



```

1  #include <iostream>
2
3  class Avion {
4  public:
5      Avion(const std::string& modelo, int capacidad) : modelo(modelo), ca
6
7      std::string obtenerModelo() const {
8          return modelo;
9      }
10
11      int obtenerCapacidad() const {
12          return capacidad;
13      }
14
15  private:
16      std::string modelo;
17      int capacidad;
18  };
19  int main() {
20
21      Avion avion1("Boeing 747", 400);
22
23      return 0;
24  }
25
  
```

The screenshot shows a C++ code editor with the following structure:

- EXPLORER** sidebar:
  - HELLOWORLD
    - .vscode
    - Airport
      - ec.edu.espeairport.controls
      - ec.edu.espeairport.model
        - Avion** (selected)
        - ec.edu.espeairport.view
- Avion** class implementation in `Avion.cpp`:
  - Includes `<iostream>`.
  - Defines the `Avion` class with a public constructor `Avion(const std::string& modelo, int capacidad)` and two public methods: `obtenerModelo()` and `obtenerCapacidad()`.
  - Private attributes are `std::string modelo` and `int capacidad`.
  - The `main` function creates an instance `Avion avion1("Boeing 747", 400);` and returns `0`.