Proyecto TREA

Estándar de Implementación

Versión 2.1

**Historia de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 27/08/2014 | 1.0 | Versión inicial (Python) | Mathias Claassen |
| 29/08/2014 | 1.1 | Agrega sección JavaScript | Nahuel Hourcade |
| 30/08/2014 | 1.2 | Verificación SQA | Ari Chamlian |
| 06/09/2014 | 2.0 | Correcciones de documento | Nahuel HourcadeMathias Claassen |
| 14/09/2014 | 2.1 | Verificación SQA | Ari Chamlian |

**Contenido**

**1.** **Convenciones Generales 3**

**2.** **Convenciones Específicas 3**

2.1. Variables 3

2.2. Constantes 3

2.3. Identación 3

**3.** **Otros 3**

**4.** **Convenciones Específicas PARA JAVASCRIPT 4**

4.1. Variables, MÉTODOS Y FUNCIONES 4

4.2. Constantes 4

4.3. GUÍAS DE ESTILO 4

**1. Convenciones Generales**

En este proyecto, dado que se hace uso de varios lenguajes de programación, se aplicarán distintos estándares o variaciones para cada lenguaje. En general vale que las variables, los métodos y los comentarios deben estar escritos en inglés y en caso de las variables y los métodos, su nombre debe ser acorde a su uso o funcionalidad.

Cada método deberá ser debidamente documentado usando comentarios pertinentes al inicio del mismo. Se debe tener cuidado con dejar estos comentarios consistentes con la funcionalidad del método si éste sufre modificaciones posteriormente.

Para la parte Python del proyecto se seguirá a grandes rasgos el estándar PEP 8.

**2. Convenciones Específicas**

**2.1. Variables**

Las variables en Python serán escritas en minúsculas, y si se componen de varias palabras éstas serán separadas por “\_”.

Ejemplo de variables: x, index, x\_index.

**2.2. Constantes**

Las constantes en Python se escribirán completamente de mayúsculas. Si se desea componer una constante de varias palabras éstas serán unidas por "\_".

Las constantes se deberán definir al principio de cada archivo, luego de las importaciones.

Ejemplos: MAX\_LENGTH, KEY.

**2.3. Identación**

En Python la identación será de 4 espacios en blanco. Aparte se deberán dejar una línea vacía antes de cada método interno a una clase y dos líneas vacías antes de la definición de una clase.

**3. Otros**

En Python las importaciones se dividirán en tres grandes grupos y se deben importar en ese orden: librerías estándares, librerías de terceros, módulos locales al proyecto. Todas las importaciones se realizan al comienzo de cada archivo.

Los operadores (asignación, booleanos, binarios, etc.) deberán rodearse de un espacio en blanco. Como ejemplo: "x = 1; x += 3".

Se deberá escribir un "String de Documentación" al inicio de cada clase, función o método. Éstos constan de rodear un comentario explicativo de tres comillas ("""). Estos comentarios sirven para que cuando se importan esas funciones o métodos se pueda saber cuál es exactamente lo que hacen.

Además se deberán hacer comentarios en secciones que se consideren puedan ser complicados de entender para otro desarrollador y de lo posible estos comentarios se harán en una línea separada y no detrás de una línea de código.

Los nombres de módulos seguirán la convención de las variables mientras que las clases se deberán nombrar como una combinación de palabras de las cuales la primera letra se escribe en mayúscula (Ejemplo: Empleado, EmpAlmacen, EmpBanco).

**4. Convenciones específicas para JavaScript**

**4.1. Variables, Métodos y funciones**

Para declarar las variables siempre se utilizará el comando *var.* Esto evita que la variable tenga un alcance global, potencialmente ocultando otras variables. Tanto las variables, como los métodos y las funciones se nombrarán con minúsculas y si se compone de varias palabras la siguiente se concatenará directamente pero con la primera letra en mayúscula.

Ejemplo: variableNamesLikeThis, functionNamesLikeThis, methodNamesLikeThis.

**4.2. Constantes**

Para identificar constantes se usarán palabras en mayúscula unidas por un guión bajo.

Ejemplo: CONSTANT\_VALUES\_LIKE\_THIS.

Nunca usar la palabra clave 'const', no es soportada por Internet Explorer.

**4.3. Guías de estilo**

A continuación se describen algunos puntos importantes que se seguirán para dar formato al código en JavaScript.

**4.3.1 Comentarios**

Se harán comentarios en formato JsDoc sobre controladores y directivas generados para AngularJs, dando una breve descripción e indicando cuales son sus funciones y responsabilidades. También se agregarán comentarios en secciones de código que se consideren complicadas de entender.

**4.3.2 Finalizar las sentencias explícitamente**

Siempre usar punto y coma al final de una sentencia. El confiar en la finalización implícita de JavaScript puede causar confusión a la hora de debuguear.

**4.3.3 Iniciar bloque en la línea que se define**

Siempre se abrirán los corchetes '{' de un bloque en la misma línea que los abre. Por ejemplo:

|  |
| --- |
| *if (something) {**// ...**}* |

**4.3.4 Agrupar sentencias según lógica**

Usar líneas vacías para agrupar sentencias que tengan una lógica relacionada. Ejemplo:

|  |
| --- |
| *doSomethingTo(x);**doSomethingElseTo(x);**andThen(x);**nowDoSomethingWith(y);* |