

https://github.com/VladShmi/airport_managment/tree/master

Objetivos del programa

El proyecto de gestión aeroportuaria es un sistema diseñado para facilitar la administración y operación de un aeropuerto. Este programa tiene como objetivo mejorar la eficiencia en la gestión de vuelos y pasajeros mediante una interfaz sencilla y funcionalidades clave. A continuación, se detallan las funcionalidades ya integradas en el sistema:

Funcionalidades Integradas:

- 1. Permite registrar nuevos vuelos en el sistema, incluyendo información como el número de vuelo, origen, destino, horarios de salida y llegada.
- 2. Facilita la asignación de pasajeros a vuelos específicos. Se pueden ingresar los datos del pasajero y vincularlos a un vuelo particular.
- 3. Proporciona una lista completa de todos los vuelos registrados en el sistema, permitiendo visualizar detalles importantes de cada uno.
- 4. Muestra la lista de pasajeros asociados a un vuelo específico, lo cual es útil para el control de embarque y gestión de pasajeros.

Objetivos del Proyecto:

·Escalabilidad: Crear una base sólida para futuras expansiones del sistema, permitiendo la integración de nuevas funcionalidades según las necesidades del aeropuerto.

Este programa busca modernizar la gestión aeroportuaria, haciendo que las operaciones sean más fluidas y efectivas en todos los aspectos relacionados con vuelos y pasajeros.

Esquema UML



Viaie

List<Vuelo> vuelos List<Escala> escalas LocalDateTime fechaSalida Aerolinea aerolinea

LocalDateTime getFechaSalida() void setFechaSalida(LocalDateTime fechaSalida) Aerolinea getAerolinea() void setAerolinea(Aerolinea aerolinea) List<Vuelo> getVuelos() List<Escala> getEscalas() void agregarVuelos(Vuelo... vuelos) void agregarEscalas(Escala... escalas) Integer cantPasajerosTotales Double duracionTotalDeVuelosEnMins Double duracionTotalDeEscalasEnMins

Vuelo

String origen; String destino;

double duracionEstimadaEnMins;

LocalDateTime fechaDeLlegadaAproximada;

LocalDateTime fecha;

List<Pasajero> pasajeros;

String getOrigen()

void setOrigen(String origen)

String getDestino()

void setDestino(String destino)

List<Pasajero> getPasajeros() LocalDateTime getFecha()

void agregarPasajero(Pasajero pasajero)

boolean tuDestinoEs(Ciudad ciudad)

double getDuracionEstimadaEnMins()

void setDuracionEstimadaEnMins(double duracionEstimadaEnMins)

int cantPasajeros()

void setFecha(LocalDateTime fecha)

LocalDateTime fechaDeLlegadaAproximada()

void setFechaDeLlegadaAproximada(LocalDateTime fechaDeLlegadaAproximada)

Aeropuerto aeropuerto; Double duracionEnMins;

Double duracionTotalEnMins

void agregarVuelo(Vuelo vuelo)

Aeropuerto getAeropuerto() void setAeropuerto(Aeropuerto aeropuerto) Double getDuracionEnMins() void setDuracionEnMins(Double duracionEnMins)

String nombre Pais pais

String getNombre()

Pais getPais()

Integer cantPasajerosQueLlegaronElDia(LocalDate dia)

Pais

String nombre Continente continente List<Ciudad> ciudades

String getNombre() void setNombre(String nombre) Continente getContinente() void setContinente(Continente continente) void agregarCiudades(Ciudad... ciudades)

Ciudad

List<Aeropuerto> aeropuertos

setNombre(String nombre)

void agregarAeropuertos(Aeropuerto... aeropuertos) Integer cantDeAeropuertos()

Aeropuerto

String nombre String codigoInternacional Ciudad ciudad

List<Vuelo> vuelos

LocalDateTime fechaDeLlegadaAproximada

List<Vuelo> vuelosQueLlegaronElDia(LocalDate d

Avión

String nombre Integer cantAsientos

void setCantAsientos(Integer cantAsientos) String getNombre() void setNombre(String nombre) int getCantAsientos() void setCantAsientos(int cantAsientos)

Aerolinea

List<Vuelo> vuelos

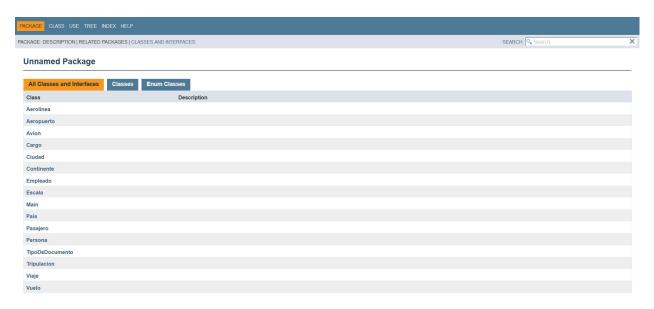
void agregarVuelo(Vuelo vuelo)

void agregarPasajeroAVuelo(String origen, String destino, Pasajero pasajero) void mostrarVuelos()

void mostrarPasajerosDeVuelo(String origen, String destino)

Persona String nombre String apellido Pasajero Integer nroDeDocumento String nombre TipoDeDocumento tipoDeDocumento String apellido List<Vuelo> vuelos String getNombre() Integer nroDePasaporte void setNombre(String nombre) Pais nacionalidad String getApellido() void setApellido(String apellido) Integer getNroDePasaporte() Integer getNroDeDocumento() void setNroDeDocumento(Integer nroDeDocumento) void setNroDePasaporte(Integer nroDePasaporte) Pais getNacionalidad() void setNacionalidad(Pais nacionalidad) TipoDeDocumento getTipoDeDocumento() void setTipoDeDocumento(TipoDeDocumento tipoDeDocumento) List<Vuelo> getVuelos() void agregarVuelos(Vuelo... vuelos) Integer cantVuelosTotales() Integer cantVecesQueVisitaste(Ciudad unaCiudad) String getNombre() void setNombre(String nombre) String getApellido() void setApellido(String apellido) Empleado Cargo cargo Aerolinea empleador Cargo getCargo() Vuelo void setCargo(Cargo cargo) Aerolinea getEmpleador() String origen void setEmpleador(Aerolinea empleador) String destino double duracionEstimadaEnMins LocalDateTime fechaDeLlegadaAproximada LocalDateTime fecha List<Pasaiero> pasaieros String getOrigen() Tripulación void setOrigen(String origen) String getDestino() String nombre void setDestino(String destino) List<Empleado> empleados List<Pasajero> getPasajeros() List<Vuelo> vuelos LocalDateTime getFecha() void agregarPasajero(Pasajero pasajero) void setNombre(String nombre) boolean tuDestinoEs(Ciudad ciudad) String getNombre() double getDuracionEstimadaEnMins() List<Empleado> getEmpleados() void setDuracionEstimadaEnMins(double duracionEstimadaEnMins) List<Vuelo> getVuelos() int cantPasajeros() void agregarEmpleados(Empleado... empleados) void setFecha(LocalDateTime fecha) void agregarVuelos(Vuelo... vuelos) LocalDateTime fechaDeLlegadaAproximada() List<Vuelo> vuelosEntre(LocalDate fechalnicial, void setFechaDeLlegadaAproximada(LocalDateTime fechaDeLlegadaAproximada) LocalDate fechaFinal) Integer cantVuelosEnMesesEntre(LocalDate fechalnicial, LocalDate fechaFinal)

<u>JavaDoc</u>



Todo el javadoc y sus correspondientes clases se encuentran en el siguiente enlace

https://drive.google.com/drive/folders/1Ua5UKqxXew_NsPAEc3s 8twq7zXqYALAU?usp=sharing

Funciones futuras y modularización

Se crean muchas clases distintas con muchos atributos y funcionalidades.

Muchas de estás se han erradicado al final del proyecto.

Otras en cambio permanecen para generar un proyecto modular y extensible de cara a futuro. Se podrán añadir tales funciones como distintas aerolineas, datos adiccionales de los pasajeros tales como pasaportes.

Aparte de esto en algún momento se añadirá funcionalidad gráfica.