Introducción

La documentación técnica es muy importante en el desarrollo de software. Es como una carta de navegación para tu equipo. Documentar tu proceso sirve como referencia explicando las razones del desarrollo, como opera y cómo utilizarlo. Los equipos de software se refieren a este proceso de documentación técnica cuando hablan de requerimientos, notas en la release o diferentes aspectos en el desarrollo del producto. Utilizan documentos para detallar el código, APIs base de datos, y realizar un seguimiento en el proceso de desarrollo de software. Externamente, la documentación se plasma en manuales, guías de usuario para administradores de sistemas, ayuda a los equipos y demás usos.

La documentación técnica de un proyecto de desarrollo de software no debe omitir la información de la base de datos. Es importante conocer con qué tipo de base de datos estamos trabajando, servidores, diagrama de información, estructura de tablas, así como los tipos de datos utilizados, etc.

Para un programador, la **documentación técnica es un deber**. La tarea no se basa en hacerlo o no, sino en cómo y qué herramientas sirven para hacer el proceso más eficiente. El hecho de documentar el proceso sirve para conocer todos los aspectos de una aplicación, cosa que beneficia al equipo y mejora la calidad del producto de software. Con la documentación técnica, tienes toda la información necesaria para el desarrollo y el mantenimiento correcto de la aplicación. Así como una mejor transferencia de conocimiento entre developers. Incluso el mejor software, puede ser inútil si los desarrolladores son incapaces de entenderlo.

Una buena documentación técnica con las mejores herramientas hace que la información sea más accesible, ofreciendo un gran número de "entry points", ayuda a los nuevos desarrolladores a aprender más rápido, simplifica el producto y ayuda en la documentación de costes. Además, genera confianza.

Por lo tanto, la documentación técnica sirve para ayudar a los nuevos miembros del equipo a adaptarse más rápido a los hábitos de trabajo

de la organización. Comparte información del funcionamiento del producto y el porqué de cada requerimiento. Hace que la curva de aprendizaje de los desarrolladores sea más suave. Y lo hace señalando aquellos puntos de la aplicación en los que ha de centrarse el desarrollador para saber más del contexto de aquella aplicación en la que están trabajando.

Por lo anterior, a continuación, se especifican los puntos necesarios que debe cumplir la documentación técnica de cada proyecto elaborado en la Unidad de Informática.

1.- Definiciones

1) Definición general del proyecto de software

Explicar en qué consiste el sistema o desarrollo en cuestión, idea general, funcionalidad principal del proyecto de software, propósitos y objetivos del desarrollo.

Se debe atender:

- Funcionalidad principal del sistema, el ¿Qué?
- Objetivos del desarrollo, y la necesidad cubierta por el sistema en cuestión, el ¿Para qué?
- Usuarios: personas o entidades que utilizarán el sistema o parte de él, y el nivel de experiencia del usuario hacia el cual el presente informe está dirigido, el ¿Quién?

2) Especificación de requerimientos del proyecto

Se deben incluir los detalles de los requerimientos técnicos y generales del mismo, los alcances y limitaciones de la implementación realizada.

Aclarar, si el proyecto de software forma parte de algún sistema ya desarrollado; de ser el caso, especificar se desarrolló una nueva versión o es una derivación.

Sobre esta especificación, se dará información del proyecto de software sobre los siguientes puntos:

- Requisitos generales: Las pautas y consignas que sigue el proyecto de software.
- Requisitos funcionales: los servicios que el sistema proporciona, las tareas que éste desarrolla.
- Información de autoría y legales del proyecto: Explicitar si el proyecto de software forma parte de desarrollos previos/preexistentes o si es original, y en el caso correspondiente, detalles de retro-compatibilidad.

3) Procedimiento de instalación y prueba: detallar cómo se realiza la obtención, instalación y/o prueba del sistema, junto las especificaciones generales de la plataforma o el entorno sobre cual el software debe ser ejecutado.

De las especificaciones de los procedimientos, se atenderá

- Herramientas utilizadas: Entornos de desarrollo integrados, plataformas y herramientas empleadas en la implementación del sistema.
- Planificación: Una descripción global de la metodología utilizada para encarar y resolver el problema; por ejemplo: los pasos ejecutados a lo largo de la resolución del proyecto, a grandes rasgos.

De la instalación y prueba:

- Requisitos no funcionales: Si los hubiese, restricciones que afectan el desempeño normal del sistema.
- Obtención e instalación: Una guía sencilla que explique el procedimiento básico para obtener e instalar el sistema. La guía debe estar dirigida a usuarios con nivel de experiencia preestablecido en la definición general del proyecto.
- Especificaciones de prueba y ejecución: Datos técnicos sobre la plataforma y/o entornos a utilizar en la prueba o ejecución del software en cuestión.

2.-Arquitectura del Sistema

Incluso un software de tamaño pequeño consta de la composición de varios módulos o partes interconectados de alguna forma. La descripción de la arquitectura del sistema informa sobre cuáles son estas partes, qué rol tienen dentro del software y la forma en que estas se organizan e interconectan.

La información sobre la arquitectura debería incluir como mínimo:

- Descripción jerárquica: Indica de qué forma se organizan jerárquicamente los componentes del sistema. Es decir, indicar si los mismos están organizados en paquetes, espacios de nombres o bien si el software posee una estructura monolítica.

- 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 deberá realizar una breve descripción, 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.

Nota: En esta sección se debe 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 estas.

3.-Diseño del modelo de base de datos.

Distinguir cuáles son las entidades involucradas en el sistema y mencionarlas(describirlas).

Puede ser un diseño orientado a objetos, relacional, etc., lo importante es tener una idea general del modelo de datos: entidades (tablas), atributos(campos) 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.

4.- 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 en 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 de la aplicación 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.

5.-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, 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 (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. (No aplica en Sistema de Mejores Prácticas).

• 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 que 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 y Glosario, si es necesario", 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.