

**ALGORITMOS Y ESTRUCTURAS DE DATOS (CC182)****Ciclo 2023-02**

Procedimiento: Identificación Big-O

Elaborado por: Robert Zubieta

Fuente: Propia

Identificación de Notación Big-O (C++)**O(1)**

```
//-----  
  
string entrada;           // O(1)  
  
cin >> entrada;          // O(1)  
int x = 5;                // O(1)  
  
if (entrada == "Hola")    // O(1)  
    cout << "Saludo " << x; // O(1)
```

O(n)

```
//-----  
  
//Donde "n" es la entrada  
for (int i = 1; i <= n; i += c) { // O(n)  
    //Cualquier sentencia      // O(1)  
}
```

```
//Donde "n" es la entrada
for (int i = n; i > 0; i -= c) { // O(n)
    //Cualquier sentencia      // O(1)
}
```

O(1)

```
//-----

//Donde "c" NO varía con la entrada
for (int i = 1; i <= c; ++i) { // O(1)
    //Cualquier sentencia      // O(1)
}
```

O(n^2)

```
//-----

//Donde "n" es la entrada
for (int i = 1; i <= n; i += c) {
    for (int j = 1; j <= n; j += c) { // O(n^2)
        //Cualquier sentencia      // O(1)
    }
}

for (int i = n; i > 0; i -= c) {
    for (int j = i + 1; j <= n; j += c) { // O(n^2)
        //Cualquier sentencia      // O(1)
    }
}
//Depende de cuantas veces se ejecuta el ciclo más interno.
```

O(log[n])

```
//-----

//Donde "n" es la entrada
for (int i = 1; i <= n; i *= c) { //O(log[n])
    //Cualquier sentencia      // O(1)
}

for (int i = n; i > 0; i /= c) { //O(log[n])
    //Cualquier sentencia      // O(1)
}
```

$O(\log[\log[n]])$

```
//-----  
  
//Donde "n" es la entrada  
for (int i = 2; i <= n; i *= pow(i, c)) {    //O(log[log[n]])  
    //Cualquier sentencia    // O(1)  
}  
  
for (int i = n; i > 1; i = fun(i)) {    //O(log[log[n]])  
    //Cualquier sentencia    // O(1)  
}  
  
//-----
```