UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
ETRUCTURAS DE DATOS

MANUAL TÉCNICO DEL SISTEMA
SISTEMA DE GESTIÓN DE AEROPUERTO

CONTENIDO

INTRODICCIÓN ……..…... 03

REQUISITOS DEL SISTEMA ……..… 04

TECNOLOGIAS Y HERRAMIENTAS UTILIZADAS …… 05 – 06

FUNCIONALIDADES DEL PROGRAMA ……..…… 07

INTRODUCCIÓN

En este proyecto se le solicita al estudiante del curso de Estructuras de Datos desarrollar un sistema de gestión para un aeropuerto que permita gestionar vuelos, pasajeros, equipajes y algunas otras características más. Este sistema utilizará diversas estructuras de datos como listas enlazadas, listas circulares, pilas y colas para manejar las diferentes operaciones y procesos del aeropuerto.

- Implementación

Un aeropuerto busca realizar la simulación de la llegada de aviones y de los pasajeros, cada uno de estos con sus respectivas características y atributos.

EL manual ofrece la información necesaria para saber como esta realizada la aplicación, para que la persona encargada del la empresa sepa de uso correcto, conociendo la estructura de su desarrollo.

El siguiente manual se encuentra las herramientas que se utilizaron para su desarrollo con una breve explicación de su funcionamiento.

REQUISITOS MINIMOS DEL SISTEMA

En esta sección se detallarán los requisitos mínimos del sistema para poder ejecutar la aplicación de sistema de gestión de aeropuertos.

- REQUISITOS MINIMOS

    1. Sistema Operativos: Windows 7
    2. Procesador: Intel Core Celeron
    3. Memoria RAM: 1 GB
    4. Disco Duro: 1 GB
    5. Resolución de pantalla: 1280 x 720 pixeles
    6. Periféricos: Teclado y ratón.

TECNOLOGIAS, HERRAMIENTAS Y DEPENDENCIAS UTILIZADAS

- C++: Es un lenguaje de programación de alto nivel que fue desarrollado por Bjarne Stroustrup como una extensión del lenguaje C. Es conocido por su poder y eficiencia, así como por soportar programación orientada a objetos, programación genérica y manipulación de memoria a bajo nivel. Es ampliamente utilizado en el desarrollo de sistemas/software, videojuegos, aplicaciones en tiempo real, y más.

- Listas doblemente enlazadas: Este es un tipo de estructura de datos que consiste en una secuencia de nodos. Cada nodo contiene tres campos: un dato y dos enlaces que apuntan al nodo anterior y al nodo siguiente en la secuencia. Esto permite recorrer la lista en ambas direcciones, lo que facilita ciertas operaciones como la inserción y eliminación de nodos.

- Pilas : Son estructuras de datos de tipo LIFO (Last In, First Out), donde el último elemento añadido es el primero en ser eliminado. Se utilizan en diversas aplicaciones como la evaluación de expresiones, navegación de páginas (botón de retroceso), y algoritmos de recorrido de grafos.

- Colas: Son estructuras de datos de tipo FIFO (First In, First Out), donde el primer elemento añadido es el primero en ser eliminado. Son usadas en simulaciones, manejo de tareas, buffering de datos y en la programación de sistemas operativos para gestionar procesos.

- Listas circulares doblemente enlazadas: Similar a las listas doblemente enlazadas, pero en este caso, el último nodo está conectado al primero y viceversa. Esto crea un ciclo que puede ser

útil para aplicaciones que requieren un ciclo continuo de nodos, como algoritmos de planificación circular.

- Graphviz: Es una herramienta de software para la visualización de gráficos. Permite a los usuarios representar estructuras de datos gráficas mediante descripciones en lenguajes de script como DOT. Es muy usado para visualizar estructuras de datos, diagramas de flujo, redes, etc.

- JSON (JavaScript Object Notation): Es un formato ligero de intercambio de datos, fácil de leer y escribir para los humanos y fácil de parsear y generar para las máquinas. JSON es utilizado para transmitir datos entre un servidor y una aplicación web como una alternativa a XML.

- La biblioteca nlohmann/json es una popular librería de C++ para manejar datos en formato JSON. Fue creada por Niels Lohmann y es ampliamente utilizada por su simplicidad y eficiencia. Aquí te explico cómo se utiliza para abrir y manejar archivos JSON:

- Características Principales:
  1. Intuitiva: La biblioteca usa tipos de C++ modernos como std::vector y std::map para que el acceso y la manipulación de los datos sean naturales y directos.
  2. Header-only: No requiere instalaciones complicadas ni construcciones previas; incluyes el archivo de cabecera en tu proyecto y estás listo para comenzar.
  3. Compatibilidad: Compatible con los estándares de JSON definidos por RFC 7159/ECMA-404.

- Uso Básico

Para utilizar nlohmann/json en tu proyecto, primero debes incluir la librería. Usualmente, se añade el archivo de cabecera directamente en el proyecto o se instala mediante un gestor de paquetes como vcpkg o conan.

#include <nlohmann/json.hpp>

Usando el espacio de nombres para facilitar el acceso: using json = nlohmann::json;

FUNCIONALIDA DEL PROGRAMA

1. Menú

La aplicación se ejecutará por medio de una consola, la cual pedirá la información necesaria para su correcto funcionamiento. La aplicación tendrá un menú donde el usuario podrá navegar y realizar el ingreso de los datos, lectura de archivos, generación de reportes y las diferentes operaciones en las estructuras de datos.

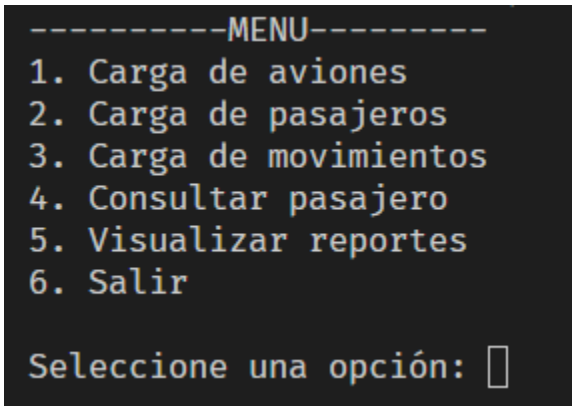


Imagen 1: Menú principal.

2. Carga de aviones

Al inicio de la simulación se establecerá el listado de aviones a partir de un archivo de entrada el cual tendrá toda la información necesaria de cada avión. Estos se estarán almacenando en 2 listas circulares dobles; el primer listado corresponderá a los aviones con estado "Disponible", y el otro listado será para los aviones con estado "Mantenimiento". Posteriormente, a través del menú (ver Imagen 4), los aviones pueden ser movidos de ambas listas por medio de su estado.

```json
[
    {
        "vuelo": "A100",
        "numero_de_registro": "N12345",
        "modelo": "Boeing 737",
        "fabricante": "Boeing",
        "ano_fabricacion": 2015,
        "capacidad": 180,
        "peso_max_despegue": 79000,
        "aerolinea": "AirlineX",
        "estado": "Disponible"
    },
    {
        "vuelo": "A102",
        "numero_de_registro":"N54321",
        "modelo": "Airbus A320",
        "fabricante": "Airbus",
        "ano_fabricacion": 2018,
        "capacidad": 150,
        "peso_max_despegue": 77000,
        "aerolinea": "AirlineY",
        "estado": "Disponible"
    }
]
```


Imagen 2: Entrada en formato json para los aviones.

3. Carga de pasajeros

Los pasajeros se registrarán a partir de un archivo de entrada el cual tendrá toda la información necesaria de cada pasajero, este simulará la ventanilla de registro que luego se almacenará en una cola de llegada.

```json
[
    {
        "nombre": "John Doe",
        "nacionalidad": "Estados Unidos",
        "numero_de_pasaporte": "A12345678",
        "vuelo": "A100",
        "asiento": "12",
        "destino": "New York",
        "origen": "Los Angeles",
        "equipaje_facturado": 2
    },
    {
        "nombre": "Jane Smith",
        "nacionalidad": "Reino Unido",
        "numero_de_pasaporte": "A98765432",
        "vuelo": "A200",
        "asiento": "05",
        "destino": "Londres",
        "origen": "Paris",
        "equipaje_facturado": 1
    }
]
```


Imagen 3: Entrada en formato json para los pasajeros.

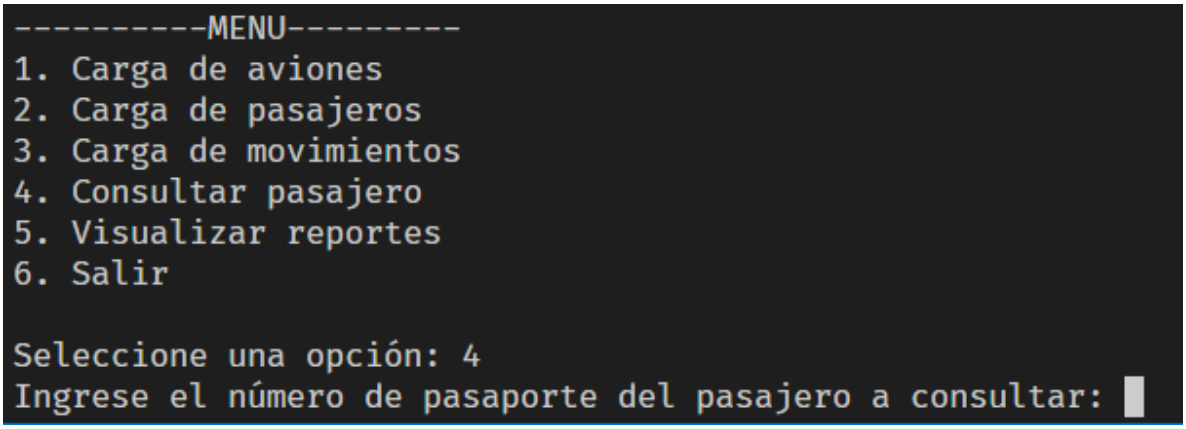
4. Carga de movimientos

Esta información se extraerá de los pasajeros que hayan pasado la ventanilla de registro. Se creará una pila que almacene la cantidad de equipaje de cada pasajero según vayan saliendo de la cola de registro. Si el pasajero no tiene equipaje, será únicamente sacado de la cola sin ingresar a la pila.

```
IngresoEquipajes;
MantenimientoAviones,Ingreso,N12345;
MantenimientoAviones,Salida,C13579;
```

5. Consultar pasajero

Simultáneamente con la creación de la pila, se generará una lista enlazada doble con los pasajeros que hayan salido de la cola de registro, estos serán ordenados por el número de vuelo, en caso de que este se repita, se tendrá como segundo criterio de ordenamiento el número de asiento. Los pasajeros almacenados en esta lista podrán ser consultados por el usuario en cualquier momento durante la ejecución del programa y deberá mostrar toda su información correspondiente en la consola.



6. Visualizar reportes

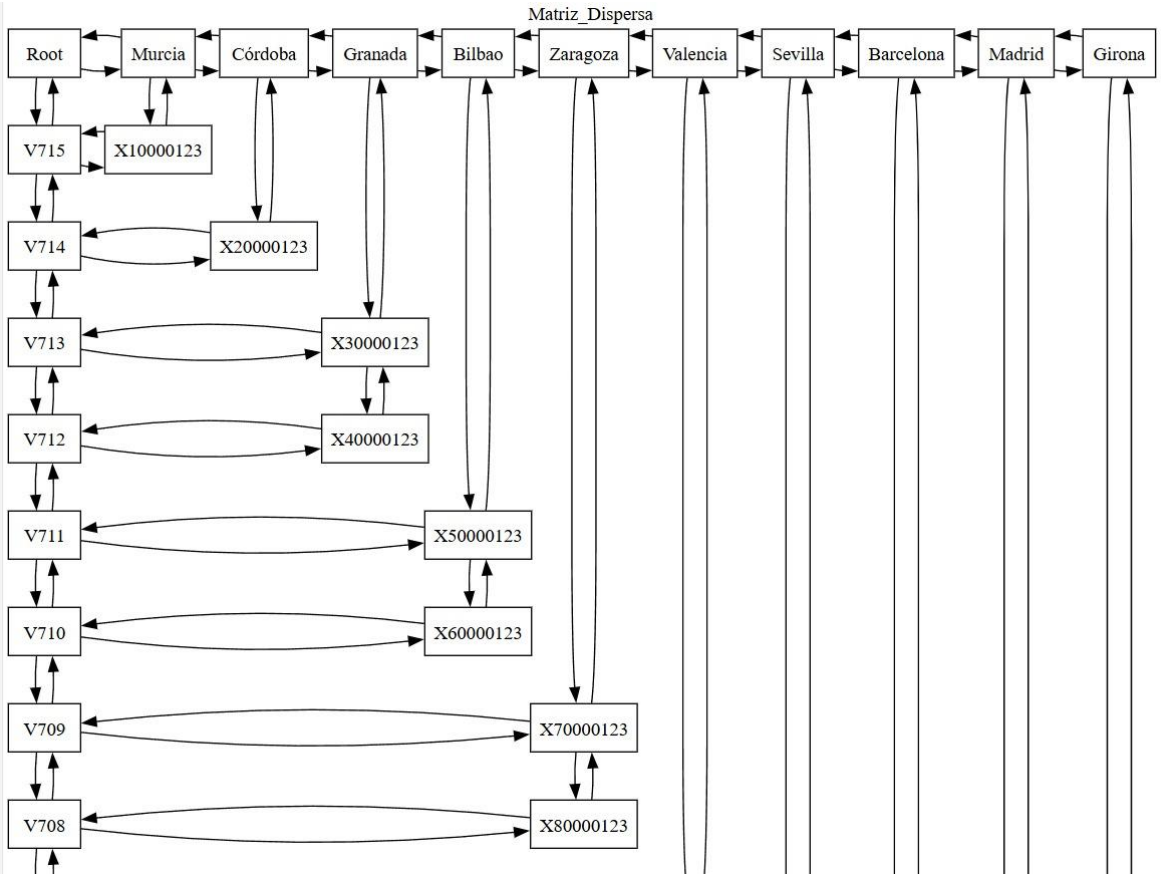
Por medio del menú en la consola el usuario podrá generar y visualizar los siguientes reportes:

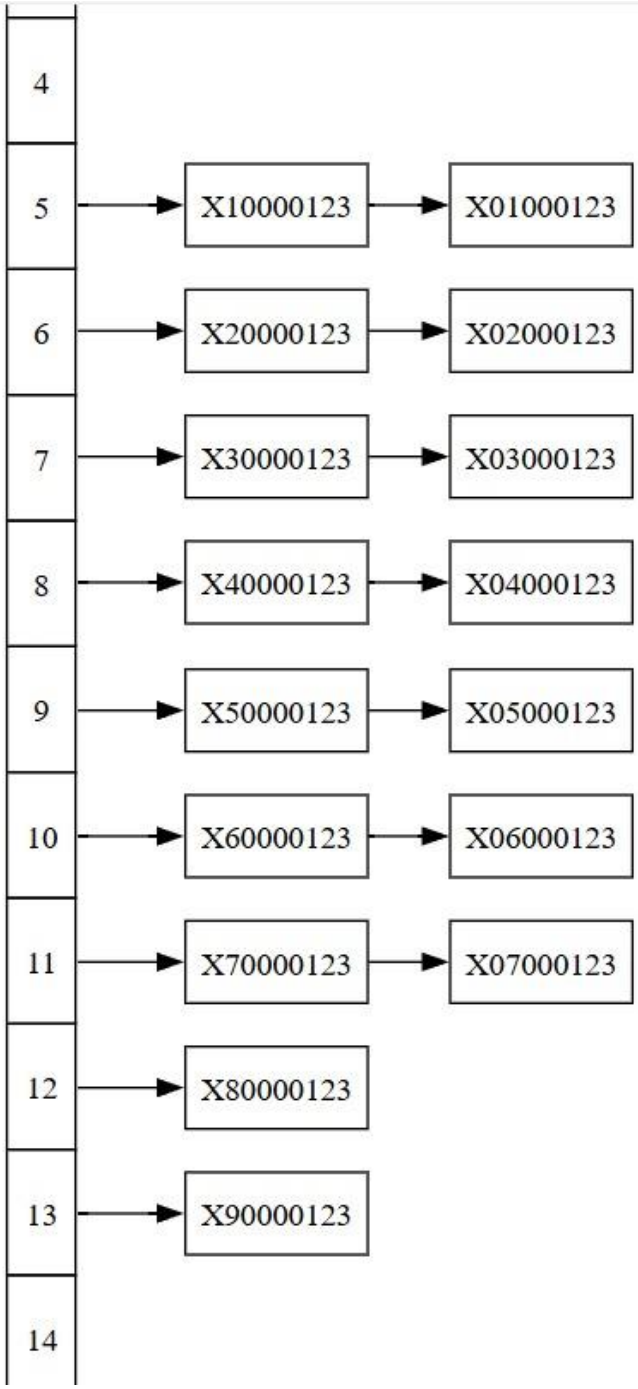
- Lista de aviones disponibles
- Lista de aviones en mantenimiento
- Cola de registro
- Pila de equipaje
- Lista de pasajeros

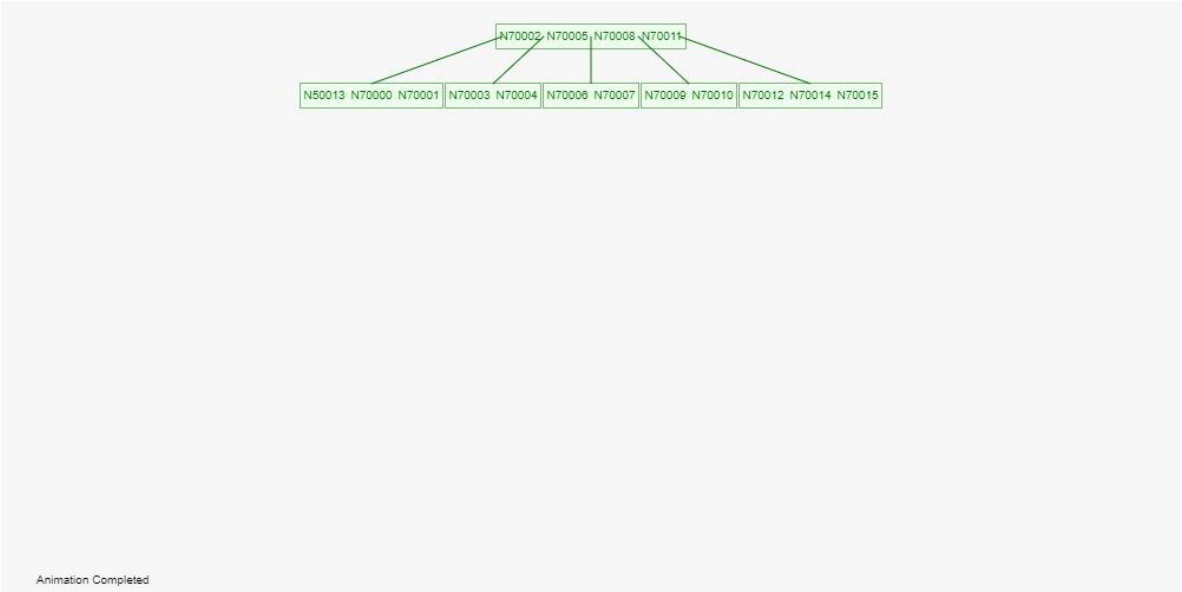
```
----MENU DE REPORTES----
1. Lista de aviones disponibles
2. Lista de aviones en mantenimiento
3. Cola de registro
4. Pila de equipaje
5. Lista de pasajeros
6. Volver al menú principal

Seleccione un reporte para visualizar:
```

Estas son algunas estructuras que se utilizaron .




4
5
X10000123
X01000123
6
X20000123
X02000123
7
X30000123
X03000123
8
X40000123
X04000123
9
X50000123
X05000123
10
X60000123
X06000123
11
X70000123
X07000123
12
X80000123
13
X90000123
14



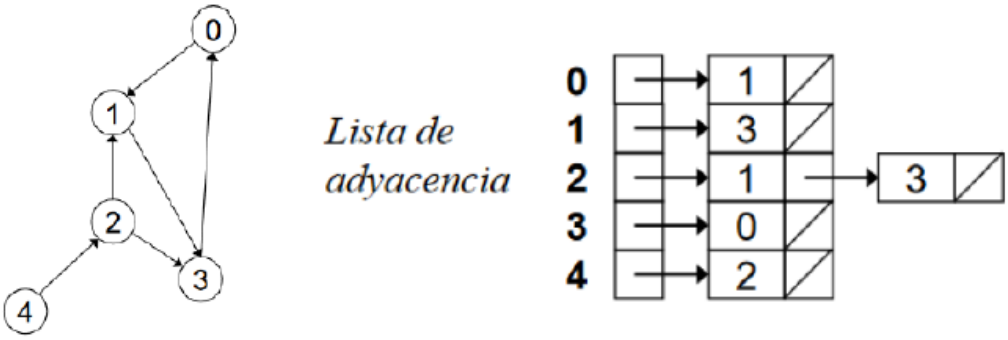


Imagen 4: Lista de adyacencia