Wikiprint Book

Title: Estándares de Desarrollo del Proyecto

Subject: Tibisay - Metodologia/Administracion/EstandaresDesarrollo

Version: 5

Date: 01/05/24 19:38:38

Table of Contents

Estándares de Desarrollo del Proyecto

Los estándares de desarrollo constituyen las normas o patrones de referencia que se deben implementar en el desarrollo de aplicaciones de software. Entre los estándares de desarrollo más comunes se encuentran: normas de codificación, normas y esquemas de seguridad, estándares de interfaz u/s, entre otros.

Normas de Codificación

- Cada archivo del proyecto deberá incluir al inicio del mismo, la licencia con la cual se distribuyen los códigos. El cuerpo de la licencia que se debe incluir es el siguiente:

```
/*
 * Tibisay Movil
 *
 * Copyright (C) 2013 Antonio Araujo (aaraujo@cenditel.gob.ve),
 * Jose Ruiz (jruiz@cenditel.gob.ve),
 *
 * CENDITEL Fundacion Centro Nacional de Desarrollo e Investigacion en
 * Tecnologias Libres
 *
 * Este programa es software libre; Usted puede usarlo bajo los terminos de la
 * licencia de software GPL version 2.0 de la Free Software Foundation.
 *
 * Este programa se distribuye con la esperanza de que sea util, pero SIN
 * NINGUNA GARANTIA; tampoco las implicitas garantias de MERCANTILIDAD o
 * ADECUACION A UN PROPOSITO PARTICULAR.
 * Consulte la licencia GPL para mas detalles. Usted debe recibir una copia
 * de la GPL junto con este programa; si no, escriba a la Free Software
 * Foundation Inc. 51 Franklin Street,5 Piso, Boston, MA 02110-1301, USA.
 *
 */
```

- Seguir y mantener un esquema de indentación en todos los códigos fuentes del proyecto. La unidad para mantener la indentación es el tabulador. Esto permitirá mejorar la legibilidad del código fuente por parte de los programadores.
- La codificación de caracteres a utilizar será `ISO-8859-1` o `latin1` para todos los archivos de texto.

Nombramiento

- Los nombres de las variables, clases y demás estructuras se utilizarán en inglés dentro del código fuente. Pero las vistas para el usuario estarán redactadas en español.
- Los nombres de clases deben comenzar con una letra mayúscula y utilizar letras mayúsculas como separador de palabras, el resto de las letras deben ir en minúscula. Por ejemplo: `SignatureDocument`.
- Los nombres de funciones deben comenzar con una letra minúscula y utilizar letras mayúsculas como separador de palabras, el resto de las letras deben ir en minúscula. Por ejemplo: `getDigneture()`.
- Las variables globales deben utilizar sólo letras mayúsculas y utilizar el caracter _ como separador de palabras. Por ejemplo: `VALID_TIME`
- Los nombres de otros archivos que no contengan definiciones de clases deben ser escritos en minúsculas y utilizar el caracter _ como separador de palabras. Por ejemplo `activity_main.xml`
- Los nombres de las variables deben ser escritos en minúsculas y utilizar el caracter _ como separador de palabras. Por ejemplo `pdf_version`

Líneas y espacios en blanco

- Dos líneas deberían siempre ser usadas en las siguientes circustancias: Two blank lines should always be used in the following circumstances:
  - Entre las distintas secciones del código
  - Entre las definiciones de clases e interfaces
- Una línea debería siempre ser usada en las siguientes circunstancias:
  - Entre dos métodos
  - Entre las variables locales de un método y su primera sentencia
  - Antes de un comentario o bloque de comentarios

- Entre las distintas secciones lógicas de un método

- Un espacio en blanco debería ser usado en las siguientes circunstancias:
  - Una palabra clave seguida por un paréntesis debería ser separada por un espacio. Por ejemplo

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

  - Un espacio debería estar después de las comas en una lista de argumentos
  - Todos los operadores excepto el . deberían ser separados de sus operandos por un espacio en blanco
  - Las expresiones en una sentencia `for` deberían ser separadas por un espacio en blanco

```
a += c + d;
a = (a + b) / (c * d);

while (d++ = s++) {
  n++;
}

printSize("size is " + foo + "\n");

for (expr1; expr2; expr3)
```

- Un espacio en blanco no debería ser usado entre el nombre de un método y su paréntesis de apertura
- Espacios en blanco no deben separar operadores unarios como el de incremento ("++"), y decremento ("--") y sus operandos