BeatIt!

Estándares de Documentación Técnica

Versión 1.2

Historia de revisiones

Fecha	Versión	Descripción	Autor
17/08/2014	1.0	Creación del Documento.	Luciana Martinez Joaquin Velazquez Cristian Bauza
19/8/2014	1.1	Ajuste a estándar de documentación.	Joaquin Velazquez
30/08/2014	1.2	Revisión de SQA	Pablo Olivera

Contenido

1 Diagramas

Se podrán utilizar todos los diagramas y artefactos disponibles en la versión 2.0 de UML y se deberá en la medida de lo posible seguir todos los estándares especificados por dicha versión.

1.1 Herramientas

Se utilizará la herramienta online DrawIO, que se ubica en www.draw.io.
La misma cuenta con persistencia de proyectos en Gdrive, en Dropbox o en Disco duro y además se integra a la cuenta de Gmail. Esto tiene la ventaja de evitar el envío de un mismo archivo a todos los integrantes del equipo.
Las diagramas incorporados son UML, notaciones para componentes de arquitectura, base de datos relacional y Azure.

1.2 Convenciones

a) El tipo de letra en todos los diagramas debe ser Helvética tamaño 12.

b) Clases: Los nombres de clases siempre deben estar en negrita, comenzar con la primera letra en mayúscula, deben ser simples y descriptivos. Ejem: Coche(), Vehiculo(), PruebaApplet().

c) Métodos: Deben comenzar con letra minúscula, y si está compuesta por 2 palabras, la primera letra de la segunda palabra debe comenzar con mayúscula. De preferencia que sean verbos. Ejem: arrancarCoche(), sumar(). En cuanto a sus parámetros, en caso de que los tengan, deberán escribirse usando la convención Camel Case junto con su tipo de dato.

d) Variables: las variables siguen la misma gramática que los métodos. Se debe evitar usar variables de una sola letra (a, b, c.). Las variables constantes o finales, las cuales no cambian su valor durante todo el programa se deben escribir en mayúsculas. Ejemp: ANCHO, VACIO.

e) Siempre indicar cuando un atributo o método es público/a o privado/a.

f) En todos los diagramas UML se debe especificar los tipos tanto de los atributos como de los parámetros de los métodos.

g) En los diagramas de clase no será necesario indicar los getters setters de los atributos así como tampoco se deberá especificar los constructores.

h) En las asociaciones siempre se deberá indicar la multiplicidad de las mismas.

2 Subsistemas

a) El tipo de letra a utilizar debe ser Helvética tamaño 12.

b) Utilizar la misma nomenclatura que la utilizada para definir las clases (deben estar en negrita, comenzar con la primera letra en mayúscula).

c) Los comentarios deberán ser lo más breves y concisos posibles, y siempre deberán ir dentro del artefacto "Note" definido en UML y que se encuentra disponible en la herramienta seleccionada para realizar los diagramas.

d) Los subsistemas deben ser de fácil mantenimiento evitando redefinición de elementos en común para apuntar a la reutilización y además no establecer comunicación redundante entre ellos para obtener un mejor rendimiento.

e) En la medida de lo posible adaptar los subsistemas a patrones arquitectónicos ya conocidos.

f) Dejar en claro cómo se realiza la interacción entre los subsistemas.

3 Componentes

a) El tipo de letra a utilizar debe ser Helvética tamaño 12.

b) Utilizar la misma nomenclatura que la utilizada para definir las clases (deben estar en negrita, comenzar con la primera letra en mayúscula).

c) Los comentarios deberán ser lo más breves y concisos posibles, y siempre deberán ir dentro del artefacto "Note" definido en UML y que se encuentra disponible en la herramienta seleccionada para realizar los diagramas.

d) Identificar los componentes de infraestructura.

e) Clases altamente relacionadas pertenecen al mismo componente.

f) El acoplamiento entre componentes deberá mantenerse lo más bajo posible, e intentar reducir el flujo de mensajes entre los componentes.

4 Interfaces

a) El tipo de letra a utilizar debe ser Helvética tamaño 12.

b) Los nombres de las interfaces deben comenzar con "I" y al igual que las clases deben ser simples y descriptivos.

c) Los nombres de las clases que implementan una interfaz deben tener relación con el nombre de dicha interfaz.