TUP - PROGRAMACIÓN Y LABORATORIO 3

AEROTAXI

Integrantes:

* Ciro López
* Sebastián Poo
* Maximiliano Fernández
* Leandro Lorenzo

Resumen de la aplicación:

Es una aplicación destinada a la operación de un sistema que vincula a un cliente con una empresa de Aero Taxi. En dicho programa, el cliente puede elegir destino, fechas, pasajes y seleccionar el avión que más le conviene para realizar el vuelo. También puede dar de baja pasajes ya adquiridos.

Informe Técnico

Funcionamiento del Sistema:

* El programa está desarrollado en JAVA.
* Los pasajeros y los tickets persisten en JSON.
* La visualización es a través de la terminal de JAVA

Decisiones tomadas para el diseño:

* Se analizaron los diferentes tipos de aviones con sus características específicas para los vuelos requeridos en la propuesta de Cátedra.
* Se tuvo en cuenta las distancias de los destinos para poder hacer las diferentes combinaciones de vuelos.
* Los tiempos de vuelo según las características de los aviones.

Clases utilizadas y sus relaciones:

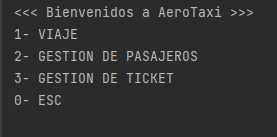
Para definir las clases, se crearon diferentes carpetas cuyos nombres sintetizan el tipo de clases que iban a alojarse dentro de las mismas.

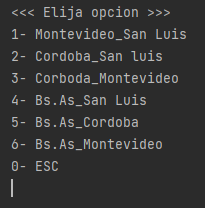
* Crud: CRUD es el acrónimo de "Crear, Leer, Actualizar y Borrar" (del original en inglés: Create, Read, Update and Delete). Dentro de la carpeta CRUD, está relacionada con las clases de archivos FileManagment y la clase de Pasajeros Passenger.
* FileManagment: dentro de la carpeta File, es la clase que gestiona la implementación de archivos con librerías Gson y llama a las excepciones try catch.
* FolderPlane es la carpeta que contiene los diferentes modelos de aviones segmentados en Clases con sus atributos correspondientes. Esta carpeta cuenta con dos clases abstractas, Gestión y Plane, y una Interfaz de servicio, Service que se implementará, según lo requieran, las diferentes naves. Las siguientes Clases corresponden a las distintas aeronaves, Bronze, Silver y Gold. Ellas extienden de Plane, abstract, atributos comunes que comparten los Aerotaxis.
* Menú: es la carpeta que contiene la clase Menú, quien es llamada desde el main para ejecutar el sistema y sus opciones.
* Passenger: es la Clase que implementa serialización para que sus objetos sean invocados remotamente y persistan.
* Person: es la carpeta que contiene la Clase abstracta Person y la Clase Passenger. Person contiene atributos comunes de personas y Passenger, extiende de Person.
* Ticket: es la carpeta que contiene las Clases Fechas y Ticket. La Clase Fechas tiene la particularidad de utilizar la librería de Calendar. La Clase Ticket, se relaciona con la mayoría de las Clases del sistema, Crud, File.FileManagment, FolderPlane.Gestion, FolderPlane.Plane, Menu.Menu, Passenger.Passenger y Travel.Distance.
* Travel: es la carpeta que contiene la Clase Enum Distances y la Clase Travel. La Clase Travel se relaciona con FolderPlane.Plane, Person.Person y Ticket.Ticket.
* Main: es la Clase principal que llama al Menú para ejecutar al programa.

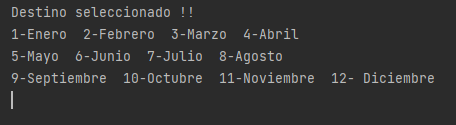
Diario de trabajo:

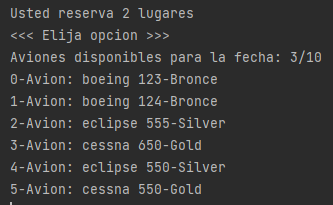
* Para llevar adelante la comunicación de las tareas concernientes al TP, se utilizaron grupos creados en plataformas tales como GitHub, Discord y WhatsApp.
* Desde el 16 de mayo que se crearon los canales de comunicación entre el grupo, se asignaron las tareas y fuimos trabajando en base lo conversado para el proyecto.

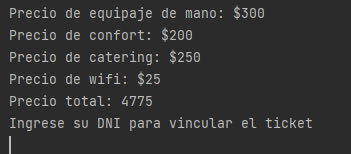
Manual de Usuario

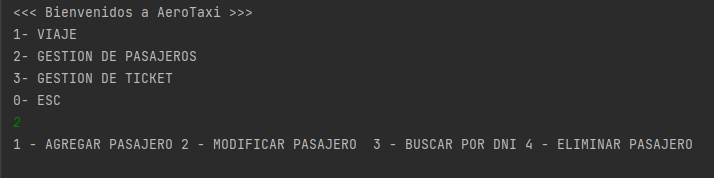
Pantalla inicial. Encontraremos un menú inicial para acceder a la adquisición de un ticket para efectuar nuestro vuelo.

 En la primer opción, Viaje, seleccionaremos el destino de nuestro vuelo.

 Una vez seleccionado nuestro destino, debemos seleccionar la fecha de nuestro interés y la cantidad de acompañantes.

Seguido, el sistema nos muestra las aeronaves disponibles para esa fecha.

Nos detalla el precio parcial del servicio que seleccionamos y el total. Una vez que nos informa esto, debemos vincular el ticket con el DNI del usuario y aceptar la compra si estamos de acuerdo.

En la segunda opción del Menú, podemos agregar, modificar, buscar y eliminar pasajero del sistema. 

En la tercera opción, Gestión de Ticket,