

Guía para la documentación de desarrollo de software

MONI-SIGO

2021

Arquitectura del sistema

La información sobre la arquitectura debería incluir como mínimo:

- **Diagrama de módulos:** Consiste en un diagrama donde se representan todas las partes que componen el sistema y las relaciones que existen entre estas. El objetivo de este diagrama consiste en presentar una perspectiva global de la arquitectura y los componentes del sistema, no debería contener detalles técnicos sobre los módulos o las conexiones entre estos.
- **Descripción individual de los módulos:** Para cada módulo o parte del sistema, se debería realizar una breve descripción del mismo, la cual debería incluir mínimamente:
 - **Descripción general y propósito:** *¿qué es y qué debería hacer el módulo?*
 - **Responsabilidad y restricciones:** *¿cuál es su función específica dentro del sistema? ¿qué cosas puede y no puede hacer?*
 - **Dependencias:** Indicar cuales son los requisitos del módulo, es decir se debe contestar a preguntas tales como *¿qué necesita o requiere el módulo para funcionar? ¿necesita de servicios brindados por otros módulos o por librerías externas?*
 - **Implementación:** indicar en qué archivo o archivos se encuentra la implementación del módulo.

No es el objetivo de esta sección dar detalles de *cómo* se realiza la implementación de los módulos, sino únicamente dar una idea general de *para qué* existe el módulo dentro del sistema.

- **Dependencias externas:** Si el software utiliza librerías o servicios externos estos deben listarse junto con una breve descripción de las mismas.

Adicionalmente en esta sección se deben listar los **aspectos técnicos o tecnologías empleadas** en el proyecto, tales como el lenguaje de programación, *frameworks*, librerías, etc. Puede resultar de utilidad incluir junto a estos una breve descripción de las decisiones de diseño asociadas que llevaron a elegir la o las tecnologías en particular, es decir responder a *¿por qué se utilizó esta tecnología y no otra?*

1. Diseño del modelo de datos

Distinguir cuáles son las entidades involucradas en el sistema y mencionarlas en un formato *language-agnostic*.

Puede ser un diseño orientado a objetos, relacional, etc., lo importante es tener una idea general del modelo de datos: entidades, atributos y las relaciones entre ellas. Para ello es imprescindible incluir diagramas o gráficos que ayuden a visualizar el modelo de datos.

Un programa, aplicación o librería puede a su vez trabajar con varios tipos de datos:

- Datos de entrada.
- Datos internos.
- Datos de salida.

Distinguir claramente cada uno de ellos y describir su modelo.

1. Descripción de procesos y servicios ofrecidos por el sistema

Mencionar cuáles son los servicios o tareas que el sistema ofrece/implementa, y describir los procesos que realizan, para entender cómo funcionan, y cómo se pueden invocar/utilizar.

Para este propósito es conveniente incluir pseudo-algoritmos, diagramas de flujo, etc.

Tener presente que la descripción del proceso no significa mostrar el código, ni consiste tampoco brindar detalles específicos de cómo lo hace (funciones utilizadas para hacer cierta tarea) sino de explicar brevemente qué hace o cuál es su propósito. Se espera también una descripción de los datos de entrada y salida (Cantidad de argumentos, tipo y significado de cada uno).

En este punto es imprescindible que el código fuente esté enriquecido con comentarios. Estos deben conformar la documentación básica de todo proyecto, y a partir de los mismos debería poder construirse la descripción de alto nivel del funcionamiento de los procesos y servicios del sistema, así como sus funciones, subrutinas, módulos, clases, etc.

2. Documentación técnica - Especificación API

Se indica el propósito y breve descripción de cada método/función, con su prototipo indicando argumentos (nombre, tipo, propósito de cada uno) y respuesta (tipo, descripción).

Para llevar a cabo esta tarea, es posible utilizar una variedad de herramientas de generación de documentación automática, a partir del código en el encabezado de cada función (por ejemplo Javadoc, PHPDoc, Doxygen, etc).

La documentación técnica debe pensarse como el manual del programador, y debe apuntar a aquellas personas que estarán a cargo de mantener, ampliar, o crear un proyecto derivado a partir de nuestro proyecto.

Aspectos relevantes

- Indicar claramente cómo invocar el programa (signatura del programa completa, como la que haría cualquier sinopsis de una página de manual), conteniendo qué parámetros son opcionales, cuales son obligatorios, y documentar bien cuál es la utilidad de cada parámetro y cuál es el comportamiento por defecto si se omite algún parámetro opcional. Esto conforma comúnmente el manual del usuario final de la aplicación.
- Incorporar diagramas de flujo y explicaciones a nivel método de la solución, debe explicarse la estrategia general de resolución donde se pueda apreciar cómo interactúan los módulos entre sí.
- Los tipos de datos abstractos (TDAs) deben estar adecuadamente documentados en el código, por otra parte, en el manual deben constar las limitaciones posee la representación, cómo se representa una determinada estructura y detalle de métodos que provee el TDA para la manipulación de los datos.
- Incluir una sección de “Conclusiones”, donde se deben resumir complicaciones encontradas durante el desarrollo del proyecto, políticas adoptadas para su resolución, restricciones al problema original, casos particulares y finalmente aspectos relacionados a la experiencia obtenida en base a la temática del proyecto.