

# **Sistema de Administración de Farmacias**

## **Estándar de Implementación**

### **Versión 1.1**

#### **Historia de revisiones**

Fecha	Versión	Descripción	Autor
26/08/2014	1.0	Redacción del documento	Manuela Viola
27/08/2014	1.1	Arreglos en redacción	Mª José Rabaza
28/08/2014	1.1	Revisión	Alfonso Methol

# ÍNDICE

<b>1. Convenciones Generales.....</b>	<b>3</b>
<b>2. Convenciones Específicas. ....</b>	<b>3</b>
2.1 Variables. ....	3
2.2 Constantes. ....	3
2.3 Indentación. ....	4
<b>3. Otros. ....</b>	<b>4</b>

# 1. Convenciones Generales.

Como convención general, los comentarios de cada método se deben realizar sobre la firma del mismo. Los métodos principales deben tener un comentario explicando qué hace y a cuál interfaz pertenece. Los demás comentarios deberán realizarse antes de la porción de código que éstos explican. Los comentarios podrán escribirse tanto en español como en inglés, sin embargo los nombres de los métodos y sus parámetros deberán ser en español.

## 2. Convenciones Específicas.

### 2.1 Variables.

Las variables deben ser declaradas en el momento de ser utilizadas por primera vez y siempre que sea posible se deben inicializar en la misma línea de declaración. Se deben usar nombres descriptivos que reflejen el significado de la misma.

Los nombres de las variables deben seguir la convención Camel Case, en donde la primera letra del identificador se pone en minúscula y la primera letra de cada palabra siguiente se pone en mayúscula. Por ejemplo: `isVarName`.

En lo posible, se debe evitar el uso de variables globales cuando en su lugar se podrían pasar como parámetros de la función.

Si se cree adecuado se puede agregar un comentario arriba de la variable explicando su existencia y uso.

### 2.2 Constantes.

Las constantes deben ser declaradas en mayúsculas ni bien se comienza el código (arriba de todo). Se deben usar nombres descriptivos y si se cree necesario se puede agregar un comentario en la línea anterior a su declaración (arriba de la constante) explicando el por qué su existencia y uso.

Los nombres de las constantes deben seguir la convención Barra Baja, en donde se debe poner un guión bajo (`_`) entre cada palabra del nombre de la constante. Por ejemplo: `ES_CONST_BARRA_BAJA`.

## 2.3 Indentación.

La medida de la indentación va a estar basada en el largo del tab para la herramienta de trabajo elegida para realizar la implementación. De esta forma si se debe indentar un lugar va a ser un tab, si se debe indentar dos lugares van a ser dos tab y así sucesivamente.

Para las estructuras de control de flujo se debe poner el corchete en la misma línea donde se declara la estructura y dentro del cuerpo se debe agregar un nivel de indentación. El corchete de finalización se debe poner con la misma indentación con la que comenzó la estructura. En el caso particular del if then else se debe poner el else en la misma línea donde se puso el corchete de finalización del if y abrir el corchete del else también en la misma línea.

Ejemplo:

```
int invento () {  
    int x = 3;  
    if (x == 3) {  
        int m = 0;  
        while (x != 0) {  
            x--;  
            m++;  
        }  
    } else {  
        m = 100;  
    }  
    return m;  
}
```

## 3. Otros.

Para realizar un código más prolijo y legible, en el momento de realizar operaciones aritméticas o lógicas debe haber un espacio entre los operandos y el operador. Por ejemplo: `x == 3`, `x && y`.

También debe haber un espacio entre el nombre de un método y sus parámetros y entre la coma de terminación de un parámetro y el comienzo del siguiente. Por ejemplo: `int sumar (int x, int y)`.