Published on *Marco de Desarrollo de la Junta de Andalucía* (http://madeja.i-administracion.junta-andalucia.es/servicios/madeja)

Convenio de codificación general

- **Área:** Especificaciones de Codificación y Construcción
- **Tipo de pauta:** Directriz
- **Carácter de la pauta:** Obligatoria

Código: LIBP-0008

Las siguientes indicaciones ayudan a generar un código más eficiente, claro y fácil de mantener, siendo aplicables independientemente del lenguaje de programación que se utilice

Es muy común que el estilo de programación dependa, en gran medida, del lenguaje de programación que se haya elegido para desarrollar la aplicación, pero hay una serie de convenciones válidas para cualquier tipo de lenguaje.

Pautas

Título	Carácter
Nomenclatura de métodos, funciones y procedimientos	Obligatoria
Nomenclatura de variables	Obligatoria
Nomenclatura de constantes	Obligatoria
Tamaño del código fuente	Obligatoria
Tamaño de los métodos	Obligatoria
Tamaño de línea	Obligatoria
Expresiones largas	Obligatoria
Alineación	Obligatoria
Líneas en blanco	Obligatoria
Espacios en blanco	Obligatoria
Variables booleanas	Obligatoria
Estructuras de control lógicas	Obligatoria
Números mágicos	No Recomendada
Número máximo de parámetros	Obligatoria
Búsqueda por cadenas de texto	Obligatoria
Codificación UTF-8	Obligatoria
Código autodocumentado	Obligatoria
Aumento de la comprensión	Obligatoria
Comentarios por bloques	Obligatoria
Comentarios en elementos	Obligatoria
Instrucciones por línea de código	Obligatoria

Nomenclatura de métodos, funciones y procedimientos

Nombrar a los métodos, funciones y procedimientos describiendo la acción que realizan.

Los métodos, funciones y procedimientos se nombrarán describiendo la acción que realizan mediante el uso de verbos. La primera letra del nombre deberá estar en minúsculas. Si el nombre está formado por más de una palabra, a partir de la segunda palabra la primera letra deberá estar en mayúsculas.

Volver al índice

Nomenclatura de variables

Utilizar nombres de variables sencillos y comprensibles

Es necesario nombrar las variables de manera que facilite la comprensión del código fuente, manteniendo una cierta lógica dentro del modelo de negocio que representan. Se utilizarán sustantivos escritos en minúsculas. Si el nombre de la variable está formado por más de una palabra, a partir de la segunda palabra, la primera letra de cada palabra deberá estar en mayúsculas. De esta manera, el desarrollador obtendrá un código más intuitivo y sencillo de mantener.

Volver al índice

Nomenclatura de constantes

Nombrar las constantes siempre en mayúsculas.

Las constantes deben nombrarse todas en mayúsculas, separando las palabras que formen parte del nombre con el carácter de subrayado (_).

Volver al índice

Tamaño del código fuente

Controlar el tamaño del código en número de líneas

Se debe controlar el tamaño del código fuente estructurado. Éste no debe sobrepasar las **2000 líneas** de código ya que el superar esta extensión indicará que, probablemente, se está empleando un mal diseño.

Volver al índice

Tamaño de los métodos

No superar las **100 líneas** de código en los métodos

La longitud de un método no debe exceder de cien líneas de código sin una causa justificada, intentando no agrupar excesiva funcionalidad dentro del mismo. El superar esta longitud provoca que la detección de un error sea compleja, dificultando la legibilidad y el mantenimiento del código.

Volver al índice

Tamaño de línea

Usar una longitud de línea de no más de **100 caracteres**

Se deben programar líneas que, como máximo, tengan una longitud de **100 caracteres**, lo que hace que el código sea más práctico y eficiente. En cada línea sólo irá una instrucción, para evitar que la comprensión del código sea dificultosa.

Volver al índice

Expresiones largas

Cortar en diferentes líneas aquellas expresiones que sean muy largas

El corte en las expresiones muy largas se realizará de tal forma que no se pierda la legibilidad, dividiendo la línea tras una coma o antes de un operador.

Volver al índice

Alineación

Utilizar sangrías y tabulaciones para facilitar la lectura del código

Se debe usar un estilo de sangría en lenguajes de programación para tabular o delimitar bloques lógicos de código. Usando un estilo lógico y consistente se consigue que el código realizado sea más legible. El estilo de sangría se propone en los convenios específicos de cada lenguaje en particular.

Volver al índice

Líneas en blanco

Utilizar líneas en blanco para agrupar secciones de código que estén lógicamente relacionadas

Se deben emplear líneas en blanco para separar distintos bloques de código, de manera que se agrupen líneas de código que estén relacionadas. Por ejemplo, deben emplearse parar separar distintas declaraciones de clases o métodos entre ellas o, dentro de una clase, para separar la declaración de la última variable de la del primer método. Además, se deberán utilizar

líneas para separar los comentarios de cabecera del fichero del resto del contenido del fichero.

Volver al índice

Espacios en blanco

Espaciar el código para mejorar su comprensión

Se utilizarán espacios en blanco tras una palabra reservada y su paréntesis asociado. También entre una palabra reservada, o un paréntesis de cierre, y una apertura de llaves. Otros casos en los que se utilizarán un espacio en blanco son tras una coma en una lista o tras un punto y coma en una sentencia **for**. No se utilizará un espacio en blanco entre el nombre de un método y la apertura de su paréntesis o tras la apertura o cierre de paréntesis o corchetes.

Volver al índice

Variables booleanas

Usar variables booleanas en estructuras de decisión

En estructuras de decisión es necesario el uso de variables booleanas, ya que la no utilización de estas variables provoca que el código sea más complejo y difícil de entender, provocando incluso algunos errores en su concepción.

Volver al índice

Estructuras de control lógicas

Usar iteraciones para hacer el código más legible

El uso de estructuras de control lógicas para bucles también es parte de un buen estilo de programación. Ayuda a quien esté leyendo el código a entender la secuencia de ejecución (en programación imperativa).

Volver al índice

Números mágicos

Usar constantes en expresions condicionales o comprobación de código en vez de valores numéricos sin identificar.

Es obligatorio que, cada vez que se necesite hacer uso de un valor numérico concreto en el código, este valor se almacene en una constante. Esto indica el origen del valor numérico y reduce el coste de mantenimiento ante cualquier modificación que afecte a la estructura.

Volver al índice

Número máximo de parámetros

No sobrecargar métodos con más de diez parámetros

Las llamadas a los métodos no deben contener más de **10 parámetros** si no existe una causa justificada para ello. De esta manera, se asegura que los métodos no se encuentran sobrecargados de funcionalidad, favoreciendo el mantenimiento eficiente del código.

Volver al índice

Búsqueda por cadenas de texto

No hacer discriminación por mayúsculas, minúsculas o acentuación en la búsqueda de cadenas de texto.

Cuando se elaboren búsqueda por cadenas de texto no se debe hacer discriminación por mayúsculas, minúsculas o acentuación a la hora de obtener los registros asociados a la búsqueda.

Volver al índice

Codificación UTF-8

Usar codificación UTF-8 siempre que sea posible

Es obligatorio emplear la codificación de caractéres UTF-8 siempre que el fichero lo permita. El empleo de esta codificación facilita la portabilidad y evitará futuros problemas en la generación de código fuente documentado.

Volver al índice

Código autodocumentado

Desarrollar el código lo más autodocumentado posible

Hay que usar nombres sencillos y descriptivos a la hora de desarrollar el código para hacer la lectura mucho más fácil. Los comentarios deben ser sólo una herramienta para decir lo que no se puede decir en el código, de modo que se pueda entender lo que dice sin necesidad de indicarlo en otro lenguaje.

Volver al índice

Aumento de la comprensión

Proporcionar comentarios que aumenten la comprensión

Los comentarios deben favorecer la riqueza de la comprensión y no introducir una mayor complejidad en la comprensión del código.

Volver al índice

Comentarios por bloques

Realizar los comentarios por bloques de código

Los comentarios se realizarán por bloques, minimizando los comentarios por líneas, salvo aclaraciones en un código complejo o para reflejar la importancia de una línea dentro de la ejecución del código. Estós comentarios se harán al final de la línea.

Volver al índice

Comentarios en elementos

Comentar las interfaces, las clases y los métodos para documentar el código

Se comentarán las interfaces, las clases y los métodos para que el código quede documentado de forma clara y concisa, facilitando así la legibilidad y el mantenimiento del código.

Volver al índice

Instrucciones por línea de código

Cada línea contendrá una sóla instrucción

En cada línea sólo irá una instrucción, para evitar que la comprensión del código sea dificultosa.

Volver al índice

Pautas

Área: Desarrollo » Especificaciones de Codificación y Construcción

Código	Título	Tipo	Carácter
LIBP-0015	Convenio de codificación específico para Java	Directriz	Obligatoria
LIBP-0089	Convenio de codificación específico para PHP	Directriz	Obligatoria
LIBP-0009	Convenio de codificación específico para PL/SQL	Directriz	Obligatoria
LIBP-0124	Convenio de codificación específico para Drupal	Directriz	Obligatoria

Recursos

Área: Desarrollo » Especificaciones de Codificación y Construcción

Código	Título	Tipo	Carácter
RECU-0735	Implementación de convenios de codificación en general	Ejemplo	Obligatorio
RECU-0620	Implementación de convenios de codificación en Drupal	Referencia	Obligatorio
RECU-0734	Implementación de convenios de codificación en Java	Ejemplo	Obligatorio
RECU-0739	Implementación de convenios de codificación en PHP	Ejemplo	Obligatorio
RECU-0733	Implementación de convenios de codificación en PL/SQL	Ejemplo	Recomendado

Source URL: http://madeja.i-administracion.junta-andalucia.es/servicios/madeja/contenido/libro-pautas/8