

Requerimientos Del Sistema Operativo

Procesador (CPU):

- Tipo: 32-bit o 64-bit
- Velocidad mínima recomendada (1 GHz o superior)

Memoria RAM:

- Mínima: (2 GB para sistemas básicos)
- Recomendada: (4 GB o más para un rendimiento óptimo)

Espacio en Disco Duro:

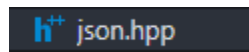
- Mínimo: (1 GB para instalación básica)
- Recomendada: (2 GB para permitir actualizaciones y almacenamiento adicional)

Tarjeta Gráfica:

- Compatibilidad: DirectX 9 o superior con controlador WDDM 1.0
- Memoria: (128 MB de VRAM mínimo)

Dependencias Instaladas

- **json.hpp**
Utilizada principalmente para poder operar con archivos .json esto para poder leer el contenido y poder manipularlos. Al ser externa se tiene que descargar y agregar el archivo a nuestro proyecto para que este funcione de la mejor manera posible.



- **Bibliotecas C++**
Estás son utilizadas para poder trabajar con C++, para mencionar algunas como iostream, cstdlib, windows.h, fstream. Cada una de estas aporta características y funcionalidades diferentes al proyecto, así como su correcta ejecución.

Cabeceras y Principal

Nodos

Estos definen los atributos que tendrá cada nodo, los nodos son utilizados por las distintas listas, son utilizados los nodos para los aviones y pilotos. Para los aviones se utilizaran nodoAviones y Nodob, cada uno tendrá una funcionalidad diferente para cada estructura de datos.

```
h nodoAviones.h
h Nodob.h
h nodoPiloto.h
```

listaCircularAviones

Esta es una lista circular doble en la cual al hacer la carga de aviones estos serán clasificados por su estado. Una lista circular doble para el estado disponible y otra para el estado mantenimiento. En este archivo hay distintas funciones y métodos que ayudaran a realizar todas las operaciones para la simulación de una lista circular doble.

```
h listaCircularAviones.h
```

ArbolB

Esta es una estructura de datos que simula un árbol B, el cual se base en su orden, por cada clave, tendrá un hijo más. En el caso del proyecto, al ser de orden cinco, tendrá cuatro claves y cinco hijos.

```
h ArbolB.h
```

tablaHash

Esta es una estructura de datos que simula una tablaHash. En esta ocasión se tiene una la primer columna que seria un vector y para cada vector tendrá enlazada una lista simple que contendrá todo lo que le mandemos, será auto incrementable y para este caso fue empleado el %18 para poder ordenar los datos conforme al módulo.

```
h tablaHash.h
```