

Proyecto programado #2

Nicole de los Ángeles Araya Ballesterero

EIF 201 - Programación I

NRC 50005 - Grupo 03

Ingeniería en Sistemas, Universidad Nacional de Costa Rica

Profesora Karol Leitón A.

21 de noviembre del 2021

Objetivos del programa

Lograr crear un programa de venta de boletos para una Aerolínea, en la cual se inserta varias clases que permiten su buen funcionamiento. Y así el usuario pueda lograr hacer lo que necesite para su buen uso.

Limitaciones del programa

- Solo se puede comprar un boleto a la vez.
- En las opciones de crear, al entrar, deberá crear lo que piden para poder salir.
- El menú se cae si escribe letras en vez de números cuando le dice que seleccione una opción.
- A la hora de ingresar un Vuelo, si hay un vuelo con un avión que sale el 30/11/2021 y sale a las 13:00 y llega a las 15:00, y después se agrega otro vuelo con la misma fecha y avión, pero sale a las 14:00, el programa va a permitir su ingreso. Sucede igual al ingresar Tripulantes a un Vuelo.
- Al ingresar Tripulantes en un Vuelo, y si no se han creado las suficientes personas, va a existir un bucle, ya que el programa no va a permitir que repita un ID y tampoco que ponga uno que no exista, tampoco que ponga un ID si hay un choque de horario con otro vuelo con ese mismo ID. También cuando ingresa un piloto en la opción Crear Vuelo, si un piloto tiene un choque de horarios o no existe su ID, va a ver un bucle.
- Solo existe un nombre de usuario y contraseña para el Menú Administrador.
- El programa no guarda información de la Lista de Vuelos.

Alcances del programa

- El menú esta dividido para Administrador y Cliente, si desea ingresar al menú de administrador deberá ingresar el nombre de usuario (que es admi2021) y la contraseña (que es **1234**).
- Al ingresar exitosamente en el menú administrador, se cargarán todos los archivos de la Lista Ruta, Lista Avión, Lista Pilotos, Lista Sobrecargo (que son los datos “permanentes” en la Aerolínea) y estarán disponibles los datos para crear Vuelos con esos datos si así se quisiera.
- El programa no permite que haya un vuelo con el mismo avión y misma hora de salida.
- El programa no permite que haya un choque de horarios de salida con otro vuelo de una persona ya existente ingresada como Piloto, Copilo o Sobrecargo.
- El programa no permite ingresar un tripulación a un vuelo que ya tiene una.
- El programa no permite que exista un Código o ID repetido en una misma clase.
- El programa muestra toda la información que requiera la persona para consultar dependiendo del menú en el que este, para mantener la confidencialidad de algunos datos que solo el administrador debería ver.
- En Venta de Boletos, el programa no permite que pueda comprar un boleto si ya no hay disponibilidad de asientos, igual el Cliente puede fijarse en la opción Detalle de Vuelo Especifico, cuantos campos disponibles quedan o igual verlo a la hora de comprar el boleto.
- En la opción de Venta de Boletos, al elegir el vuelo que se quiere comprar el boleto, si ya no hay disponibilidad no va a entrar para realizar el proceso de compra.
- Se puede guardar nueva información creada en el menú Administrador.
- Si no existen los nombres de los archivos aun, donde guardara la información deseada, el programa se encarga de crearlos para su buen funcionamiento.

Descripción de la solución planteada

Justificación de clases involucradas

- La clase Ruta y la clase Avión contienen sus atributos y funciones específicos.
- La clase Persona se creó para que herede su información a la clase Pasajero y Tripulante, y la clase Tripulación al igual heredera su información a la clase Piloto y Sobrecargo.
- Clase Vuelo contienen sus atributos específicos como punteros a otras clases y sus funciones específicas.
- La Lista Pasajeros guarda pasajeros de cada diferente vuelo.
- Se creo Lista Avión, Lista Ruta, Lista Piloto, Lista Sobrecargo y Lista Vuelo para que tenga una cadena de vectores de sus atributos asignados.
- Clase Archivos Aerolínea se encarga de leer y cargar la información.
- Clase Aerolínea es la controladora.

Descripción de la funcionalidad y objetivo de cada clase

- Ruta: Tiene todos los atributos para la información de Ruta y sus métodos que devuelven o cambia esos atributos.
- Avión: Tiene todos los atributos para la información de Avión y sus métodos que devuelven o cambia esos atributos. También tiene vectores de asiento, que se encarga de guardar los asientos ocupados.
- Persona: Clase que heredera sus atributos y métodos a las clases hijos.
- Pasajero: Tiene todos los atributos para la información de Pasajero y sus métodos que devuelven o cambia esos atributos.

- Tripulación: Clase que heredera sus atributos y métodos a las clases hijos.
- Piloto: Tiene todos los atributos para la información de Piloto y sus métodos que devuelven o cambia esos atributos.
- Sobrecargo: Tiene todos los atributos para la información de Sobrecargo y sus métodos que devuelven o cambia esos atributos.
- Vuelo: Tiene todos los atributos para la información de Vuelo y con ello, contiene punteros a Ruta, Avión, Lista Pasajero, Piloto y Sobrecargo. Y sus métodos que devuelven o cambia esos atributos.
- Lista Pasajero: Contiene una colección de pasajeros que se relacionan con un Vuelo. En donde se puede crear finitos pasajeros hasta donde lo permita la clase Vuelo. Sus métodos hacen todo lo que permita un buen manejo de su información.
- Lista Avión, Piloto, Sobrecargo, Ruta, Vuelo: Son colecciones que pueden contener información de aviones, rutas, pilotos, sobrecargos, vuelos sin tener un tamaño limite. Sus métodos hacen todo lo que permita un buen manejo de su información.
- Archivos Aerolínea: No contiene atributos, por lo que sus métodos son statics, que se encargan de leer la información y cargarla según su método.
- Aerolínea: Contiene punteros a las todas las Listas excepto Lista Pasajero, y se encarga de crear, mostrar y guardar la información. Sus métodos son la controladora, por lo que son menús.
- main: Donde se ejecuta el programa.

Descripción relaciones incluidas

- Clase Vuelo tiene relación con una clase Avión, Lista Pasajeros, Piloto, Sobrecargo y Ruta.
- La clase Persona hereda a Pasajeros y Tripulación.
- La Tripulación hereda a Piloto y Sobrecargo.
- Aerolínea tiene relación con las Listas excepto Lista Pasajero y a Archivos Aerolínea.

Descripción del uso de listas

Las clases Listas ayudaran a tener un ordenado las clases que se relacionen a ellas y acceso a toda la información ingresada, al tener todos los métodos que se basen en sus Nodos, y hacer un recorrido más rápido a la hora de programar.

- Utilidad Lista Pasajero:
 - Tiene métodos que devuelven si la lista está vacía.
 - Tiene métodos que devuelven la búsqueda de un pasajero.
 - Tiene los métodos para insertar en la lista pasajero.
 - Tiene un método que imprime la información que contiene.
- Utilidad Lista Avión:
 - Tiene métodos que devuelven si la lista está vacía.
 - Tiene métodos que devuelven la búsqueda de un avión.
 - Tiene los métodos para insertar en la lista avión.
 - Tiene un método que imprime la información que contiene.
 - Tiene un método guardar.
- Utilidad Lista Ruta:
 - Tiene métodos que devuelven si la lista está vacía.
 - Tiene métodos que devuelven la búsqueda de una ruta.
 - Tiene los métodos para insertar en la lista ruta.
 - Tiene un método que imprime la información que contiene.
 - Tiene un método guardar.
- Utilidad Lista Sobrecargo:
 - Tiene métodos que devuelven si la lista está vacía.
 - Tiene métodos que devuelven la búsqueda de un sobrecargo.
 - Tiene los métodos para insertar en la lista sobrecargo.
 - Tiene un método que imprime la información que contiene.
 - Tiene un método guardar.
- Utilidad Lista Piloto:
 - Tiene métodos que devuelven si la lista está vacía.
 - Tiene métodos que devuelven la búsqueda de un piloto.
 - Tiene los métodos para insertar en la lista piloto.
 - Tiene un método que imprime la información que contiene.

- Tiene un método guardar.
- Utilidad Lista Vuelo:
 - Tiene métodos que devuelven si la lista está vacía.
 - Tiene métodos que devuelven la búsqueda de un vuelo.
 - Tiene los métodos para insertar en la lista vuelo.
 - Tiene un método que imprime la información que contiene.
 - Tiene un método que imprime la información detallada de un vuelo.
 - Tiene un método que compara para que no exista un mismo avión repetido a una misma fecha y hora de un Vuelo.
 - Tiene un método que compara para que no exista sobrecargos repetidos a una misma fecha y hora de un Vuelo.
 - Tiene un método que compara para que no exista copilotos repetidos a una misma fecha y hora de un Vuelo.
 - Tiene un método que compara para que no exista pilotos repetidos a una misma fecha y hora de un Vuelo.
 - Tiene un método que muestra información de los asientos de un vuelo en específico ingresado, pasa ver su capacidad y lista de pasajeros.