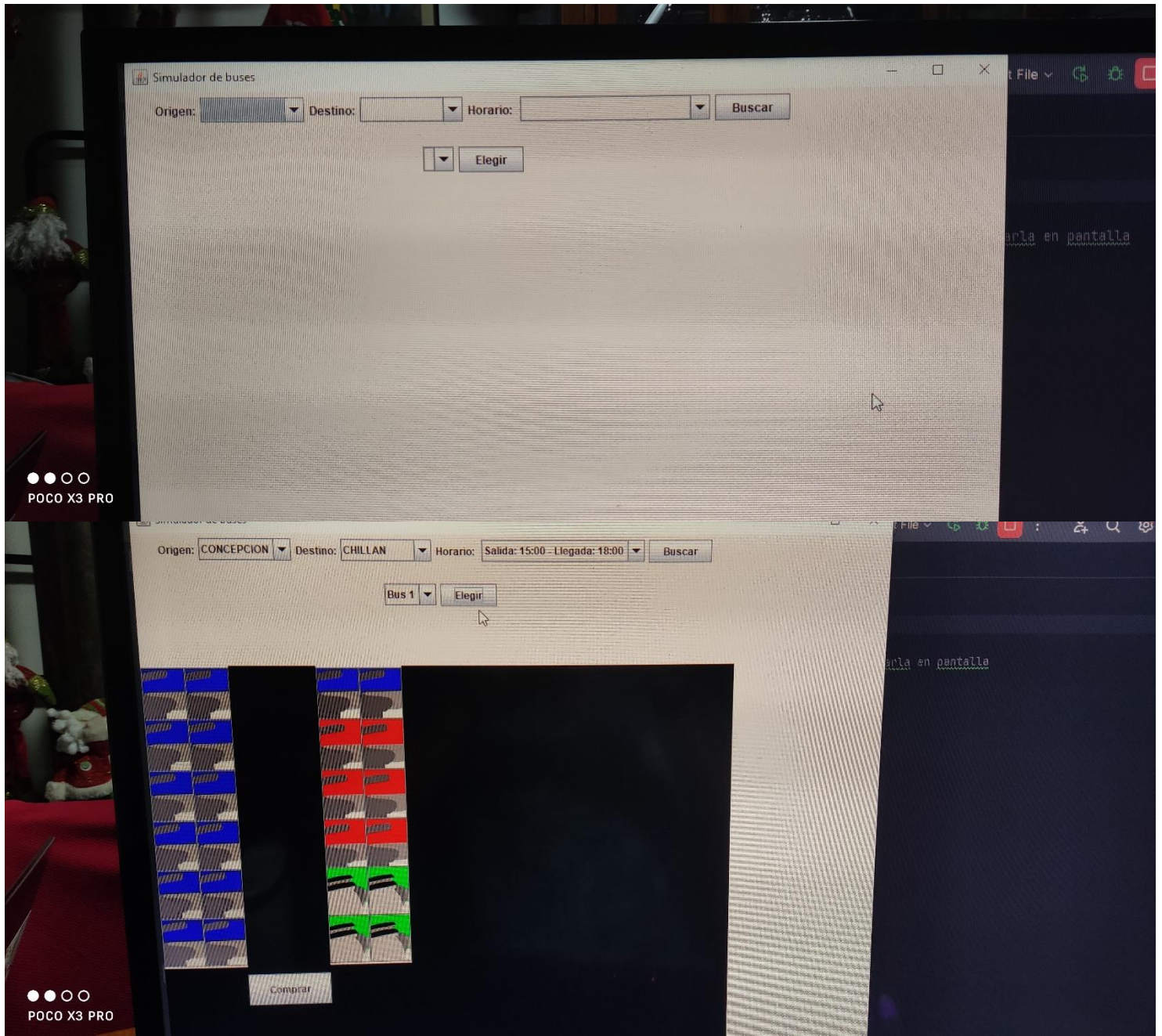
Made with
Visual Paradigm
For non-commercial use

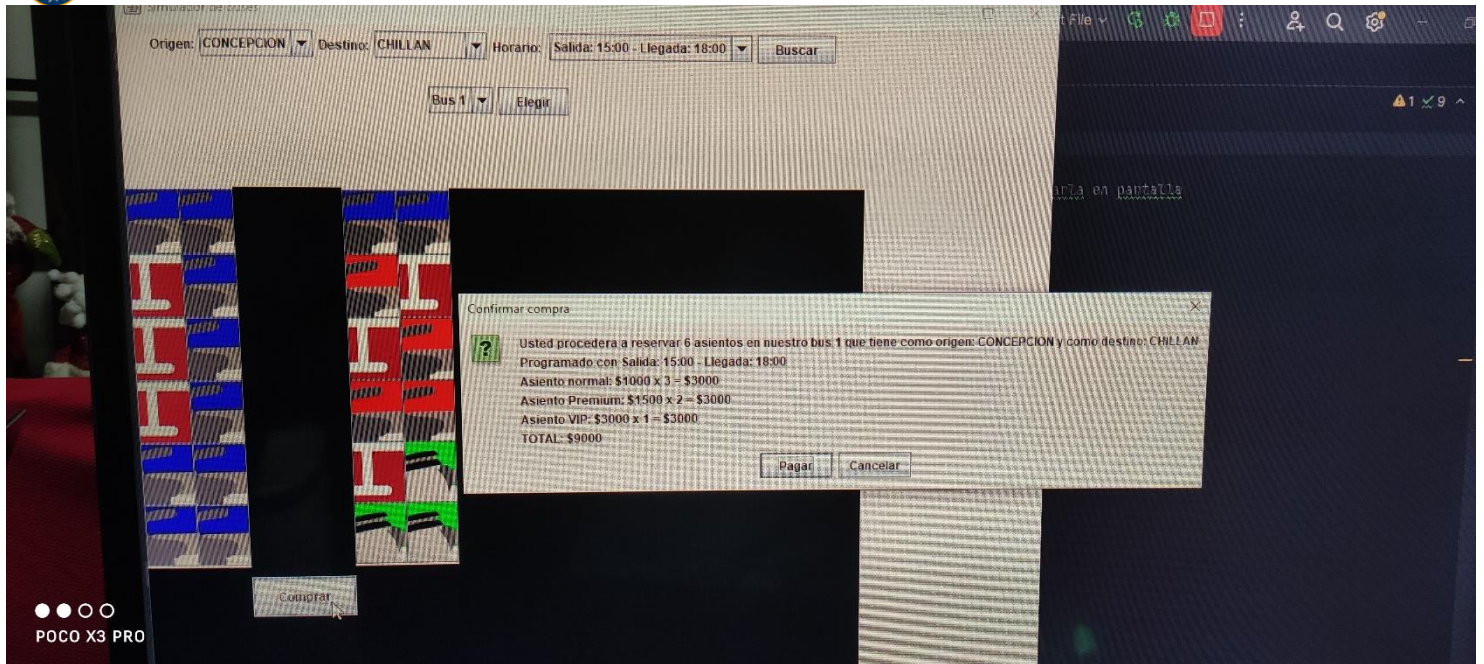


usecase



Interfaz programa





Patrones utilizados

- Patrón Observer: Este patrón fue utilizado porque teníamos la necesidad de crear una dependencia entre la clase “SelectorBus”, la cual actúa como el observador para recibir actualizaciones del objeto “BusDisponible”.
- Patrón Singleton: Se utilizó para que hubiera una única de las clases “Pasajero” y “BusDisponible”.
- Patrón Composite: Empleado con la finalidad de tratar un grupo de objetos como si fueran un solo objeto, a la vez que se establece una jerarquía entre ellos. En nuestro caso en la clase “Ventana”, se agregan varios componentes (“PanelRecorrido”, “Panel”, “SelectorBus”) al panel principal “Panel”.
- Patrón Abstract Factory: Utilizado para crear familias de objetos, en nuestro caso las clases Asiento y Bus;
- Patrón Iterator: Al momento de recorrer los Arraylist para facilitar el almacenamiento de los objetos “Asientos”, que simulaban los asientos de un Bus.

Nuestro planteamiento para el desarrollo del programa consiste en que el usuario pudiera elegir a través de la interfaz gráfica un bus y asientos que ya estuvieran creados, por lo cual decidimos crear las instancia a estos objetos al inicio del programa.

Luego en la interfaz gráfica el usuario puede filtrar por origen, destino, horario y tipo de bus, para seleccionar el asiento que más le convenga.

Al momento de ya tener hecha su selección hay un botón que desplegará un aviso en la pantalla para confirmar los datos del asiento y de los bus seleccionados, a su vez informa de los valores implicados y el total a pagar. Es aquí donde el usuario puede pagar o si bien lo desea, cancelar la operación, pudiendo modificar los parámetros que considere pertinentes.

Durante el desarrollo del proyecto tuve una serie de problemas a la hora de implementar el código, partiendo por el hecho de que mi compañero nunca más me contesto y tuve que hacerlo solo pero si hablamos de lo que me costó en particular diría que en especial es la parte gráfica, en donde hasta la redacción del presente informe hay problemas que no pudimos resolver, como por ejemplo, no logramos que la interfaz se recargara correctamente, lo cual impide ver los cambios al momento de reservar un asiento. De igual manera ocurre al momento de querer realizar una segunda reserva, ya que no pudimos hacer que las variables implicadas se reiniciaran para realizar múltiples reservas.

A pesar de lo anterior, se intento llevar a cabo lo solicitado de la mejor manera posible, dando lugar a nuestro programa y el presente informe.

conclusión

para finalizar este trabajo y hacer una introspección podemos decir que logramos poner en práctica los contenidos aprendidos durante el transcurso del semestre en la elaboración de este programa, siendo uno de estos el uso de github el cual resultó de gran utilidad al momento de controlar las versiones del programa.

A raíz de que hubieron problemas que no logramos solucionar, no hemos quedado del todo conforme con el resultado obtenido, ya que teníamos varias ideas que por diversas razones no pudimos implementar, lo cual me deja con un amargo sabor de boca ya que lo ideal era realizar un programa que cumpliera todas las expectativas y lograra desarrollarse realmente como lo esperaba pero al fin si pudimos lograr al menos algo que reflejara de alguna forma los conocimientos aprendidos en este curso haciendo este proyecto con relación a la gestión de asientos en viajes de autobús.