Wikiprint Book

Title: Estándares de Desarrollo del Proyecto

Subject: Simulación - 2015/Metodologia/EstandaresDesarrollo

Version: 3

Date: 19/05/24 07:39:53

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

En el desarrollo del Sistema de Simulación se implementarán algunos estándares básicos para su codificación, los cuales contemplan lo establecido en la ■PEP-8 (Guía de estilo para código python):

Documentación de Código

Los archivos con extensión .py deberán contener la documentación de licencia, funciones, clases, atributos y métodos, de acuerdo a lo establecido en el estándar ■docstring para la documentación de código fuente python.

Cada archivo de python deberá contener, en la cabecera del mismo, una sección en donde se indica la licencia bajo la cual se distribuye el archivo de la siguiente forma:

```
"""
SICP - Simulador Integral de Cadenas Productivas

Copyright (C) 2015 CENDITEL nodo Mérida ■http://www.cenditel.gob.ve

Fundación Centro Nacional de Desarrollo e Investigación en Tecnologías Libres

Este programa es software libre; Usted puede usarlo bajo los términos de la licencia de software GPL versión 2.0 de la Free Software Foundation.

Este programa se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTIA; tampoco las implícitas garantías de MERCANTILIDAD o ADECUACIÓN A UN PROPÓSITO 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.
"""
```

Las clases deben contener, inmediatamente después de haber sido declaradas, la documentación relacionada a la misma contentiva de lo siguiente:

```
"""
@note <Descripción detallada sobre el objetivo de la clase>
@licence <Tipo de licencia de la clase>
@organization <Nombre de la institución u organización que desarrolla la clase>
@author <Nombre del autor que desarrolló la clase. En caso de ser varios autores, incorporar una línea con @author para cada uno de ellos>
@contact <Dirección de contacto de la persona que desarrolla la clase. El formato es nombre at dominio.com>
@date <Fecha en la que se crea la clase. En caso de modificación de la misma se incorporarán líneas adicionales con @date. El formato de fecha es YYYY-MM-DD>
"""
```

Antes de la declaración de atributos de una clase, se debe especificar una descripción breve sobre el mismo. Ejemplo:

```
#Descripción sobre el atributo de la clase que se esta declarando
atributo = <valor_por_defecto_del_atributo>
```

La documentación de los métodos de una clase y/o funciones debe ser indicada posterior a la declaración de la misma siguiendo el siguiente esquema:

```
"""
@note <Descripción detallada del método o función>
```

```
@licence <Licencia bajo la cual se distribuye el método o función>
@author <Autor que desarrollo la función>
@contact <Dirección de contacto del autor. Ej. nombre at dominio.com>
@param <Parámetro que recibe el método o función. Agregar tantos @param como parámetros contenga la función>
@return <Registros que retorna el método o función en caso de poseerlos, de lo contrario no se coloca este ítem dentro del comentario>
"""
```

Cualquier otra documentación que se desee agregar al archivo para resaltar algún proceso o procedimiento, se debe indicar siguiendo el siguiente esquema:

```
"""
<Descripción de una documentación extensa que plantee de manera detallada el proceso. Esta documentación debe comprender varias líneas>
"""
```

```
o
```

```
#<Descripción precisa del procedimiento en una sola línea>
```

La documentación a utilizar en los archivos de plantillas .html deberán llevar el siguiente esquema:

- Documentación de bloque o secciones dentro de etiquetas propias html se documentarán siguiendo los lineamientos de la ■W3C.
- Documentación de etiquetas o procesos del motor de plantillas de ■Django, deberá seguir el estándar establecido en la ■documentación del framework en su sección Comments.

Documentación de funciones javascript deberán seguir el siguiente esquema:

```
/**
* <Descripción detallada de la función>
*
* @author <nombre de la persona que crea la función>
* @param <nombre del parámetro que recibe la función> <Descripción breve del tipo de valor recibido>
* @date <Fecha de creación de la función>
* @return <descripción del valor o procedimiento que retorna la función>
*/
```

Documentación de hojas de estilo serán declaradas de la siguiente forma:

Encabezado:

```
/**
* @style <Descripción de la hoja de estilos>
* @media <Nombre del dispositivo de medios a utilizar para la hoja de estilo. Los valores son: screen|print|all>
* @version <Número de versión de la hoja de estilo>
* @author <Nombre del autor>
* @date <Fecha de creación de la hoja de estilo. El formato es YYYY-MM-DD>
*/
```

Secciones:

```
/**
* @section <nombre corto de la sección a describir>
*
* <Descripción detallada de la sección>
*/
```

Código Fuente

- **Identación:** La indentación del código se establece a 4 espacios, debido a la implementación de la indentación en python para definir sentencias y bloques de código dentro de ellas, es importante que la indentación del código sea realizada con espacios y no con el tabulador.

- **Variables:** El nombramiento de las variables será todo en minúsculas, y en casos en los cuales la misma se encuentre compuesta por varias palabras las mismas serán separadas por un guión bajo (_), ejemplo: palabra1_palabra2.

- **Clases:** Las clases se nombrarán siempre la primera letra en mayúscula y en caso de estar compuesta por otra frase o palabra se seguirá el mismo esquema, la primera letra de cada palabra en mayúscula. Cada clase debe contener un método __init__(self) que la inicialice, en caso de no contener instrucciones al momento de inicializar la clase, se debe agregar la palabra reservada de python "pass".

- **Métodos y/o Funciones:** Para el nombramiento de los métodos o funciones se indicará dicho nombre en letras minúsculas, en caso de estar compuesta por varias palabras, serán separadas por un guión bajo (_), ejemplo: palabra1_palabra2().