

## **Manual técnico**

### **Clases Ventana:**

Estas clases contienen como atributos todos los componentes que se utilizan en la interfaz como es el caso de JLabel, JButton, JPanel, JTextField, JTextArea, JComboBox, JRadioButton, etc.

### **Métodos:**

Contiene métodos constructores para la inicialización de componentes, getters para los manejadores y métodos para colocar los componentes en la interfaz gráfica.

### **Clases Manejador:**

Estas clases contienen toda la lógica que será utilizada en las ventanas. Contienen como atributo el nombre de la ventana a la que está asociada esta clase.

### **Métodos:**

Contiene el método constructor para inicializar su único atributo, distintos métodos vacíos dependiendo de la ventana, puede contener métodos de compra, búsqueda, actualización, listar datos, dependiendo de la ventana.

### **Clases Archivo:**

Clases sin atributos utilizadas para la carga de datos a archivos binarios. Los datos son: Empleado, Pasaporte, Tarjeta, Avión, Vuelo, AeroPuerto, Aerolinea, entre otros.

### **Métodos:**

Cada clase contiene únicamente dos métodos estáticos: guardarDato y leerDato. Dependiendo de la clase que sea será la variación de nombre, La Clase ArchivoPasaporte cuenta con un método extra que es el método RenovarPasaporte. Sirve para cambiar la fecha de vencimiento del pasaporte enviado.

### **Clase GenerarCodigo:**

Clase sin atributos utilizada para generar una cadena de texto de forma aleatoria para contraseñas, usuarios, códigos, etc.

### **Métodos:**

Contiene dos métodos, un público y uno privado, el método público estático generarCodigo es el encargado de agregar números aleatorios y retorna la cadena de texto, el método privado agregarLetra es llamado por el método anterior y agrega letras con un número aleatorio y retorna la cadena.

### **Clase Constante:**

Clase que no contiene ningún método y contiene todas las constantes usadas en el programa.

### **Enums:**

Contiene los distintos enums utilizados en el programa, enums para Sexo, Estado Civil, Estado del vuelo, estado del asiento. No contiene ningún método.

### **Clase HiloCarga:**

Método que hereda de Thread encargado de la carga de datos. Como atributo contiene un objeto tipo File que es el archivo que se procesa y un JFrame para hacer referencia al espacio en memoria de la ventana de carga.

#### **Métodos:**

Contiene el método heredado run que es el que pone a correr el método vacío leerArchivo.

El método leerArchivo mediante un ciclo va recorriendo el archivo que fue enviado y verifica qué tipo de dato es para hacer las respectivas verificaciones y la carga del dato de la línea que está siendo leída.

### **Clase Verificaciones:**

Clase que no contiene ningún atributo y solamente métodos estáticos encargada de verificar la información antes de cargarla a un archivo binario.

#### **Métodos:**

Contiene una gran cantidad de métodos para verificar la información:

MÉTODO aeroPuertoExistente: encargado de retornar true o false si el aeropuerto ya existe.

MÉTODO aeroLineaExistente: retorna true o false si la aerolínea existe en el aeropuerto solicitado.

MÉTODO verificarVuelo: retorna true si el vuelo se encuentra en espera y false de lo contrario.

MÉTODO verificarTarjeta: retorna true si la tarjeta ha sido encontrada y false de lo contrario.

MÉTODO gerenteConAeroLinea: retorna true si un gerente ya cuenta con la aerolínea y false de lo contrario.

MÉTODO verificarAvion: retorna true si el avión ha sido encontrado y false de lo contrario.

MÉTODO verificarDistancia: retorna true si se ha encontrado la distancia y false de lo contrario.

MÉTODO verificarGasolina: retorna true si la cantidad de gasolina del avión es la suficiente para recorrer la distancia solicitada y false de lo contrario.

MÉTODO verificarPasaporte: retorna true si el pasaporte ya existe y false de lo contrario.

MÉTODO verificarPasaporteVuelo: retorna true si el pasaporte enviado cuenta con un vuelo EN ESPERA y false de lo contrario.

MÉTODO verificarVigencia: retorna true si el pasaporte se encuentra vigente en una fecha en específico y false de lo contrario.

### **Clase Objeto:**

Clase padre que no contiene ningún dato o método.

**Clase Asiento:**

Clase encargada de los asientos del avión, posee dos atributos, noAsiento y EstadoAsiento.

**Métodos:**

Método constructor, getters y setters.

**Clase Pasillo:**

Clase vacía utilizada para instancias solamente.

**Clase Avion:**

Clase encargada de la creación de aviones, posee distintos atributos como:

Datos tipo String: nombreAeroLinea, aeroPuertoActual.

Datos tipo int: codigoAvion, capacidadPasajeros:

Datos tipo double: gasolinaConsumida, capacidadGasolina, consumoPorMilla.

Datos tipo Objeto: (matriz) objetos.

**Métodos:**

Contiene métodos constructores, getters y setters y métodos generarMatriz encargados de crear la matriz de asientos y pasillos dependiendo de las filas y columnas solicitadas por el usuario.

**Clase AeroLinea:**

Clase encargada de la creación de aerolíneas, contiene dos atributos, nombreAerolinea, nombreAeropuerto. Ambos de tipo String.

**Métodos:**

contiene métodos constructores y métodos getters.

**Clase AeroPuerto:**

Clase encargada de la creación de aeropuertos, Contiene tres atributos, nombreAeroPuerto, ciudad y país, todos de tipo String.

**Métodos:**

contiene métodos constructores y métodos getters.

**Clase Distancia:**

Clase encargada de la creación de distancias, contiene dos atributos de tipo String (nombreAeroPuertoOrigen y nombreAeroPuertoDestino) y un atributo tipo double cantidadMillas.

**Métodos:**

Contiene un método constructor y métodos getter.

**Clase Vuelo:**

Clase encargada de la creación de vuelos. Esta clase implementa la interfaz Runnable para el hilo de vuelos. Contiene una gran cantidad de atributos para el código del vuelo, código del avión, el número de pasajeros, el nombre del aeropuerto de origen y destino, entre otros.

**Métodos:**

Esta clase tiene métodos constructores, getters, setters. Un método llamado asignarEtiqueta que hace referencia a la etiqueta para el diseño visual de los vuelos, el método run encargado de realizar el vuelo, y métodos para cambiar el país en el que se encuentran todos los pasaportes que viajaron en el vuelo así como cambiar el aeropuerto en el que se encuentra el avión después del vuelo.

**Clase Persona:**

Clase encargada de la creación de Personas. Esta clase cuenta con dos atributos, el nombre y el apellido de tipo String.

**Métodos:**

Método constructor y métodos getter.

**Clase Empleado**

Clase encargada de la creación de empleados, esta clase hereda de Persona y tiene dos atributos propios, usuario y contraseña.

**Métodos:**

Método constructor y métodos getter.

**Clase Administrador:**

Clase encargada de la creación de administradores, esta clase hereda de Empleado. Posee un atributo propio que es el nombre del aeropuerto que se encuentra administrando.

**Métodos:**

Método constructor, método getter, un menú para escoger el aeropuerto que quiere administrar y el menú de acciones que puede realizar el administrador.

**Clase Gerente:**

Clase encargada de la creación de gerentes. Esta clase hereda de Empleado. Posee un atributo propio que es el nombre de la aerolínea de la que está encargado.

**Métodos:**

Método constructor, método getter y un menú con todas las acciones que puede realizar.

**Clase Operador:**

Clase encargada de la creación de operadores. Esta clase hereda de Empleado. Posee un atributo propio que es el nombre de la aerolínea de la que está encargado.

**Métodos:**

Método constructor y método getter.

### **Clase Pasaporte:**

Clase encargada de la creación de pasaportes, contiene una gran cantidad de atributos. NoPasaporte, constrasena, fechaNacimiento, nacionalidad, estadoCivil, nombre, apellido, sexo, fechaVencimiento, fechaEmision, paisActual, millasRecorridas, boletosComprados.

#### **Métodos:**

Métodos constructores, métodos getters y setters.

### **Clase Reservación:**

Clase encargada de la creación de reservaciones. Contiene atributos noPasaporte, codigoVuelo de tipo int, noTarjeta de tipo long y noAsiento de tipo String.

#### **Métodos:**

Método constructor, métodos getters.

### **Clase Tarjeta:**

Clase encargada de la creación de tarjetas. Contiene atributos noPasaporte, codigoCVC de tipo int, noTarjeta de tipo long y dineroActual de tipo double.

#### **Métodos:**

Método constructor, métodos getters y setters.