Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Estructura de Datos, Vacaciones 1er. Semestre 2024.

Proyecto-Sistema de Gestión de Aeropuerto

202100081 – Javier Andrés Monjes Solórzano

Resumen

Se debe desarrollar un sistema de gestión de aeropuerto aplicando los conocimientos del curso de Estructuras de Datos en C++.

Este sistema permitirá gestionar vuelos, pasajeros, equipajes, pilotos y rutas utilizando diversas estructuras de datos como árbol binario de búsqueda equilibrado, árbol B, matriz dispersa, tabla hash y grafos.

El proyecto implica la correcta utilización de memoria dinámica y apuntadores, y la generación de reportes con Graphviz.

Palabras clave

1. Estructura de Datos
2. Lista Doble Circular
3. Matriz Disperza
4. Tabla Hash
5. Árbol B
6. Árbol Binario

Abstract

An airport management system must be developed, applying the knowledge from the Data Structures course in C++.

This system will enable the management of flights, passengers, luggage, pilots, and routes using various data structures such as balanced binary search trees, B-trees, sparse matrices, hash tables, and graphs.

The project involves the correct use of dynamic memory and pointers, as well as the generation of reports using Graphviz.

Keywords

1. Data Structure

2. Circular Doubly Linked List

3. Sparse Matrix

4. Hash Table

5. B-tree

6. Binary Tree

Introducción

Este manual técnico describe la implementación de una simulación de aeropuerto, gestionando la llegada y el estado de los aviones, así como el manejo de pasajeros y su equipaje.

El objetivo principal es proporcionar una herramienta interactiva que permite al usuario controlar diferentes aspectos operativos del aeropuerto. El desarrollo de software abarca desde la generación de ideas, análisis, diseño y creación, hasta la implementación y verificación de su funcionalidad. Se han utilizado metodologías modernas y eficientes para abordar las necesidades emergentes durante el proceso. Este proyecto, parte del curso de Estructuras de Datos, implica el uso de C++ para manejar archivos JSON y TXT, recopilando y organizando la información del aeropuerto.
Se implementaron clases y estructuras de datos complejas, como árboles binarios de búsqueda, árboles B, grafos con matrices de adyacencia y tablas hash. La generación de reportes con Graphviz y la búsqueda de información son objetivos clave del proyecto, continuando el desarrollo de la prácticas previa.

Desarrollo del tema

El desarrollo de este programa se llevó a cabo en el sistema Operativo Microsof con Windows 11 Home Single.

El lenguaje de programación utilizado para dicho proyecto fue C++, implementando con el compilador de MinGW y utilizando como editor de código fuente Visual Studio Code.

Cabe mencionar que, por el momento, este programa es compatible únicamente con Windows, ya que utiliza ejecutables y dependencias específicas de este sistema operativo

1. Instalación en Windows

1. Descargar e instalar MinGW
2. Asegurarse de seleccionar la opción “mingw32-gcc-gg++” durante la instalación
3. Agregar la ruta a “C:\MinGW” al PATH del sistema.
4. Para comprobar podemos abrir el “cmd” y ejecutar los siguientes comandos:
   gcc –versión

```
C:\Users\javie>gcc --version
gcc (Rev3, Built by MSYS2 project) 14.1.0
Copyright (C) 2024 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Figura 1. Gcc --version.

Fuente: elaboración propia, 2024

g++ --version

```
C:\Users\javie>g++ --version
g++ (Rev3, Built by MSYS2 project) 14.1.0
Copyright (C) 2024 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Figura 2. G++ --version.

Fuente: elaboración propia202

2. Tecnologías Utilizadas

1. *Lenguaje de Programación C++*
2. *Herramienta de Reportes: Graphviz*
3. *Compilador: MinWG-g++*
4. *Editor/IDE: Visual Studio Code*

3. Requerimientos Mínimos del Sistema:

1. *Sistema Operativo:* Windows 7 o superior.
2. *RAM:* 4GB(mínimo) y 8GB (recomendado)
3. *Espacio Disco:* 500MB Libres, para instalación de herramientas, dependencias y la compilación del proyecto.
4. *Dependencias:*
   - Graphviz
   - Compilador C++

4.Descripción de las Estructuras de Datos Utilizadas y su Funcionamiento

4.1 Arból B

- Función: Almacena aviones con estado "Disponible".
- Características: Es un árbol B de orden 5, donde cada nodo puede contener hasta 4 llaves y 5 hijos. Los aviones se organizan según su número de registro.
- Uso: Facilita la búsqueda eficiente de aviones disponibles y permite movimientos de aviones entre "Disponible" y "Mantenimiento".

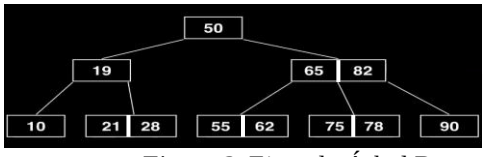

Figura 3. Ejemplo Árbol B.

Fuente: Laboratorio EDD

4.2 Lista Circular Doble

- Función: Almacena aviones con estado "Mantenimiento".
- Características: Es una lista enlazada donde cada nodo contiene información de un avión en mantenimiento, con referencias tanto al siguiente como al anterior nodo para permitir recorridos circulares.
- Uso: Facilita el manejo de aviones en mantenimiento y su posterior regreso a la disponibilidad.

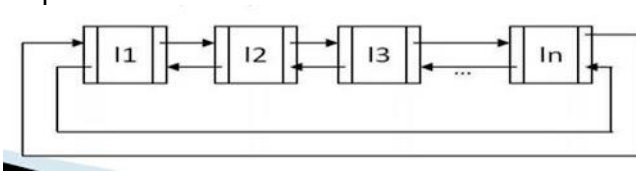

Figura 4. Ejemplo Lista Circular Doble.

Fuente: Laboratorio EDD

4.3 Árbol Binario de Búsqueda

- Función: Almacena pilotos y los ordena por horas de vuelo.
- Características: Cada nodo contiene información de un piloto y está organizado de manera que cada nodo izquierdo contiene pilotos con menos horas de vuelo y cada nodo derecho contiene pilotos con más horas.
- Uso: Permite la búsqueda eficiente de pilotos según su experiencia y facilita la visualización en diferentes formas como preorden, inorden y postorden.

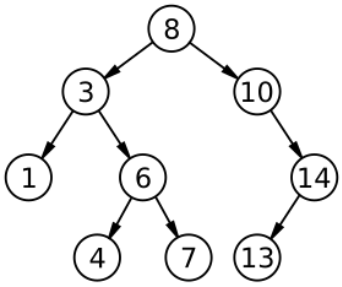

Figura 5. Ejemplo Árbol Binario de Búsqueda.

Fuente: Laboratorio EDD

4.4 Tabla Hash

- Función: Almacena información de pilotos utilizando su número de ID como clave.
- Características: Utiliza una función de dispersión simple (h(llave) = llave mod M) para asignar cada piloto a una posición en la tabla hash inicializada con tamaño 18.
- Uso: Permite un acceso rápido a la información de pilotos mediante su ID, útil para operaciones como dar de baja a un piloto.

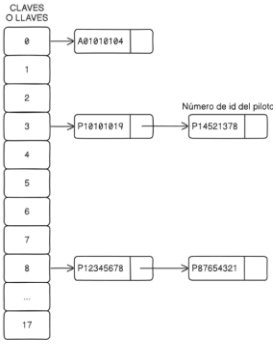

Figura 6. Ejemplo Tabla Hash.

Fuente: Laboratorio EDD

4.5 Grafo Dirigido (Lista Adyacencia)

- Función: Representa las rutas que pueden seguir los aviones entre diferentes ciudades.
- Características: Cada nodo del grafo representa una ciudad y las aristas dirigidas representan las rutas con distancias asociadas.
- Uso: Facilita la gestión de rutas entre ciudades y permite la implementación de algoritmos para recomendar la ruta más corta entre dos ciudades.

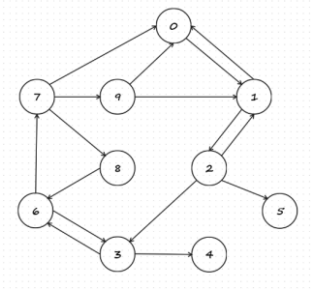

Figura 7. Ejemplo Grafo Dirigido.

Fuente: Laboratorio EDD

4.6 Matriz Disrpesa

- Función: Relaciona vuelos con ciudades destino a través de pilotos asignados.
- Características: Utiliza una estructura eficiente para almacenar la conexión entre vuelos y ciudades, eliminando filas y columnas cuando un piloto se da de baja y no está asociado a ningún vuelo.
- Uso: Mantiene un registro estructurado de qué piloto está asignado a qué vuelo y ciudad destino.

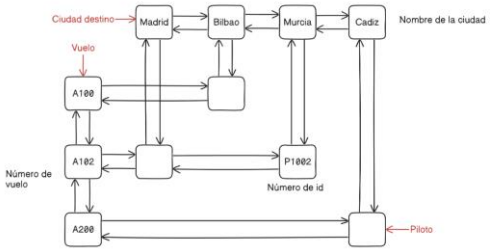

Figura 8. Matriz Dispersa.

Fuente: Laboratorio EDD

5. Menú Principal

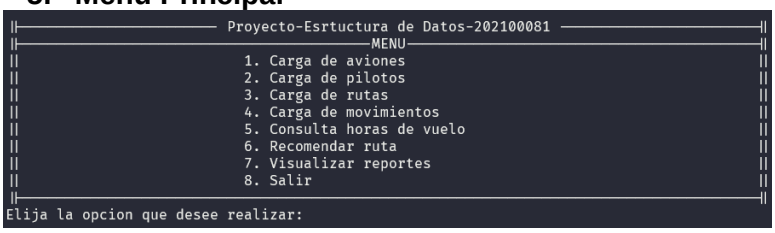

Figura 8. Menú Principal.

Fuente: elaboración propia, 2024

1. Carga Archivos de entrada
   - Permite la carga de archivos JSON y TXT con la información para procesarla.
2. Gestionar Aviones
   - Opción para mover aviones entre las listas de disponibles y mantenimiento
3. Gestionar Pilotos
   - Opción para registrar pilotos y viajes desde un archivo de entrada.
4. Gestionar Rutas
   - Opción para gestionar la rutas de los viajes de “aviones” y “pilotos”
5. Consultar de Horas de Vuelo
   - Opción para la consulta de horas de vuelo de los pilotos
6. Recomendar Rutas
   - Opción para la recomendación de la ruta más corta entre 2 ciudades para futuros viajes.
7. Generar Reportes
   - Genera y se visualizan los respectivos reportes de las estructuras de datos implementadas usando Graphviz.
8. Salir
   - Termina la ejecución de la aplicación

Archivos de Entrada

- Aviones: Archivo en formato JSON que contiene la información de cada avión:

Figura 9. Entrada en formato json para los aviones.

Fuente: Laboratorio EDD Archivos de prueba

- Pilotos: Archivos en formato JSON que contiene la información de cada piloto:

Figura 10. Entrada en formato json para los pilotos.

Fuente: Laboratorio EDD Archivos de prueba

- Movimientos: Archivos en formato TXT que contiene los movimientos de ingreso, salida o dardebaja.

```
MantenimientoAviones,Ingreso,N12345;
MantenimientoAviones,Salida,N13579;
DarDeBaja(P12345678);
DarDeBaja(P98765432);
```

Figura 10. Entrada en formato txt para los movimientos.

Fuente: Laboratorio EDD Enunciado_Proyecto_EDD

- Rutas: Archivos en formato TXT que contiene la información de las rutas que pueden seguir los aviones.

```
Cadiz/Sevilla/125;
Sevilla/Cadiz/125;
Sevilla/Granada/256;
Granada/Sevilla/256;
Sevilla/Jaen/242;
Jaen/Sevilla/242;
Granada/Jaen/99;
```

Figura 11. Entrada en formato txt para las rutas.

Fuente: Laboratorio EDD Enunciado_Proyecto_EDD

Conclusiones:

Este manual técnico presenta de manera explícita las ideas principales y propuestas desarrolladas durante la implementación del Sistema de Gestión de Aeropuertos. A lo largo de este documento, se ha enfatizado la importancia de utilizar estructuras de datos avanzadas como Árbol B, Lista Circular Doble, Árbol Binario de Búsqueda, Tabla Hash, Grafo Dirigido y Matriz Dispersa para optimizar la administración de vuelos, pilotos, rutas y movimientos. La implementación de estas estructuras permite gestionar grandes volúmenes de información de manera eficiente y efectiva, aspecto crucial en el entorno aeroportuario.

Entre las propuestas destacadas se encuentra la integración de Graphviz para la generación de reportes visuales, lo cual añade un valor significativo al proyecto al facilitar la interpretación y análisis de los datos. Este enfoque no solo aplica los conocimientos teóricos adquiridos en el curso, sino que también fortalece la capacidad de los usuarios para desarrollar soluciones tecnológicas avanzadas.

Referencias Bibliográficas

- Graphviz. (n.d.). Gallery: Directed. Retrieved
- Cplusplus.com (n.d) Retrieved
- GeeksforGeeks. (s.f.). Data Structures.