

Dynamic cast

https://en.cppreference.com/w/cpp/language/dynamic_cast

¿Para qué sirve?

Puedes convertir los punteros a las clases que actúan en la herencia.



```
1  class Padre {
2
3  public:
4      virtual ~Padre() {}
5  };
6
7  class Hijo : public Padre {
8  public:
9      void printHijo() {
10         cout << "Soy un hijo" << endl;
11     }
12 };
13
14 class Hija : public Padre {
15 public:
16     void printHija() {
17         cout << "Soy una hija" << endl;
18     }
19 };
20
21 int main() {
22     vector<Padre*> v;
23
24     v.push_back(new Hijo());
25     v.push_back(new Hija());
26
27     v[0]→printHijo(); // error
28     v[1]→printHija(); // error
29
30     return 0;
31 }
```



```
1  class Padre {
2
3  public:
4      virtual ~Padre() {}
5  };
6
7  class Hijo : public Padre {
8  public:
9      void printHijo() {
10         cout << "Soy un hijo" << endl;
11     }
12 };
13
14 class Hija : public Padre {
15 public:
16     void printHija() {
17         cout << "Soy una hija" << endl;
18     }
19 };
20
21 int main() {
22     vector<Padre*> v;
23
24     v.push_back(new Hijo());
25     v.push_back(new Hija());
26
27     dynamic_cast<Hijo*>(v[0])→printHijo();
28     dynamic_cast<Hija*>(v[1])→printHija();
29
30     return 0;
31 }
```

Puedes verificar si es una clase u otra

Debes de compararlo con `nullptr`

```
1  class Padre {
2
3  public:
4      virtual ~Padre() {}
5  };
6
7  class Hijo : public Padre {
8  public:
9      void printHijo() {
10         cout << "Soy un hijo" << endl;
11     }
12 };
13
14 class Hija : public Padre {
15 public:
16     void printHija() {
17         cout << "Soy una hija" << endl;
18     }
19 };
20
21 int main() {
22     vector<Padre*> v;
23
24     v.push_back(new Hijo());
25     v.push_back(new Hija());
26
27     cout << boolalpha << (dynamic_cast<Hijo*>(v[0]) == nullptr) << endl; // false
28     cout << boolalpha << (dynamic_cast<Hija*>(v[0]) == nullptr) << endl; // true
29
30     cout << boolalpha << (dynamic_cast<Hijo*>(v[1]) == nullptr) << endl; // true
31     cout << boolalpha << (dynamic_cast<Hija*>(v[1]) == nullptr) << endl; // false
32
33     return 0;
34 }
```