Indentación del código fuente

Por

[**Cristóbal González Almirón**](https://www.adictosaltrabajo.com/author/cristobal-gonzalez/)

 -

17 abril, 2008

Indentación del código fuente

1. [Indentación del  
   código fuente](https://www.adictosaltrabajo.com/2008/04/17/indentacion/#mozTocId476441)
   1. [Objetivo de la indentación](https://www.adictosaltrabajo.com/2008/04/17/indentacion/#mozTocId342817)
   2. [Requerimientos para definir un estilo  
      de indentación](https://www.adictosaltrabajo.com/2008/04/17/indentacion/#mozTocId309683)

Todos los desarrolladores utilizan algún estilo de colocación de los elementos del código fuente dentro del texto que los contiene. Esto se define como el estilo de codificación. Las normas de indentación indican la posición en la que se deben colocar los diferentes elementos que se incluyen en el código fuente, por lo que forman parte del estilo de codificación. Otro ejemplo de ello es la separación con espacios en blanco entre los diferentes elementos que componen las líneas de código.

Objetivo de la indentación

El objetivo fundamental de la indentación del código fuente es facilitar su  
lectura y comprensión. Hay dos tipos de posibles lectores del código fuente: programas y personas. A los programas les da igual la indentación, leen bien nuestro código siempre que cumpla la sintaxis del lenguaje. Luego la indentación debe centrarse en la lectura y comprensión del código por personas.

Para entender cómo hay que indentar un código primero hay que entender cómo lo lee una persona. Actualmente la mayor parte de las personas que leen código fuente lo hacen con editores de texto simples o con la ayuda de los editores de los entornos de desarrollo. Por tanto, nos vamos a centrar en este tipo de lectores.

Cuando una persona lee un código fuente, lo hace siguiendo una serie de patrones de lectura. Si queremos facilitar la lectura del código lo primero que hay que entender son estos patrones. La mejor bibliografía sobre patrones de lectura son los libros que enseñan sistemas de lectura rápida; dos puntos fundamentales de estos sistemas son el estudio de los patrones de movimiento de la vista y el reconocimiento de la estructura del texto.

Los patrones de lectura nos indican el movimiento que siguen los ojos al leer un texto. Cuando leemos un texto, la vista se desliza a saltos, no se desplaza de una manera continua. Nuestra vista capta la información y que haya alrededor del punto en que se detiene la vista. Un buen lector hace pocas paradas para leer una línea, aprovechando más su campo visual.

Por otra parte, al leer también utilizamos la visión periférica, aprovechando la estructura y disposición que tienen las letras para extraer información adicional del texto. Así, por ejemplo, instintivamente nos fijamos en el principio del párrafo para buscar la información relevante, y nos ayudamos de elementos como las listas, los tabuladores, tablas y otros elementos tipográficos al leer.

De estos puntos deducimos que la posición absoluta y relativa de los elementos que componen el texto del código fuente juegan un papel importante a la hora de facilitar la lectura y comprensión del mismo.

Requerimientos para definir un estilo de indentación

Vamos a enumerar los principales requisitos para poder definir un buen estilo de indentación. Algunos de los más importantes serán:

* Que el posicionamiento de los elementos dentro del texto sea predecible. Cuanto más predecible es la posición, el movimiento de los ojos llega antes a los puntos relevantes del código fuente. El punto más relevante de información es siempre el principio de la línea, por lo que hay que tener especial atención en él.
* Que la relevancia de la agrupación de los elementos refleje la relevancia de éstos dentro del programa.
* Prioriza la identificación del flujo del programa frente al detalle de los elementos que lo componen. Por ejemplo, es más importante siempre tener una visión global de lo que hace una función que del detalle de cómo se declaran sus variables.
* Que los comentarios respeten la indentación de los elementos a los que van asociados, posicionándose de manera relativa a ellos según su relevancia. Si el lenguaje lo permite, hay que distinguir la relevancia de un comentario posicionándolo sobre el elemento (más importante) o al lado de él (menos importante). Es mejor situar los comentarios antes del elemento, ya que nos sirven de introducción a lo que va a continuación, y muchas veces nos permitirá saltarnos un bloque de código que no nos interesa.
* Utilizar la indentación basada en tabuladores. La indentación basada en el uso de caracteres de espacio en blanco es menos flexible que la basada en tabuladores, ya que los editores y entornos de desarrollo modernos permiten adaptar con facilidad el ancho de los tabuladores al gusto de cada programador.
* Mantener el tamaño de las líneas dentro de los márgenes de la pantalla, siempre que sea posible. Debemos tener en cuenta al lector potencial del código fuente. No se debe pensar que el lector tiene una pantalla demasiado pequeña o grande, sino asumir que el lector tiene una pantalla típica dentro del entorno de desarrollo habitual. Por ejemplo, actualmente los desarrolladores suelen tener pantallas capaces de mostrar 40 líneas y 120 columnas de caracteres con facilidad. En este caso, utilizar el antiguo margen de 80 columnas por línea y 25 líneas por pantalla es más bien escaso.
* Utilizar el menor número de normas de indentación requerido. Indentar elementos en exceso complica visualmente la lectura del código, si no son estrictamente necesarios.