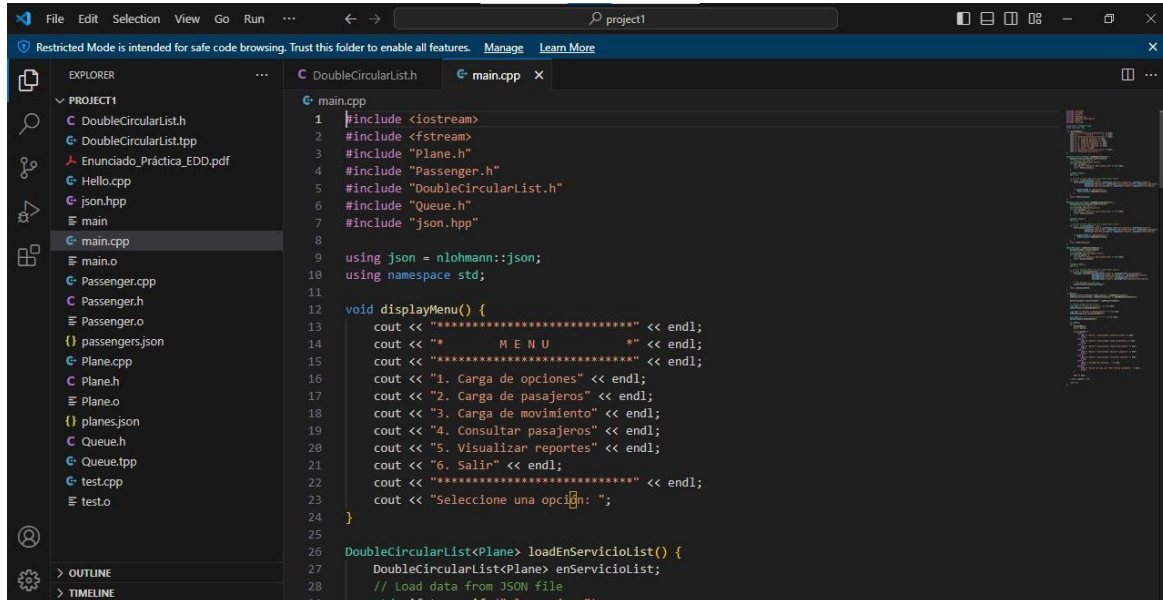


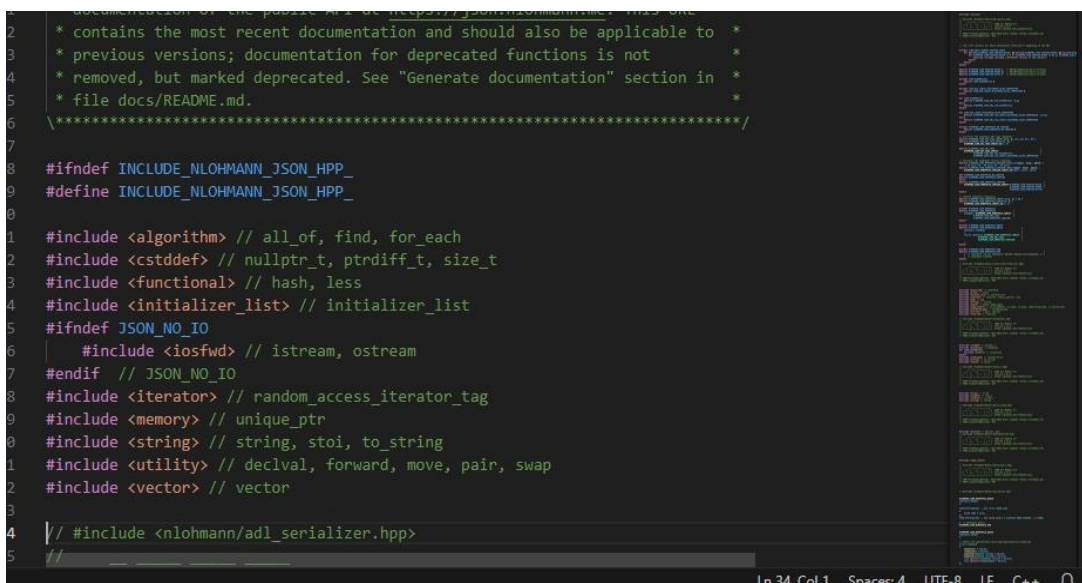
Manual Técnico

Se va a tener un proceso de gestión de vuelos con el cual nos permita gestionar vuelos, pasajeros y equipajes, y algunas características como saber el estado del vuelo, Así como las dependencias usadas de igual manera



```
1 #include <iostream>
2 #include <fstream>
3 #include "Plane.h"
4 #include "Passenger.h"
5 #include "DoubleCircularList.h"
6 #include "Queue.h"
7 #include "json.hpp"
8
9 using json = nlohmann::json;
10 using namespace std;
11
12 void displayMenu() {
13     cout << "*****" << endl;
14     cout << "      M E N U      " << endl;
15     cout << "*****" << endl;
16     cout << "1. Carga de opciones" << endl;
17     cout << "2. Carga de pasajeros" << endl;
18     cout << "3. Carga de movimiento" << endl;
19     cout << "4. Consultar pasajeros" << endl;
20     cout << "5. Visualizar reportes" << endl;
21     cout << "6. Salir" << endl;
22     cout << "*****" << endl;
23     cout << "Seleccione una opción: ";
24 }
25
26 DoubleCircularList<Plane> loadEnServicioList() {
27     DoubleCircularList<Plane> enServicioList;
28     // Load data from JSON file
29     ifstream f("planes.json");
30     if (!f.is_open()) {
31         cout << "Error al abrir el archivo planes.json" << endl;
32         return enServicioList;
33     }
34     json j;
35     f >> j;
36     f.close();
37     for (auto p : j) {
38         Plane pl(p["id"], p["nombre"], p["modelo"], p["precio"]);
39         enServicioList.add(pl);
40     }
41     return enServicioList;
42 }
```

También se usó Json para las listas y tablas



```
1 // documentation of the public API is https://nlohmann.me/ this one
2 * contains the most recent documentation and should also be applicable to *
3 * previous versions; documentation for deprecated functions is not *
4 * removed, but marked deprecated. See "Generate documentation" section in *
5 * file docs/README.md. *
6 //*****
7
8 #ifndef INCLUDE_NLOHMANN_JSON_HPP_
9 #define INCLUDE_NLOHMANN_JSON_HPP_
10
11 #include <algorithm> // all_of, find, for_each
12 #include <cstdint> // nullptr_t, ptrdiff_t, size_t
13 #include <functional> // hash, less
14 #include <initializer_list> // initializer_list
15 #ifndef JSON_NO_IO
16     #include <iosfwd> // istream, ostream
17 #endif // JSON_NO_IO
18 #include <iterator> // random_access_iterator_tag
19 #include <memory> // unique_ptr
20 #include <string> // string, stoi, to_string
21 #include <utility> // declval, forward, move, pair, swap
22 #include <vector> // vector
23
24 // #include <nlohmann/adl_serializer.hpp>
25 //
```

Se usaron listas circulares, también en la parte del main donde se explicó que va el menú de opciones, así como los pasajeros y aviones, en su respectivo bloque.

