

Estándar IEEE Std 1063-2001, para documentación de usuarios

"Contar con un estandar para la documentación de usuarios de software es necesario. El Std 1063-2001 ofrece una buena guía para crear un manual de usuario adecuado"



Muchos de nosotros nos encontramos en el área de la documentación de sistemas (al menos fue mi caso) por coincidencia. La primera vez que uno tiene un acercamiento con el área tiene una noción de los documentos que deben ser parte de un desarrollo o producto de software, pero sin embargo desconoce qué partes o secciones debe contener esos documentos. Por ejemplo ¿qué secciones debe tener un manual de usuario?, o ¿una memoria técnica?, etc.

El estándar IEEE std 1063-2001 brinda ese marco de referencia para establecer qué partes deben conformar cualquier documento que deba ser utilizado por un usuario del sistema o programa en cuestión.

Este estándar solo se aplica a la documentación de usuario, para la documentación de carácter técnico se utilizan otras recomendaciones de las cuales hablaremos en su momento. A grandes rasgos la recomendación del IEEE establece las siguientes partes para un documento que usarán los usuarios. Estas partes son:

- 1.- Identificación de los datos (paquete, título)
- 2.- Tabla de contenidos, en documentos con más de 8 páginas
- 3.- Lista de ilustraciones (optativo)
- 4.- Introducción
- 5.- Información para el uso de la documentación
- 6.- Conceptos de las operaciones
- 7.- Procedimientos
- 8.- Información sobre los comandos de software
- 9.- Mensajes de error y resolución de problemas
- 10.- Glosario

- 11.- Referencias
- 12.- Características de navegación
- 13.- Índice o Index
- 14.- Herramientas de búsqueda (en documentos electrónicos)

¿Qué es la documentación?

"Pensemos en nombres como los de Leonardo Da Vinci, Isaac Newton, Galileo o algún otro. Todos sabemos que han hecho pero, ¿Acaso podríamos acordarnos de ellos si no tuvieramos registro de sus ideas o de sus invenciones?."

En el caso de los grandes pensadores, mujeres y hombres, de la humanidad su legado a quedado plasmado en libros, esto es su obra y pensamiento ha sido documentado.

La documentación tiene su fundamento en el método científico. Con la intención de poder replicar el o los experimentos realizados para validar alguna tesis, el científico registra todas las condiciones bajo las cuales se realiza el experimento, de igual manera los resultados se escriben en tablas y estas a su vez generan gráficos estadísticos que nos permiten interpretarlos. En la actualidad los medios para llevar el registro de alguna actividad son muy diversos, siendo los digitales los más utilizados en la actualidad.

Ahora bien definiremos a la documentación como el proceso de registrar de manera ordenada los procedimientos y resultados obtenidos a lo largo de una investigación científica o proyecto o en general cualquier actividad que se necesite repetir más de una vez, por ejemplo: una receta de cocina bien documentada deberá llevar el tiempo exacto de cocción y la temperatura exacta en la que el platillo obtiene sus mejores propiedades.

Documentar nos permite dar el salto entre lo empírico y lo científico, entre el azar y las mejores prácticas.

Documentar un sistema de información implica almacenar y organizar la información necesaria (en forma de documento escrito o gráfico) con la intención de que al terminar el proyecto podamos: mantenerlo, mejorarlo y/o repetirlo. La cantidad de documentos generados a lo largo del proyecto va a depender del equipo que construye el software y del o los marcos de desarrollo que utilice la organización. Además en el caso de proyectos empresariales se deberá guardar documentación incluso de la configuración del hardware donde se despliegan los sistemas construidos.

De lo anterior me atrevería a dar la siguiente Clasificación de la Documentación:

1. Documentación administrativa:

Se refiere a los documentos que forman parte del contexto de la administración del proyecto como tal: Planes de trabajo, Definición del proyecto, Definición de roles y perfiles de los integrantes del proyecto. Análisis de costos, Análisis de riesgos, etc. En este lugar, por ejemplo, podríamos colocar todos los documentos de la gestión de proyectos que establece el PMBOOK.

2. Documentación técnica:

a) Desarrollo de software.

En este apartado pueden colocarse los documentos relacionados con el ciclo de desarrollo del sistema. De igual manera la cantidad de documentos varia de acuerdo al framework utilizado para crear el producto de software. En el caso de una aproximación clásica donde se define al ciclo de vida de los sistemas como: Análisis, Desarrollo, Pruebas e Implementación, tendríamos documentos tales como: Catálogo de requerimientos, digramas UML, Solución propuesta, Diccionario de Datos, Diccionario de formularios, Catálogo de objetos, Código fuente comentado, Plan de pruebas, pruebas y resultados, Plan de implementación o implantación, Memoria de instalación, Manual de usuario, Manual de administración.

Un marco más agil como SCRUM, no consideraria tantos documentos en su lugar consideraría más que suficiente que el código se comente de acuerdo al estandar del lenguaje, por ejemplo en Java se deberá utilizar el formato JavaDoc.

b) Aplicativos utilizados.

En el caso de aplicaciones empresariales donde además de desarrollar software a la medida se utilizan algunos programas que ya son comercializados, debemos guardar la memoria de instalación de los diversos aplicativos que forman la plataforma.

c) Componentes de hardware

Se debe guardar registro de la configuración del hardware sobre el cual se está instalando el aplicativo, debemos recordar que lo que deseamos es replicar el proceso más veces, por lo que la omisión de la configuración que tiene el hardware puede ser crucial para obtener los resultados esperados.

3. Procesos.

Tener escritos los procesos con base a estándares permitirá a las empresas evaluarlos y mejorarlos. Los procesos del cliente nos sirven para adecuar el producto de software a la empresa. La empresa que desarrolla el software o que ofrece servicios de IT puede encontrar un gran apoyo en marcos de referencia como ITIL.

Vamos a cerrar el tema considerando lo siguiente:

1 La documentación permite repetir el ciclo de desarrollo del software

2 Permite la mejora continua

3 Establece las bases del conocimiento del negocio

4 Marca el estandar de calidad del software

5 Permite medir el avance del proyecto

6 Establece un legado para la siguiente generación de desarrolladores

Finalmente considero que cualquier proyecto de software de mediana a gran escala, que no esté bien documentado está condenado al fracaso y al olvido. Por otro lado no existe una lista de documentos estandar, en todo caso la lista de documentos dependen de las metodologías y marcos de referencia utilizados en su desarrollo y de las indicaciones del personal directivo, esto es ¿qué quiero que se documente, de un producto en específico?.

¿Cómo escribir un manual de usuario?

"Conocer a detalle el proceso o sistema para el cual se requiere el manual, es la columna vertebral de toda buena documentación de usuario"

Recomendaciones para escribir un manual de usuario

El proceso de creación de un manual de usuario es cosa sencilla siempre y cuando se tenga en cuenta algunos puntos, los cuales deben seguirse en secuencia para no perdernos. El primer paso para escribir el manual sería: **Identificar al usuario de la documentación** o ¿quienes serán los lectores potenciales del documento?. Esto nos ayudará a ubicar cuál es el nivel técnico que se deberá utilizar en el documento y de alguna manera a planter su contenido.

El segundo paso será **definir su tabla de contenidos**. Necesitamos saber que

información será colocada en él y en qué orden. Una buena práctica es colocar la tabla de contenidos en una matriz, en una hoja de cálculo, para ir señalando el avance de cada tema o proceso y de esa manera conocer el avance de lo que vamos escribiendo y su relación.

Tercero **Organizar el contenido**. Una pregunta recurrente es ¿qué se debe colocar en un manual de usuario o cómo se debe organizar?. Bien yo utilizó dos aproximaciones: Colocar en el índice los casos de uso y clasificarlos de acuerdo a los usuarios que los utilizan o organizar la información por funcionalidad.

Por ejemplo, supongamos que se diseñó un programa de punto de venta y algunos de sus casos de uso son:

7 Dar de alta un producto

8 Modificar el precio del producto

9 Iniciar venta

10 Terminar venta

11 Revisar existencias

Pues bien puede colocarse una lista con los casos de uso y comenzar a describirlos o podemos organizar el índice de acuerdo a las funcionalidades o a los procesos que se pueden realizar en el sistema, por ejemplo:

1 Realizar una venta (este proceso seguro necesita incluir los casos de uso Iniciar venta y terminar venta)

2 Revisar inventario (en este proceso quiza se necesite cargar nuevos productos o modificar el precio de alguno y revisar las existencias en el almacen).

Cuarto **Recopilar y clasificar la información** para cada tema o subtema dentro del documento. Aunque este punto se coloca en el cuarto lugar es indispensable que contemos con un almacen de información antes de la elaboración del documento. Contar con un repositorio de información nos brinda la posibilidad de tener un almacen centralizado de información, para que buscarla y clasificarla sea más facil. (puedes ver como instalar SVN en tu organización en el siguiente [link](#)).



The world's largest professional association for the advancement of technology

About IEEE

Membership & Services

Societies & Communities

Publications & Standards

Conferences & Events

Search IEEE

Google™ Custom Search

Search

Una vez que hemos cumplido con los cuatro puntos anteriores debemos **elegir una estructura o marco de referencia** formal para nuestro manual. De manera personal utilizo como referencia el estandar IEEE std-1063-2001. En el se especifican las partes que debe contener cualquier documentación de usuario.

Ahora si debemos **comenzar a escribir el manual**, aunque antes debemos decidir en qué formato y con qué software será creado, mi recomendación es DocBook (puedes ver más información de su uso en el siguiente [tutorial de DocBook](#)). Debe considerarse que un manual esta diseñado para diferentes perfiles de usuario, así que aveces conviene realizar diferentes versiones para que el actor en turno pueda ejecutar todas las funciones que permita su perfil. Por lo tanto debe ser claro y descriptivo, es decir, no dejar nada a la imaginación, ni a las suposición del usuario, dejar claras las notas o prerequisites que el usuario requerirá para ejecutar el procedimiento sin problemas,

deberá contener imágenes claras y detalladas las cuáles deberán colocarse en un índice y contar con una descripción de la imagen. Un vez que el manual se ha terminado deberá ser evaluado y revisado por los programadores o los ingenieros de pruebas con la intención de validar que sus indicaciones son adecuadas y conducen a un resultado exitoso si se siguen al pie.

Es seguro que escribir un manual o cualquier documento técnico requieren, de quien lo escribe, habilidades, destrezas y conocimientos sólidos sobre la gramática, redacción y ortografía con la finalidad de que este bien hecho. Se debe recordar además que una buena documentación, en este caso un buen manual, puede hacer la diferencia entre el éxito o el fracaso del sistema en su implementación.

Documentación mínima de un proyecto de software

"La documentación de un proyecto es una parte esencial de las actividades que se deben realizar para el correcto desarrollo, implementación y mantenimiento del mismo."

¿Qué documentación debe tener el sistema?

Una de las primeras cuestiones que debe hacer un documentador es conocer el área, tareas que estarán bajo su responsabilidad y las particularidades del proyecto.

La siguiente definición de wikipedia nos presenta un vistazo a lo que el Documentador de sistemas se va a enfrentar:

"En sentido restringido, la **documentación** como ciencia documental se podría definir (a grandes rasgos) como la ciencia del procesamiento de la información, que proporciona información sobre algo con un fin determinado, de ámbito multidisciplinar o interdisciplinar.

Siguiendo a Fuentes y Pujol se puede señalar a la Documentación como una ciencia auxiliar e instrumental. También es una ciencia en si misma y una de las finalidades primordiales de la Documentación es informar.¹A falta de un consenso, hay diversos autores, como Juan Ros García o José López Yepes, que la consideran una ciencia (documental), a la vez que una disciplina, no sólo una técnica. También pueden considerarse, en sentido general, las ciencias de la documentación y la documentación como sinónimos, si el contexto no perturba la intención del emisor, es decir, si no se distorsiona el mensaje del interlocutor porque no se dé ambigüedad semántica."

Para ITIL V3 el conocimiento que poseen los colaboradores (conocimiento propio o propietario) y que han ido ganando a lo largo del tiempo, no sirve si no se adapta a estándares y se socializa. Siempre será recomendable documentar los procesos,

apegarse a estándares de la industria y evitar que el conocimiento del negocio se acumule en solo algunas personas.

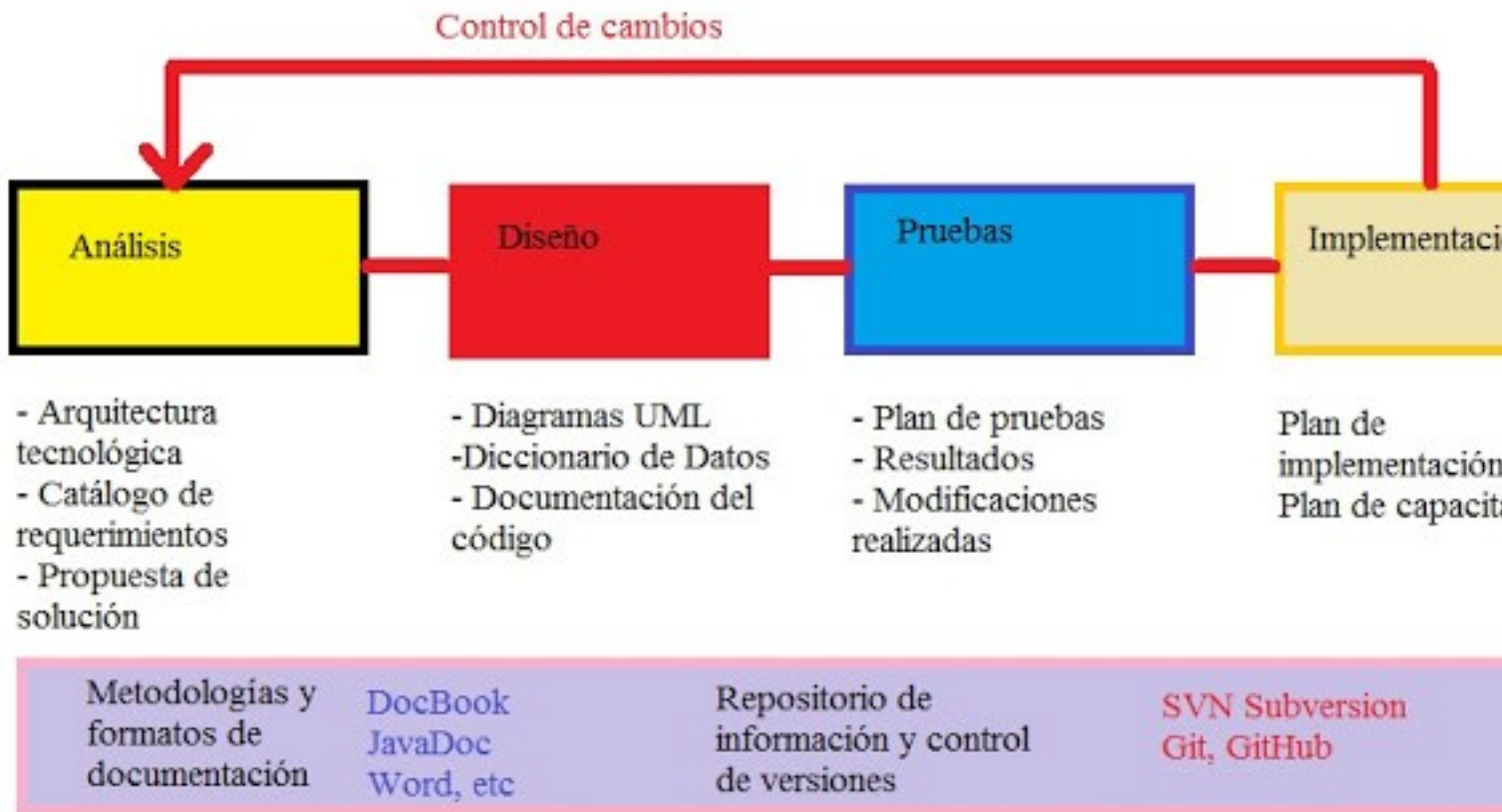
La documentación del proyecto de TI debe adecuarse a las necesidades establecidas y a la metodología o marco de referencia sobre el que se este desarrollando. Sin embargo existen documentos fundamentales de acuerdo a la etapa o ciclo de vida del sistema en que se encuentre.

El PMBOOK por ejemplo, establece la documentación administrativa y de control que debe asociarse al proyecto. ITIL nos da una idea de la documentación que debe existir para proporcionar los servicios de manera adecuada. Otros estándares específicos para la codificación como: JavaDoc o de documentación de usuario propuestos por el IEEE, deben ser establecidos por el documentador y seguidos por los programadores.

Las metodologías de desarrollo de software brindan una guía de la documentación que es necesaria tener, aunque de manera general siempre se requerirá crear: el catálogo de requerimientos, diccionario de datos, diagramas UML, el plan de trabajo, el plan de implementación o implantación, el plan de capacitación y el control de cambios.

Dependiendo del lenguaje en el cual se esta desarrollando el proyecto se debe establecer el patrón de documentación por ejemplo: JavaDoc, JSDoc, JSON, PHP, etc.

El siguiente diagrama muestra las diferentes etapas y áreas de un proyecto de TI y los documentos asociados a cada una de ellas:



Finalmente todos los archivos incluyendo la documentación de usuario , administrativa, presupuestal y procedimientos, deben colocarse en un repositorio bajo el cual se tenga un control de versiones. De manera personal recomiendo el uso de SVN Subversion o Git, aunque existen muchos más.

Una vez que se ha establecido el ambiente para el almacenamiento y control de la documentación y después de determinar cuales serán los documentos que han de realizarse a lo largo del proyecto, debe determinarse el formato en que serán capturados y distribuidos. Estos formatos pueden ser: formatos en Word u algún otro procesador de texto, LaTeX, DocBook, etc.