```cpp
#include <iostream>
#include <string>
#include <list>
#include <map>

using namespace std;

struct piloto
{
    string nombre, apellido, escuderia;
    int pos_clas;
};

ostream &operator<<(ostream &f, const piloto &pil)
{
    f << "DATOS DEL PILOTO::" << endl
      << "NOMBRE->" << pil.nombre << endl
      << "APELLIDO->" << pil.apellido << endl
      << "ESCUDERIA->" << pil.escuderia << endl
      << "POSICION->" << pil.pos_clas << endl;
    return f;
}

typedef map<string, list<piloto>::iterator> dicc_String;
typedef map<int, list<piloto>::iterator> dicc_Int;

class datos_pilotos
{
private:
    list<piloto> pilotos;
    dicc_String dicc_Nombre;
    dicc_String dicc_Apellido;
    dicc_String dicc_Escuderia;
    dicc_Int dicc_Posicion;

public:
    void add_Piloto(const piloto &nuevo);
    piloto getPorNombre(string nombre) const;
    piloto getPorApellido(string apellido) const;
    piloto getPorEscuderia(string escuderia) const;
    piloto getPorPosicion(int posicion) const;
};

void datos_pilotos::add_Piloto(const piloto &nuevo)
{
    pilotos.push_back(nuevo);
    list<piloto>::iterator it;
    it = this->pilotos.end();
    it--;
    this->dicc_Nombre.insert(dicc_String::value_type(nuevo.nombre, it));
    this->dicc_Apellido.insert(dicc_String::value_type(nuevo.apellido, it));
    this->dicc_Escuderia.insert(dicc_String::value_type(nuevo.escuderia, it));
    this->dicc_Posicion.insert(dicc_Int::value_type(nuevo.pos_clas, it));
}

piloto datos_pilotos::getPorNombre(string nombre) const
{
    return *(dicc_Nombre.find(nombre)->second);
}
piloto datos_pilotos::getPorApellido(string apellido) const
```

```
{
    return *(dicc_Apellido.find(apellido)->second);
}
piloto datos_pilotos::getPorEscuderia(string escuderia) const
{
    return *(dicc_Escuderia.find(escuderia)->second);
}
piloto datos_pilotos::getPorPosicion(int posicion) const
{
    return *(dicc_Posicion.find(posicion)->second);
}

int main()
{
    datos_pilotos prueba;
    piloto P1;
    P1.nombre = "LEWIS";
    P1.apellido = "HAMILTON";
    P1.escuderia = "MERCEDES";
    P1.pos_clas = 1;
    prueba.add_Piloto(P1);
    piloto P2;
    P2.nombre = "CHARLES";
    P2.apellido = "LECLERC";
    P2.escuderia = "FERRARI";
    P2.pos_clas = 3;
    prueba.add_Piloto(P2);
    piloto P3;
    P3.nombre = "MAX";
    P3.apellido = "VERSTAPPEN";
    P3.escuderia = "RED BULL";
    P3.pos_clas = 2;
    prueba.add_Piloto(P3);
    piloto P4;
    P4.nombre = "CARLOS";
    P4.apellido = "SAINZ";
    P4.escuderia = "MCLAREN";
    P4.pos_clas = 4;
    prueba.add_Piloto(P4);
    cout << "BUSQUEDA POR APELLIDO" << endl << prueba.getPorApellido("LECLERC");
    cout << "------------------------------" << endl;
    cout << "BUSQUEDA POR NOMBRE" << endl << prueba.getPorNombre("CARLOS");
    cout << "------------------------------" << endl;
    cout << "BUSQUEDA POR ESCUDERIA" << endl << prueba.getPorEscuderia("RED BULL");
    cout << "------------------------------" << endl;
    cout << "BUSQUEDA POR POSICION" << endl << prueba.getPorPosicion(1);
}
```