

# Unidad I: Conceptos fundamentales

## 1.1. Estilos de programación

Estilo de programación (también llamado *estándares de código* o *convención de código*) es un término que describe convenciones para escribir código fuente en ciertos lenguajes de programación.

El estilo de programación es frecuentemente dependiente del lenguaje de programación que se haya elegido para escribir. Por ejemplo el estilo del lenguaje de programación C variará con respecto al del lenguaje BASIC.

El buen estilo, al tratarse de algo subjetivo, es difícil de categorizar concretamente; con todo, existen un número de características generales. Con el advenimiento de software que da formato al código fuente de forma automática, el foco en cómo éste debe de ser escrito debe evolucionar de forma más amplia a la elección de nombres, lógica y otras técnicas. Desde un punto de vista práctico, el uso de un computador para dar formato al código fuente ahorra tiempo, y es posible forzar el uso de estándares a nivel de una compañía completa sin debates *religiosos*.

## 1.2. Evaluación de expresiones

Definiciones tomadas de news:comp.lang.functional

*Programación Funcional*: es un estilo de programación que enfatiza la evaluación de expresiones, en lugar de la ejecución de comandos. Las expresiones en estos lenguajes se forman utilizando funciones para combinar valores básicos.

*Lenguaje Funcional*: es un lenguaje que **soporta** e **incentiva** la programación en un estilo funcional.

## 1.3. Definición de funciones

Cuando escribas un nuevo programa o biblioteca, sigue un estilo consistente de ubicación de llaves y de indentación. Si no tienes ninguna preferencia personal de estilo, recomendamos el estilo de programación del núcleo de Linux o el estilo de programación de GNU.

Lee el nodo de info (Standards)Writing C en la documentación de GNU. Luego, obtén el código fuente de Linux y lee el archivo `linux/Documentation/CodingStyle`, e ignora los chistes de Linus. Estos dos documentos te darán una buena idea de nuestras recomendaciones para el código de GNOME.

### Estilo de indentación

Para el código del núcleo de GNOME preferimos el estilo de indentación del núcleo de Linux. Usa tabuladores de 8 espacios para la indentación.

Usar tabuladores de 8 espacios para indentación proporciona un número de beneficios. Permite que el código sea más fácil de leer, ya que la indentación se marca claramente. También ayuda a mantener el código ordenado forzando a dividir funciones en trozos más modulares y bien definidos — si la indentación va más allá del margen derecho, significa que la función está mal diseñada y que debiera dividirse para hacerla más modular o bien, repensarla.

Los tabuladores de 8 espacios para indentación también ayudan al diseño de funciones que encajen bien en la pantalla, lo cual significa que las personas puedan entender el código sin tener que desplazarse atrás y adelante para entenderlo.

## 1.4. Disciplina de tipos

En los lenguajes de programación de la familia C se recomienda también evitar el uso de caracteres tabulador en medio de una línea, ya que diferentes editores de textos muestran su anchura de forma diferente.

El lenguaje de programación Python usa indentación para indicar estructuras de control, por tanto se requiere obligatoriamente una buena indentación. Haciendo esto, la necesidad de marcar con llaves (`{}` y `}`) es eliminada, y la legibilidad es mejorada sin interferir con los estilos de codificación comunes. Con todo, esto lleva frecuentemente a problemas donde el código es copiado y pegado dentro de un programa Python, requiriendo un tedioso reformateado. Adicionalmente, el código Python se vuelve inusable cuando es publicado en un foro o página web que elimine el espacio en blanco.

## 1.5. Tipos de datos

Si usas Emacs, entonces puedes seleccionar el estilo de indentación del núcleo de Linux incluyendo en el archivo `.emacs` lo siguiente:

```
(add-hook 'c-mode-common-hook
  (lambda ()
    (c-set-style "k&r")
    (setq c-basic-offset 8)))
```

En los nuevos Emacs o con el nuevo `cc-mode`, puedes ser capaz de hacerlo más simple con:

```
(add-hook 'c-mode-common-hook
  (lambda ()
    (c-set-style "linux")))
```

El estilo de indentación de GNU es el predeterminado para Emacs, así que no es necesario agregar nada en el archivo `.emacs` para habilitarlo. Si deseas seleccionarlo explícitamente, sustituye «gnu» por «linux» en el ejemplo anterior.

Si usas Vim, entonces puedes seleccionar el estilo de indentación del núcleo de Linux incluyendo el siguiente fragmento en el archivo `.vimrc`:

```
set ts=8
if !exists("autocommands_loaded")
  let autocommands_loaded = 1
  augroup C
    autocmd BufRead *.c set cindent
  augroup END
endif
```

Como alternativa puedes seleccionar el estilo de indentación de GNU en Vim usando lo siguiente en el archivo `.vimrc`:

```
augroup C
  autocmd BufRead *.c set cinoptions={.5s,:.5s,+.5s,t0,g0,^-2,e-2,n-2,p2s,(0,=.5s formatoptions=croql cindent shiftwidth=4 tabstop=8
augroup END
```