

Table of Contents

Estándares de Desarrollo del Proyecto	2
Normas de Codificación	2
Nombramiento	2

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.

[A continuación se deben identificar los estándares de desarrollo que serán implementados en el desarrollo de la aplicación]

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 implícitas garantías 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: `getDignature()`.
 - 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`
- La estructuras `if - else - for - while - do-while - switch - try-catch` deben tener un espacio entre la sentencia y la llave de apertura, deben además tener un espacio entre la palabra reservada de la estructura y los paréntesis que contienen los parámetros si estos están presentes, si la senetencia tiene una senetencia de extensión la llave de cierre debe compartir la misma linea con la sentencia de apertura de esta extensión, si por el contrario la senetencia no se extiende la llave de cierre debe estar sola en una linea. Por ejemplo:

```
if (condition) {
    statements;
} else if (condition) {
    statements;
```

```
} else {  
    statements;  
}  
  
try {  
    statements;  
} catch (ExceptionClass e) {  
    statements;  
}
```