



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

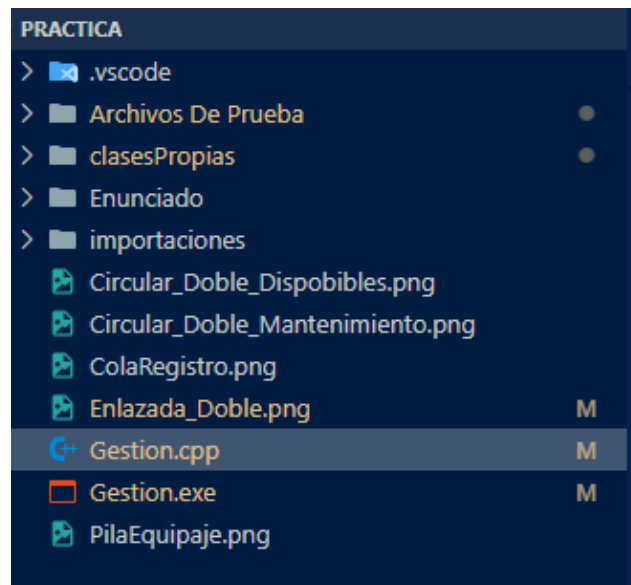
PRÁCTICA

MANUAL TÉCNICO

Estructura De Datos

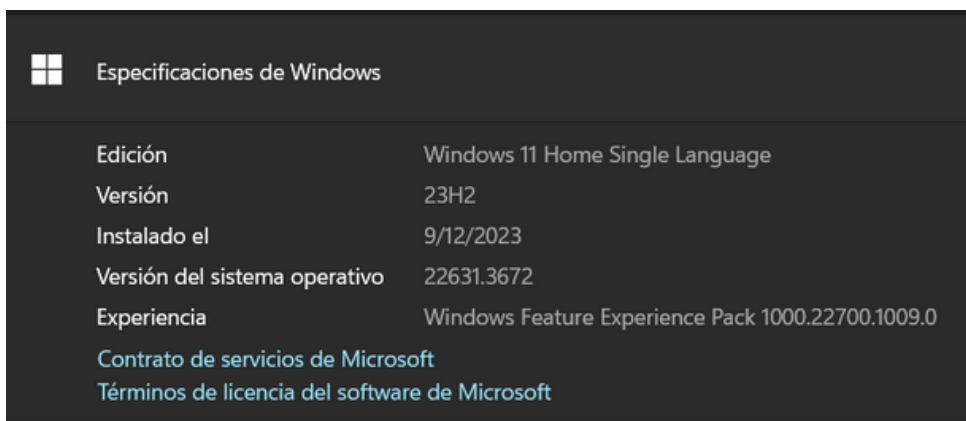
Miguel Adrian Tubac Agustin
202101927

Descripción de la solución



El programa resuelve el problema del control de un aeropuerto, que permite ingresar vuelos, pasajeros y movimiento de los vuelos. Todo lo anterior implementando estructuras de datos creadas para el sistema, es decir, listas, pilas y colas.

Requerimientos mínimos del entorno de desarrollo

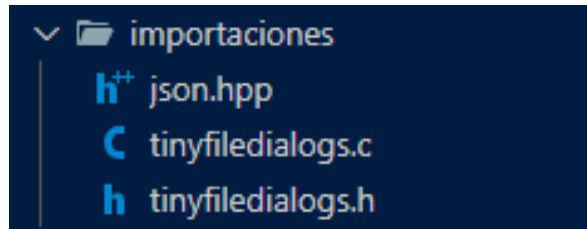
A screenshot of the Windows 'Especificaciones de Windows' (Windows Specifications) window. The window has a dark background with white text. At the top left is the Windows logo. The title bar reads 'Especificaciones de Windows'. Below this, a table lists system information: 'Edición' (Edition) is 'Windows 11 Home Single Language', 'Versión' (Version) is '23H2', 'Instalado el' (Installed on) is '9/12/2023', 'Versión del sistema operativo' (Operating system version) is '22631.3672', and 'Experiencia' (Experience) is 'Windows Feature Experience Pack 1000.22700.1009.0'. At the bottom, there are two links: 'Contrato de servicios de Microsoft' and 'Términos de licencia del software de Microsoft', both in a light blue color.

Edición	Windows 11 Home Single Language
Versión	23H2
Instalado el	9/12/2023
Versión del sistema operativo	22631.3672
Experiencia	Windows Feature Experience Pack 1000.22700.1009.0
Contrato de servicios de Microsoft	
Términos de licencia del software de Microsoft	

Los requerimientos necesarios para la ejecución del programa de gestión del aeropuerto, deberán ser los siguientes para poder garantizar la correcta depuración del mismo:

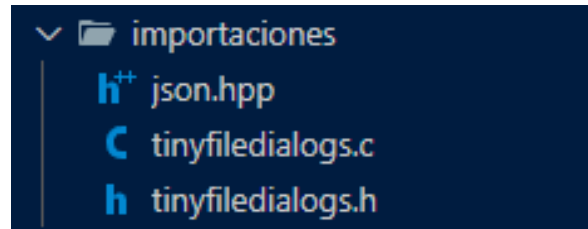
- Capacidad de disco duro 250 GB o mayor.
- Memoria RAM 8 GB mínimo.
- Sistema Operativo Windows 11

Dependencias Instaladas



json.hpp: esta librería nos permite leer archivos en formato json, lo cual es de gran ayuda para el proyecto. Esto debido a que la entrada de los archivos de Aviones y Pasajeros se realiza en el formato anteriormente mencionado.

tinyfiledialogs.c: esta librería nos permite abrir un explorador de archivos del sistema, lo cual es de gran ayuda para poder acceder a los archivos del equipo. Esta característica se emplea en la búsqueda y selección de los archivos de entrada, estos archivos posteriormente son analizados por la librería anteriormente mencionada.



tinyfiledialogs.h: esta librería es la cabecera de la anterior librería, ya que se utiliza para el acceso al explorador de archivos.

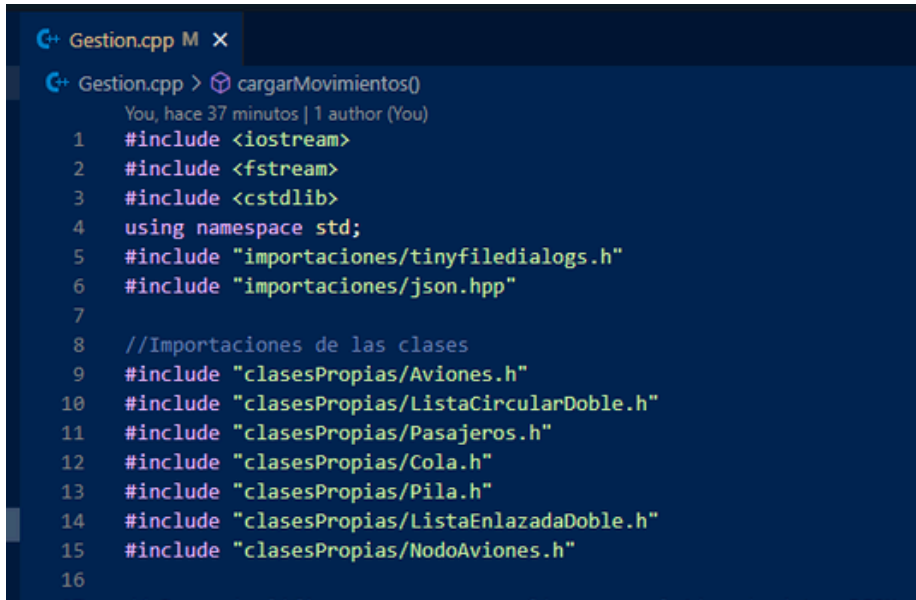
Estas librerías están disponibles en la web, la cual se encuentran en el siguiente repositorio:

<https://github.com/native-toolkit/libtinyfiledialogs>

Para el caso de la librería de json, esta se encuentra en el siguiente repositorio:

<https://github.com/nlohmann/json>

Desarrollo del Proyecto

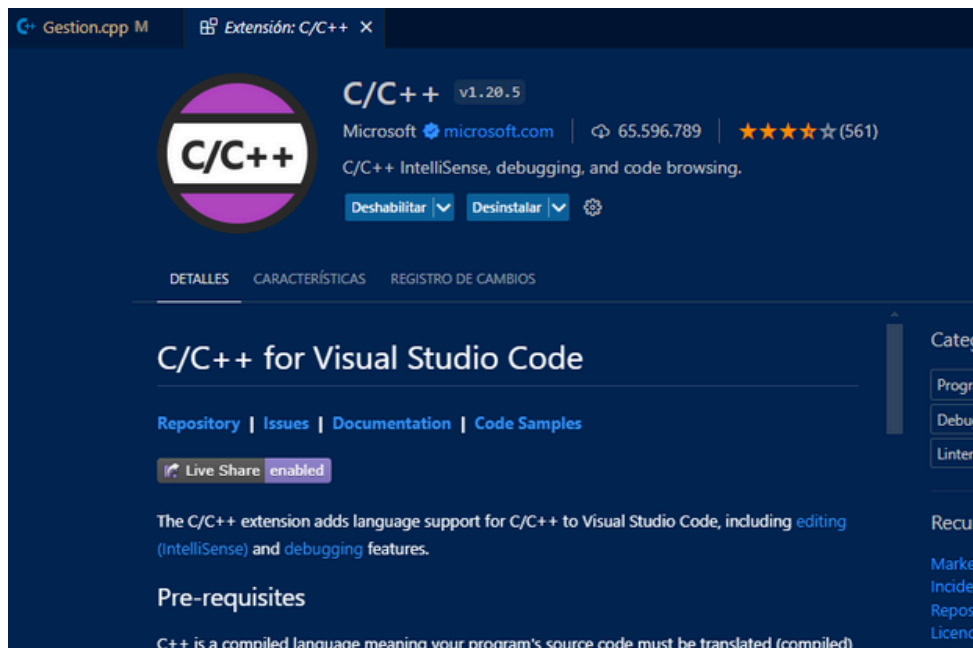


```
G+ Gestion.cpp M X
G+ Gestion.cpp > cargarMovimientos()
You, hace 37 minutos | 1 author (You)
1  #include <iostream>
2  #include <fstream>
3  #include <cstdlib>
4  using namespace std;
5  #include "importaciones/tinyfiledialogs.h"
6  #include "importaciones/json.hpp"
7
8  //Importaciones de las clases
9  #include "clasesPropias/Aviones.h"
10 #include "clasesPropias/ListaCircularDoble.h"
11 #include "clasesPropias/Pasajeros.h"
12 #include "clasesPropias/Cola.h"
13 #include "clasesPropias/Pila.h"
14 #include "clasesPropias/ListaEnlazadaDoble.h"
15 #include "clasesPropias/NodoAviones.h"
16
```

El programa fue creado con el editor de visual studio code, inicialmente este IDE no cuenta con el compilador adecuado para C++, por lo cual se necesita realizar los siguientes pasos:

Se debe de instalar el compilador para C++, esto debido a que el sistema operativo de Windows no lo trae por defecto. Este compilador se encuentra en la siguiente dirección:

<https://code.visualstudio.com/docs/languages/cpp>



El segundo paso consiste en buscar la anterior extensión dentro del propio visual studio code e instalarla, con lo cual ya tendremos el entorno necesario para la ejecución del sistema.

Ejecución del sistema:

Para la ejecución del sistema primero se deberá de crear el archivo ejecutable con el siguiente comando:

```
g++ -o Gestion Gestion.cpp clasesPropias/Aviones.cpp  
clasesPropias/ListaEnlazadaDoble.cpp  
clasesPropias/Pila.cpp  
clasesPropias/ListaCircularDoble.cpp  
clasesPropias/Pasajeros.cpp      clasesPropias/Cola.cpp  
importaciones/tinyfiledialogs.c -lole32 -lcomdlg32
```



 Gestion.cpp	M
 Gestion.exe	M

El anterior comando nos generará el .exe, con lo cual deberemos de ingresar el siguiente comando:

./Gestion.exe

Este ejecutará el programa y se utilizará de la manera que mejor le convenga.

