


## Metodología de la Programación

### Directiva de aserción "assert"



**Sintaxis:**

assert(<expresión>)

#include <cassert>

➔ Si la <expresión> evalúa a **true**, entonces no hace nada y continúa la ejecución del programa.

➔ Si la <expresión> evalúa a **false**, entonces muestra un mensaje de error y termina la ejecución del programa.


```
#include <cassert>
#include <iostream>
using namespace std;
int funcion(int x) {
    assert(x>=0 && x<10);
    return x*2;
}
int main() {
    cout << funcion(4)
    cout << funcion(100)
}
```

**ejemplo.cpp**

➔ Ejecución


```
> g++ ejemplo.cpp -o ejemplo
> ./ejemplo
8
ejemplo: ejemplo.cpp:5: int funcion(int):
Assertion `x>=0 && x<10' failed.
Abort
```

Metodología de la Programación - Javier Martínez Baena/Joaquín Fdez-Valdivia



## Metodología de la Programación

### Directiva de aserción "assert"




```
#include <cassert>
#include <iostream>
using namespace std;
int funcion(int x) {
    assert(x>=0 && x<10);
    return x*2;
}
int main() {
    cout << funcion(4) << endl;
    cout << funcion(100) << endl;
}
```

```
#include <cassert>
#include <iostream>
#include <cstdlib>
using namespace std;
int funcion(int x) {
    if !(x>=0 && x<10) {
        cout << "ERROR" << endl;
        exit(EXIT_FAILURE);
    }
    return x*2;
}
int main() {
    cout << funcion(4) << endl;
    cout << funcion(100) << endl;
}
```


El resultado es similar a este:

Metodología de la Programación - Javier Martínez Baena/Joaquín Fdez-Valdivia - Dpto. Ciencias de la Computación e I.A. (Univ. Granada)



## Metodología de la Programación

### Directiva de aserción "assert"



→ **assert** es una MACRO (definida con **#define**).

Se suele usar durante el proceso de desarrollo.

Se anula definiendo la macro **NDEBUG**.

```
#define NDEBUG
g++ -DNDEBUG ...
```

```
#include <cassert>
#include <iostream>
using namespace std;
int funcion(int x) {
    assert(x>=0 && x<10);
    return x*2;
}
int main() {
    cout << funcion(4)
    cout << funcion(100)
}
```

**ejemplo.cpp**

Ejecución

```
> g++ -DNDEBUG ejemplo.cpp -o ejemplo
> ./ejemplo
8
200
```

Metodología de la Programación - Javier Martínez Baena/Joaquín