
Manual Tecnico - Proyecto

202200075 – Dominic Juan Pablo Ruano Pérez

Resumen

Este manual técnico describe la instalación, configuración y uso del Sistema de Gestión de Aeropuertos desarrollado en C++. El sistema gestiona vuelos, Pilotos y Ciudades utilizando diversas estructuras de datos como listas enlazadas, Arboles B, arboles Binarios de búsqueda, tablas hash, grafos y matrices esparcidas. El manual incluye pasos para instalar C++ en diferentes sistemas operativos, detalla las tecnologías utilizadas (C++, Graphviz, JSON) y enumera los requisitos mínimos del sistema. Proporciona una descripción completa de las estructuras de datos utilizadas, incluyendo tablas hash, grafos dirigidos y arboles B. El menú principal de la aplicación permite a los usuarios cargar archivos de entrada, gestionar aviones y pilotos. Los archivos de entrada para aviones y pilotos deben estar en formato JSON. Además, el manual incluye espacios reservados para imágenes que ilustran la interfaz de la aplicación y la estructura de los datos. Esta guía asegura una comprensión clara de cómo operar y navegar el sistema de manera efectiva.

Palabras clave

1. Estructuras de Datos
2. Árbol B
3. Árbol BB
4. Tabla hash
5. Grafo Dirigido

Abstract

This technical manual describes the installation, configuration and use of the Airport Management System developed in C++. The system manages flights, Pilots and Cities using various data structures such as linked lists, B-Trees, Binary search trees, hash tables, graphs and sparse arrays. The manual includes steps to install C++ on different operating systems, details the technologies used (C++, Graphviz, JSON) and lists the minimum system requirements. It provides a complete description of the data structures used, including hash tables, directed graphs and B-trees. The main menu of the application allows users to upload input files, manage aircraft and pilots. Input files for aircraft and pilots must be in JSON format. In addition, the manual includes space reserved for images illustrating the application interface and data structure. This guide ensures a clear understanding of how to operate and navigate the system effectively.

Keywords

1. Data Structures
2. B Tree
3. BB Tree
4. Hash Table
5. Digraph

Introducción

El Sistema de Gestión de Aeropuertos desarrollado en C++ se presenta como un proyecto universitario fundamental para la formación de un ingeniero en ciencias y sistemas. En un contexto donde la optimización de operaciones aeroportuarias es crucial, este sistema utiliza estructuras de datos avanzadas como listas enlazadas, listas circulares, pilas y colas para manejar grandes volúmenes de información y procesos en tiempo real. La importancia de este proyecto radica en su capacidad para mejorar la organización y rapidez en la gestión de datos críticos, garantizando así un servicio más eficiente y confiable. Basado en teorías de estructuras de datos y algoritmos, este proyecto no solo pone en práctica los conocimientos adquiridos en el curso, sino que también contribuye con herramientas visuales como Graphviz para la generación de reportes. ¿Cómo puede este sistema transformar la administración de aeropuertos? Este manual técnico busca responder a esta interrogante, demostrando los aportes significativos de la solución propuesta y guiando al usuario en su implementación y uso.

1.1. Instalación Windows:

1. Descargar e instalar MinGW.
2. Asegurarse de seleccionar la opción "mingw32-gcc-g++" durante la instalación.
3. Agregar la ruta C:\MinGW\bin al PATH del sistema.

1.2. Linux (Ubuntu):

1. Abrir la terminal.
2. Ejecutar el comando:
 1. "sudo apt-get update"
 2. "sudo apt-get install g++"

1.3. macOS:

1. Instalar Xcode desde la App Store.
2. Instalar las herramientas de línea de comandos ejecutando en la terminal:
 1. xcode-select --install

2. Tecnologías Empleadas

- *Lenguaje de Programación:* C++
- *Herramientas de Reportes:* Graphviz
- *Compilador:* g++
- *Editor/IDE:* Visual Studio Code

3. Requerimientos Mínimos del Sistema

- *Sistema Operativo:* Windows 7 o superior, Ubuntu 18.04 o superior, macOS Mojave o superior.
- *Memoria RAM:* 4 GB (mínimo), 8 GB (recomendado).
- *Espacio en Disco:* 500 MB libres para la instalación de herramientas y compilación del proyecto.
- *Dependencias:*
 - Graphviz: Para generar reportes visuales de las estructuras de datos.
 - Compilador para C++.

4. Descripción de las Estructuras de Datos y Funcionamiento

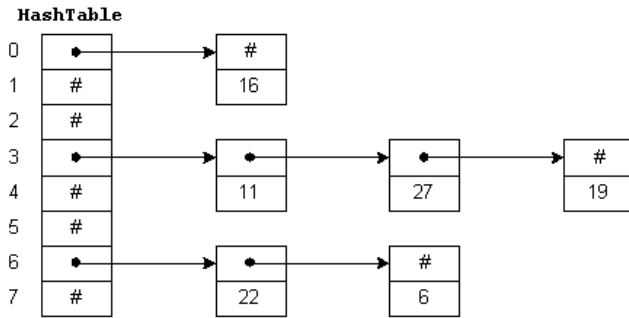
Tabla Hash

Descripción:

Una tabla hash es una estructura de datos que asocia claves con valores. Utiliza una función hash para convertir una clave en una posición en la tabla donde se almacenará el valor correspondiente.

Operaciones:

- **Insertión:** Agrega un par clave-valor en la tabla.
- **Búsqueda:** Encuentra el valor asociado a una clave.
- **Eliminación:** Remueve un par clave-valor de la tabla.



Ventajas:

- Acceso rápido a los elementos.
- Manejo eficiente de colisiones mediante técnicas como la direccionamiento abierto o el encadenamiento.

Árbol B

Descripción:

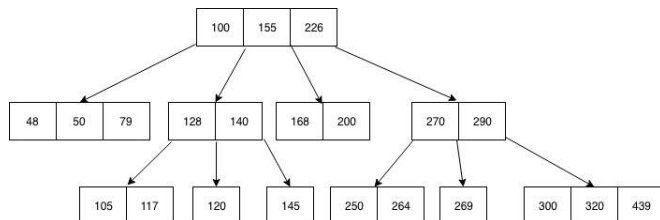
El Árbol B es una estructura de datos de árbol balanceado autoajutable que mantiene los datos ordenados y permite búsquedas, inserciones, y eliminaciones de manera eficiente.

Operaciones:

- **Inserción:** Agrega un elemento en el árbol.
- **Búsqueda:** Encuentra un elemento en el árbol.
- **Eliminación:** Remueve un elemento del árbol.

Ventajas:

- Mantiene un balance de los nodos para asegurar tiempos de operación logarítmicos.
- Es eficiente en operaciones de lectura y escritura en discos.



Árbol BB (Árbol Binario de Búsqueda)

Descripción:

Un Árbol BB es una estructura de datos en forma de árbol donde cada nodo tiene un máximo de dos hijos. Los elementos a la izquierda de un nodo son menores y los elementos a la derecha son mayores.

Operaciones:

- **Inserción:** Agrega un elemento en el árbol respetando el orden.
- **Búsqueda:** Encuentra un elemento en el árbol.
- **Eliminación:** Remueve un elemento del árbol, reorganizando los nodos para mantener el orden.

Ventajas:

- Eficiencia en operaciones de búsqueda, inserción y eliminación.
- Fácil implementación y uso.

Grafo Dirigido

Descripción:

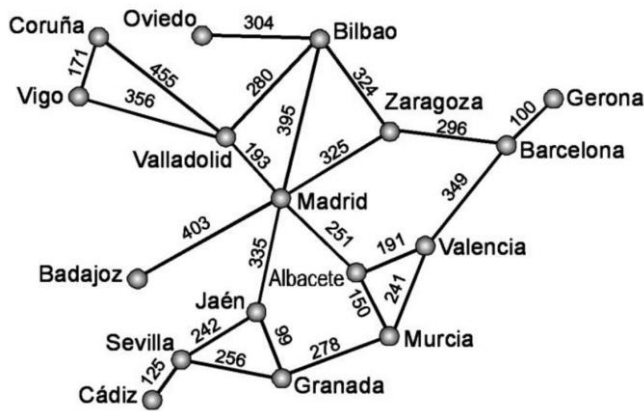
Un grafo dirigido es una colección de nodos conectados por aristas, donde cada arista tiene una dirección asociada.

Operaciones:

- **Inserción de vértice:** Agrega un nuevo nodo al grafo.
- **Inserción de arista:** Crea una conexión dirigida entre dos nodos.
- **Búsqueda:** Encuentra un nodo o una arista en el grafo.
- **Eliminación:** Remueve un nodo o una arista del grafo.

Ventajas:

- Representa relaciones unidireccionales.
- Utilizado en diversas aplicaciones como redes, rutas y modelos de dependencias.



Matriz Esparcida

Descripción:

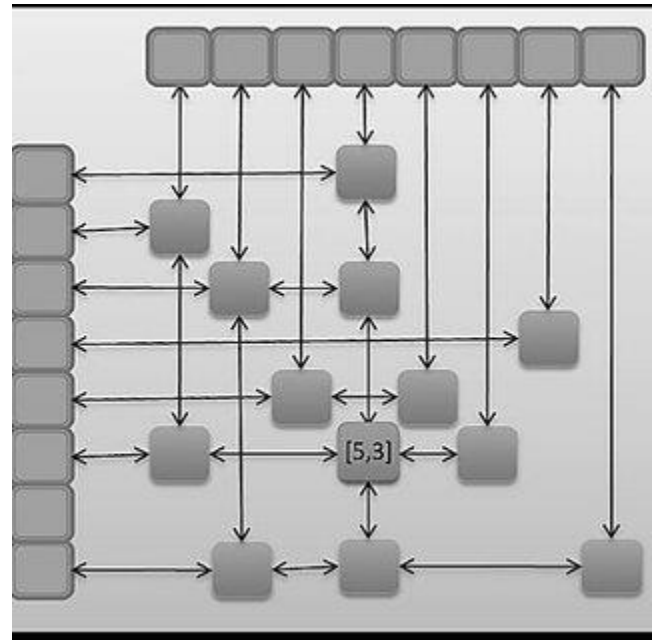
Una matriz esparcida es una matriz en la que la mayoría de los elementos son cero o nulos. Se utiliza para ahorrar espacio y tiempo de procesamiento.

Operaciones:

- **Insertión:** Agrega un elemento en una posición específica.
- **Búsqueda:** Encuentra el elemento en una posición específica.
- **Eliminación:** Remueve un elemento de una posición específica.

Ventajas:

- Eficiente en términos de espacio para matrices con muchos elementos cero.
- Acceso rápido a los elementos no nulos mediante estructuras de almacenamiento compactas.



5. Menú Principal

El menú principal de la aplicación permite al usuario realizar varias operaciones y está estructurado de la siguiente manera:

```
----- MENU ----- [ingrese "exit", para salir de cualquier submenu]
1. Carga de Aviones.
2. Carga de Pilotos.
3. Carga de Rutas.
4. Carga de Movimientos.
5. Consulta de Horas de Vuelo (Pilotos).
6. Recomendar Ruta.
7. Visualizar Reportes.
8. Salir.
Seleccione una opcion: 8
```

1. Cargar Archivos de Entrada:

- Permite cargar los archivos JSON con la información de los aviones y pasajeros.

2. Gestionar Aviones:

- Opción para mover aviones entre las listas de disponibles y en mantenimiento.

3. Registrar Pilotos:

- Opción para registrar Pilotos desde un archivo de entrada.

4. Cargar Rutas:

- Permite gestionar el equipaje de los pasajeros a través de la pila.

5. Generar Reportes:

- Genera y visualiza reportes de las estructuras de datos usando Graphviz.

6. Salir:

- Termina la ejecución de la aplicación.

Archivos de Entrada

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

```
[
  {
    "id": "A123",
    "modelo": "Boeing 747",
    "estado": "Disponible"
  },
  {
    "id": "A124",
    "modelo": "Airbus A320",
    "estado": "Mantenimiento"
  }
]
```

- **Pasajeros:** Archivo en formato JSON que contiene la información de cada pasajero.

```
[
  {
    "pasaporte": "P001",
    "nombre": "John Doe",
    "vuelo": "A123",
    "asiento": "12A",
    "equipaje": 2
  },
  {
    "pasaporte": "P002",
    "nombre": "Jane Smith",
    "vuelo": "A124",
    "asiento": "15B",
    "equipaje": 1
  }
]
```

importancia de utilizar estructuras de datos avanzadas como listas enlazadas, listas circulares, pilas y colas para optimizar la gestión de vuelos, pasajeros y equipajes. La implementación de estas estructuras permite manejar grandes volúmenes de información de manera eficiente y efectiva, lo cual es crucial en el contexto aeroportuario.

Las propuestas destacadas incluyen la integración de Graphviz para la generación de reportes visuales, lo cual añade un valor significativo al proyecto al facilitar la interpretación de los datos y su análisis. Este enfoque no solo pone en práctica los conocimientos teóricos adquiridos en el curso, sino que también mejora la capacidad de los usuarios para desarrollar soluciones tecnológicas complejas.

Referencias bibliográficas

magjac. (n.d.). Graphviz Visual Editor. Retrieved from <http://magjac.com/graphviz-visual-editor/>
Graphviz. (n.d.). Gallery: Directed. Retrieved from <https://graphviz.org/Gallery/directed/>
cplusplus.com. (n.d.). Retrieved from <https://cplusplus.com/>
nlohmann. (n.d.). nlohmann/json. GitHub. Retrieved from <https://github.com/nlohmann/json?tab=readme-ov-file>

Conclusiones

Este manual técnico evidencia claramente las principales ideas y propuestas generadas durante el desarrollo del Sistema de Gestión de Aeropuertos. A lo largo de este documento, se ha detallado la