

CUESTIONES DE ESTILO

Escribimos código para que lo puedan leer otras personas, no sólo para que lo traduzca el compilador.

× Identificadores

Los identificadores deben ser descriptivos

+ p, i, s... ❌

+ precio, izquierda, suma..✅

- En ocasiones, se permite el uso de nombres cortos para variables locales cuyo significado es evidente (p.ej. bucles controlados por contador)

+ for (elemento=0; elemento<N; elemento++)... ❌

+ for (i=0; i<N; i++) ...✅

× Constantes

Se considera una mala costumbre incluir literales de tipo numérico (“números mágicos”) en medio del código. Se prefiere la definición de constantes simbólicas (con #define o, mejor aún, con enum).

for (i=0; i<79; i++) ...❌

for (i=0; i<columnas-1; i++) ..✅

Expresiones

- Expresiones bo oleanas:

Es aconsejable escribirlas como se dirían en voz alta.

```
if ( !(bloque<actual) ) ... ❌
```

```
if ( bloque >= actual ) ... ✅
```

- Expresiones complejas:

Es aconsejable dividir las para mejorar su legibilidad

```
x += ( xp = ( 2*k<(n-m) ? c+k : d-k ) ); ❌
```

```
if ( 2*k < n-m )
    xp = c+k;
else
    xp = d-k;
x += xp; ✅
```

Comentarios

- Comentarios descriptivos: Los comentarios deben comunicar algo. Jamás se utilizarán para “parafrasear” el código y repetir lo que es obvio.

```
i++; /* Incrementa el contador */ ❌
```

```
/* Recorrido secuencial de los datos*/ ✅
for (i=0; i<N; i++) ...
```

Estructuras de control

- Sangrías: Conviene utilizar espacios en blanco o separadores para delimitar el ámbito de las estructuras de control de nuestros programas.
- Líneas en blanco: Para delimitar claramente los distintos bloques de código en nuestros programas dejaremos líneas en blanco entre ellos.
- Salvo en la cabecera de los bucles for, sólo incluiremos una sentencia por línea de código.
- Sean cuales sean las convenciones utilizadas al escribir código (p.ej. uso de sangrías y llaves), hay que ser consistente en su utilización.

```
while (...) {
    ...
}

while (...)
{
    ...
}

for (...;...;...) {
    ...
}

for (...;...;...)
{
    ...
}

if (...) {
    ...
}

if (...)
{
    ...
}
```

***El código bien escrito es más fácil de leer, entender y mantener
(además, seguramente tiene menos errores)***