

MÓDULO PROFESIONAL ENTORNOS DE DESARROLLO

UD 4.2 – Documentación

CONTENIDO

- Documentar.
- Tipos.
- Formatos.

DOCUMENTACIÓN

Un sistema software que sea:

- Difícil de comprender.
- Sólo utilizable por sus realizadores.
- Difícil de modificar.

NO ES VÁLIDO

DOCUMENTACIÓN

En todo **proyecto de desarrollo de software** es muy importante la documentación, no solo para los usuarios finales, también para los propios desarrolladores: tanto para uno mismo, como para el resto del equipo de desarrolladores y los futuros miembros del mismo.

Ya veremos como ayuda la **refactorización** en el entendimiento del código, pero además es también comprensible con una buena **documentación**.

TIPOS DE DOCUMENTACIÓN

Tipos de documentación

Documentación interna: va vinculada con el código.

- Documentación de código (para los desarrolladores).

Documentación externa: localizada aparte del programa en el sentido que no forma parte del código.

- Documentación técnica (para los administradores).
- Documentación de usuario (para los usuarios).

DOCUMENTACIÓN DE CÓDIGO

Un complemento muy útil para los desarrolladores es la documentación del código fuente que se está implementando.

Existen muchas formas diferentes de documentar con muchos niveles diferentes de profundización. La opción más sencilla es la primera que se aprende cuando se empieza a programar, que son los **comentarios a lo largo del código fuente**.

Documentar el código de un programa es dotar al software de toda la información que sea necesaria para explicar lo que hace. Los desarrolladores que llevan a cabo el software (y el resto del equipo de trabajo) deben entender qué está haciendo el código y el porqué.

DOCUMENTACIÓN DE CÓDIGO

Hemos hablado antes de los **comentarios** y señalado que si un código necesita ser comentado es porque no es suficientemente descriptivo por si solo y necesita ser **refactorizado**. Aunque esto sea efectivamente así, el uso de comentarios sigue siendo una buena **ayuda al entendimiento del código**.

Además, se utiliza de manera recurrente con el fin de omitir algunas instrucciones para que no se ejecuten, muy importante en un proceso de depuración.

DOCUMENTACIÓN DE CÓDIGO

Los comentarios se encuentran intercalados con las sentencias de programación, de hecho se consideran parte del código fuente. Son pequeñas frases que explican pequeñas partes de las sentencias de programación implementadas.

Los comentarios no son tenidos en cuenta por parte de los compiladores a la hora de convertir el código fuente en código objeto y, posteriormente, en código ejecutable. Esto da libertad al programador para poder escribir cualquier cosa en ese espacio que él habrá indicado que es un comentario, sin tener que cumplir ninguna sintaxis específica.

DOCUMENTACIÓN DE CÓDIGO

Existen **diferentes formas de implementar comentarios**, irá en función del lenguaje de programación que se utilice. A continuación se muestra la forma de implementar comentarios con Java.

En el lenguaje **Java** existen **tres formas** de poner comentarios.

- la forma de mostrar **comentarios intercalados con el código fuente** es añadiendo los caracteres `//`. Esto indicará que a partir de ese punto todo lo que se escriba hasta el final de la línea estará considerado en comentario y el compilador no lo tendrá en cuenta.

```
1. // Comentario de una línea
```

DOCUMENTACIÓN DE CÓDIGO

- Otra forma de mostrar comentarios es añadiendo los caracteres `/*...*/`, Donde el comentario se encontraría ubicado en el lugar de los puntos suspensivos y, a diferencia de la anterior, se pueden escribir **comentarios de más de una línea**.

```
1. /* Comentario  
2. de varias  
3. líneas */
```

- El último caso son los comentarios para la **herramienta de documentación JavaDoc**. En este caso, antes del comentario pondremos una barra inclinada y dos asteriscos (`/**`) y finaliza con un asterisco y una barra inclinada (`*/`)

```
1. /** Comentario para JavaDoc */
```

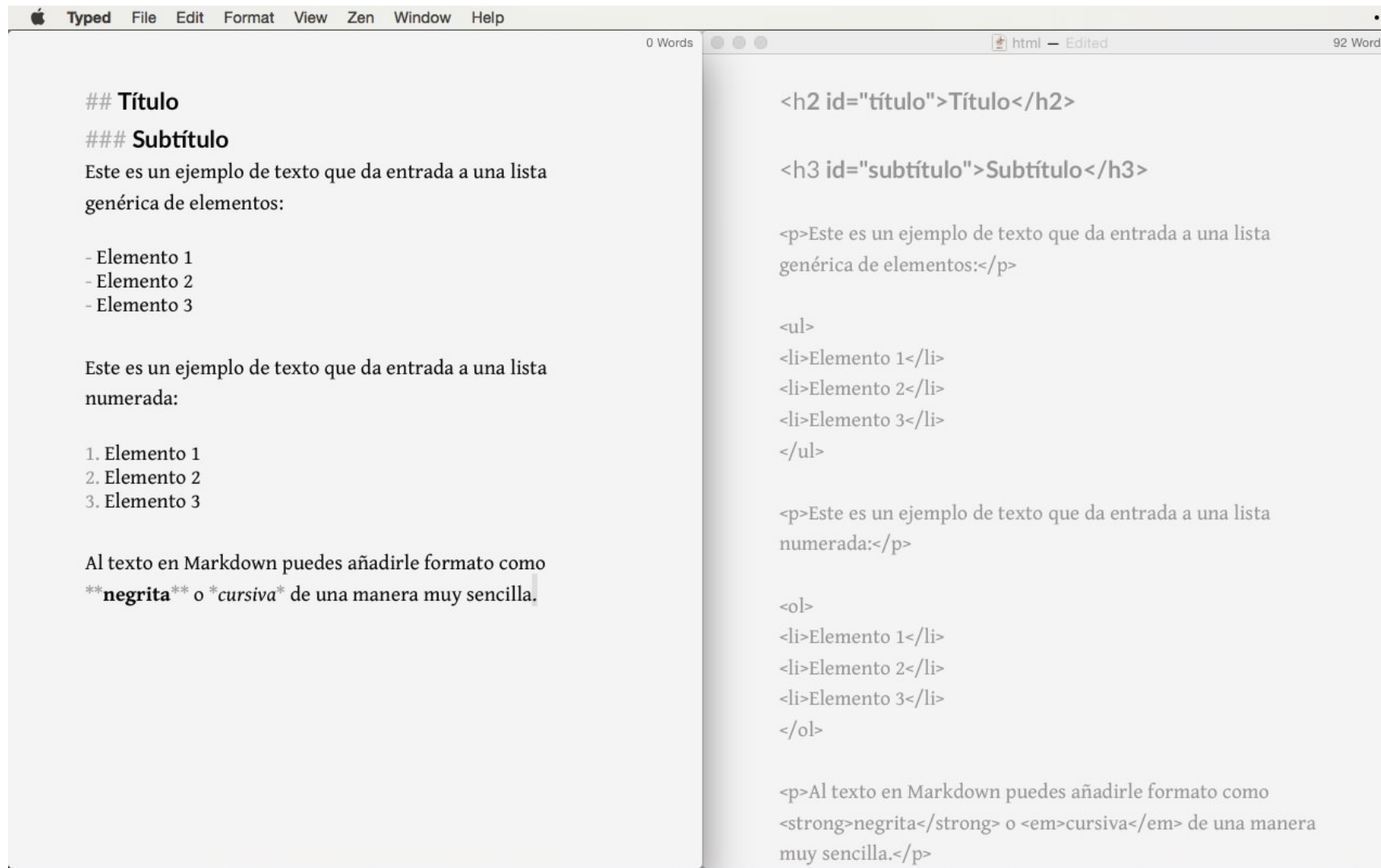
¿QUÉ HAY QUE DOCUMENTAR?

- Añadir explicaciones a todo lo que **NO** es evidente.
- ¿De qué se encarga una clase? ¿un paquete?
- ¿Qué hace un método?
- ¿Cuál es el uso esperado de un método?
- ¿Para qué se usa una variable?
- ¿Cuál es el uso esperado de una variable?
- ¿Qué algoritmo estamos usando? ¿de dónde lo hemos sacado?
- ¿Qué limitaciones tiene el algoritmo?
- ¿Qué se debería mejorar ... si hubiera tiempo?

FORMATOS DE DOCUMENTACIÓN

Formatos de documentación

- **HTML** (p. ej. Javadoc)
- **Markdown** (p. ej. Gitbook): es un lenguaje de marcado ligero. Una cantidad de sitios como GitHub, Reddit, Diaspora, Stack Exchange, OpenStreetMap, y SourceForge usan algunas variantes de Markdown para facilitar la discusión entre usuarios. Markdown convierte el texto marcado en documentos XHTML utilizando html2text.
- **reStructuredText** (p. ej. Readthedocs): lenguaje de marcas ligero creado para escribir textos con formato definido de manera cómoda y rápida. Es parte del proyecto Docutils dentro de la comunidad de Python. Tiene la principal ventaja de que ese texto puede usarse para generar documentos equivalentes en HTML, LaTeX, docbook, etc
- **asciidoc**: formato de documento de texto para escribir notas, documentación, artículos, libros, libros electrónicos, presentaciones de diapositivas, páginas web, páginas de manual y blogs. Los archivos AsciiDoc se pueden traducir a muchos formatos, incluyendo HTML, PDF, EPUB, página de manual.



Observa la diferencia entre sintaxis **Markdown** (izquierda) y sintaxis **HTML** (derecha), ¿con cuál te quedas?