

Despliegue de aplicaciones web

UD 02. Cómo documentar tus aplicaciones

Actualizado Septiembre 2023

Licencia



Reconocimiento – NoComercial - Compartir Igual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

ÍNDICE DE CONTENIDO

1. Introducción	4
2. PlantUML	4
3. Doxygen	4
4. Markdown	4
5. PHPDocumentor, JavaDoc y JSDoc	5
6. RAML	5
7. Bibliografía	5
8. Autores (en orden alfabético)	6

1. INTRODUCCIÓN

La documentación de aplicaciones de software es una parte esencial del proceso de desarrollo de software y desempeña un papel crucial en la comprensión, mantenimiento y colaboración en proyectos de software. La documentación proporciona una guía detallada sobre cómo funciona una aplicación, cómo se instala, cómo se utiliza y cómo se mantiene. Además, facilita la comunicación entre los miembros del equipo de desarrollo, los usuarios finales y otras partes interesadas.

A continuación, veremos diferentes herramientas y lenguajes empleados en la documentación de Software:

2. PLANTUML

El análisis de una aplicación conlleva la creación de diferentes diagramas: casos de uso, entidad/relación, diagrama de clases, de objetos, de despliegue, etc.

Además, al desarrollar una aplicación trabajamos con otro tipo de elementos, como JSON (formato de intercambio de datos estándar más utilizado, otro formato de datos muy empleado para ficheros descriptores y ficheros de configuración de una aplicación: YAML.

PlantUML es una aplicación web que te permite crear diagramas de forma rápida y sencilla, conocer y trabajar con JSON, YAML, crear diagramas de Gantt para ver la temporalización y dedicación en un proyecto etc.

<https://plantuml.com/es/>

3. DOXYGEN

Doxygen es una herramienta utilizada en el desarrollo de software para generar documentación técnica de código fuente. **Su principal objetivo es automatizar la generación de documentación a partir del código fuente y los comentarios incrustados en él.**

Doxygen es ampliamente utilizado para documentar programas y bibliotecas de software en varios lenguajes de programación, incluyendo C++, C, Java, Python, PHP, y más.

<https://doxygen.nl/>

4. MARKDOWN

Markdown es un lenguaje de marcado ligero que se utiliza comúnmente para formatear y estructurar texto de manera sencilla y rápida en documentos de texto plano. Tiene su propia sintaxis y es muy utilizado en software para escribir la documentación técnica del programa. Por ejemplo, los típicos *readme.md* están escritos con el lenguaje Markdown.

Su uso es debido a la gran estandarización del lenguaje, su integración en IDEs (como Visual Studio

Code) y su portabilidad, ya que es un lenguaje multiplataforma. Además, es muy utilizado en blogs ya que es posible transformar Markdown a HTML, por lo tanto, es posible crear entradas HTML sencillas con Markdown.

<https://www.markdownguide.org/>

5. PHPDOCUMENTOR, JAVADOC Y JSDOC

Son herramientas de generación de documentación utilizada específicamente en el contexto de desarrollo de software en PHP, Java y Javascript. Su principal función es generar documentación técnica automáticamente a partir del código fuente y los comentarios incrustados en él.

Básicamente funciona como Doxygen pero PHPDocumentor está orientado únicamente a PHP, JavaDoc a Java y JSDoc a Javascript.

PHPDocumentor: <https://www.phpdoc.org/>

JavaDoc: <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>

JSDoc: <https://jsdoc.app/>

6. RAML

RAML (RESTful API Modeling Language) es un lenguaje de modelado específicamente diseñado para describir APIs (Interfaces de Programación de Aplicaciones) RESTful de manera clara y estructurada. RAML permite a los desarrolladores y equipos de desarrollo definir y documentar sus API de manera efectiva, lo que facilita la comprensión, el diseño y la implementación de servicios web REST.

<https://raml.org/>

7. BIBLIOGRAFÍA

[1] <https://plantuml.com/es/>

[2] <https://doxygen.nl/>

[3] <https://www.markdownguide.org/>

[4] <https://www.phpdoc.org/>

[5] <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>

[6] <https://jsdoc.app/>

[7] <https://raml.org/>

8. AUTORES (EN ORDEN ALFABÉTICO)

A continuación ofrecemos en orden alfabético el listado de autores que han hecho aportaciones a este documento.

- Miguel Mira