

where2night	Versión: 1.0
Guía de estilos	Fecha: 18/11/2013

UCM-FDI

where2night
Guía de estilos

Versión 1.0

where2night	Versión: 1.0
Guía de estilos	Fecha: 18/11/2013

Historial de revisiones

Fecha	Versión	Descripción	Autor
18/11/2013	1.0	Creación Documento	Luis Cubero

where2night	Versión: 1.0
Guía de estilos	Fecha: 18/11/2013

Tabla de contenidos

[1. Introducción](#)

[1.1 Propósito](#)

[2. Guía documentación](#)

[2.1 Formato](#)

[2.2 Especificaciones generales](#)

[3. Guía programación:](#)

[3.1 Formato de líneas](#)

[3.2 Ficheros](#)

[3.3 Clases](#)

[3.4 Identificadores](#)

[3.5 Sentencias de control](#)

[3.6 Documentación](#)

where2night	Versión: 1.0
Guía de estilos	Fecha: 18/11/2013

1. Introducción

1.1 Propósito

El propósito de éste documento es facilitar la labor de dar un formato a nuestros documentos por ese motivo vamos a describir unos estilos que serán implantados entre los miembros del grupo para la creación de documentación y desarrollo del software.

2.Guía documentación

2.1 Formato

Todos los documentos de nuestro proyecto(visión,gestión y planificación,requisitos...) estarán estructurados de la siguiente forma:

- Portada: incluye nombre de grupo, proyecto, tipo documento y versión
- Historial de revisiones: fecha, versión, breve descripción de las modificaciones y los autores
- Tabla contenidos: índice que será generado automáticamente.

Todas las hojas contendrán una cabecera en la que se indica el nombre del proyecto, fecha y versión de la última modificación junto con el tipo de documento.

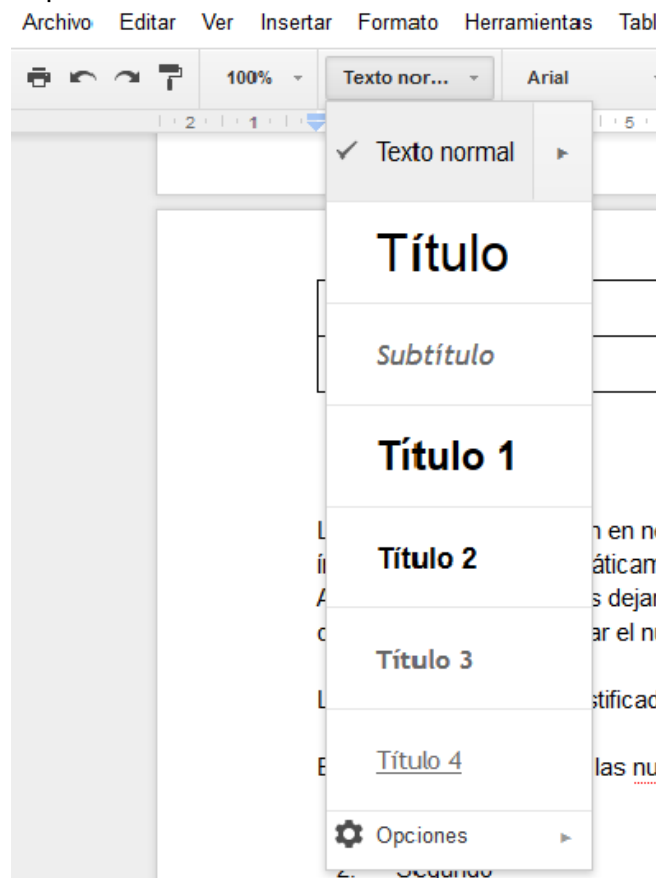
A pie de página se indicará el número de hoja del documento.

2.2 Especificaciones generales

En primer lugar vamos a definir el texto normal que será del tipo de letra Arial y tamaño 11, es el tipo que viene predefinido por google docs. El tipo de letra se aplicará a todo el documento incluyendo los títulos y subtítulos con la salvedad de que estos modificarán su tamaño.

where2night	Versión: 1.0
Guía de estilos	Fecha: 18/11/2013

Los títulos y subtítulos irán en negrita y su tamaño será respectivamente 18 y 13. Para que el índice se actualice automáticamente hay que usar Título 1 y 2 que deben estar configurados con el tipo de letra.



Antes de cada título debes dejar dos líneas en blanco para que el texto se vea más ordenado y claro, así el podrá destacar el nuevo apartado.

La alineación debe ser justificada a la izquierda y el interlineado de 1.15.

El uso de las listas serán las enumeradas, ejemplo:

1. Primero
2. Segundo
3. Tercero
 - 3.1. Tres punto 1
 - 3.1.1.
 - 3.1.2.

where2night	Versión: 1.0
Guía de estilos	Fecha: 18/11/2013

Se aconseja que cuando se vaya a empezar un nuevo documento coja la base de éste o de cualquier otro ya terminado y lo modifique encima para que la configuración sea la correcta

3. Guía programación:

La guía está enfocada para el lenguaje java pero las ideas se pueden extraer para cualquier otro, es importante reseñar que todo el código y comentarios deberá realizarse en inglés.

3.1 Formato de líneas

No usar más de 80 caracteres por línea, de esta forma se puede visualizar el código correctamente desde un editor de texto o una hoja impresa de tamaño DIN4

Cada vez que se comience un nuevo bloque de instrucciones debe estar sangrado adecuadamente, a la hora de sangrar es recomendable utilizar entre 2-4 espacios en vez de una tabulación para evitar incompatibilidades entre editores de texto.

3.2 Ficheros

Incluir una sola clase o interfaz por fichero, si es necesario realizar algún comentario global deberá de realizarse al principio.

Si la clase forma parte de algún paquete deberá ser mencionado al principio y a continuación los paquetes importados si fuera necesario.

3.3 Clases

En primer lugar irán los comentarios de la clase, las variables irán ordenados de la siguiente forma:

1. Estáticas
2. Públicas
3. Protegidas
4. Privadas

where2night	Versión: 1.0
Guía de estilos	Fecha: 18/11/2013

A continuación se declararán las constructoras, y lo seguirán las funciones ordenadas según su interrelación y no por visibilidad. Por ejemplo si un método público A utiliza dos privados B y C , A irá en primer lugar; esto se hace de esta forma para leer el código y sea más fácil realizar un seguimiento de su funcionalidad.

Es importante no dejar un espacio entre el nombre de la función y el paréntesis de los parámetros.

3.4 Identificadores

Escoger identificadores significativos y lo más breves posible

Las clases comenzarán con las iniciales de cada palabra en mayúscula: `class UserParty;`

Las variables irán en minúscula en caso de necesitar dos palabras se usará “_” : `int id_local;`

Las constantes se representan igual que las variables pero en mayúsculas.

Los métodos deben ser nombrados a ser posibles con un verbo si se precisan más palabras irán con la inicial en mayúsculas: `public void getUser(int value){`

3.5 Sentencias de control

Colocar la apertura de bloque ({) al final de la línea inicial de la sentencia; el fin de bloque (}) colocarlo en una línea a parte, si la sentencia es un if- else, el else debe situarse en la siguiente línea de fin de bloque:

```
if (a==0){
    date=6;
}
else {
    date=2;
}
```

where2night	Versión: 1.0
Guía de estilos	Fecha: 18/11/2013

Aunque solo haya una sentencia en el bloque colocar los paréntesis.

3.6 Documentación

Documentar el código nos puede facilitar a guiarnos y a aclararnos de la finalidad de la clase o función por ese motivo se aconseja:

1. En la cabecera añadir una pequeña descripción que incluya información acerca de su objetivo.
2. Indicar la fecha de modificación y sus autores, por si surgiera alguna duda para tenerlos identificados y facilitar la comunicación con ellos.