PREPARACIÓN DE LA INSTALACIÓN

**Aquí esta mi cambio ahora si**

Instalación del sistema

El Manual de Instalación tiene como objetivo servir de guía en la instalación del sistema. Para ello, en primer lugar, deberá especificar los requerimientos hardware y software necesarios para el correcto funcionamiento del sistema, para posteriormente describir cada uno de los pasos necesarios para la configuración, compilación e instalación del sistema. Además, se deberán incluir las pruebas que se deberán realizar para asegurar que la instalación se ha realizado correctamente, así como el procedimiento de marcha atrás a aplicar en caso de que no haya resultado exitosa la instalación del sistema.

El [software](https://basicoyfacil.wordpress.com/2008/10/06/hardware-y-software/) que vayamos a utilizar en nuestro ordenador necesita de unos requisitos, es decir, unas condiciones para poder funcionar en nuestro equipo.

Los requisitos mínimos son una serie de componentes o valores de estos, para los que está diseñado el software en cuestión y que son, los que debemos disponer en nuestro equipo para un funcionamiento adecuado.

Algunos de estos requisitos pueden ser:

Velocidad y tipo de procesador.

Cantidad de memoria Ram.

Tipo de tarjeta gráfica.Espacio libre en disco duro.

Definición de escenarios

Un escenario es una descripción parcial y concreta del comportamiento de un sistema en una determinada situación. Es una descripción parcial, porque no necesita describir todas las características de las entidades involucradas, sólo se describe aquello que está relacionado con un comportamiento particular del sistema analizado. A pesar de estar acotados a un determinado comportamiento, describen todo el contexto que involucra a esa actividad: recursos del sistema, objetivos de los usuarios, contexto social en que se desarrolla, entidades involucradas. Proveen un “retrato” de como esa actividad se lleva a cabo. Los escenarios describen situaciones teniendo en cuenta aspectos de uso, permitiendo: conocer el problema, unificar criterios, ganar compromiso con clientes / usuarios, organizar los detalles involucrados y entrenar a nuevos participantes.

Los escenarios logran diferentes objetivos dependiendo de la fase donde se usen durante el proceso de desarrollo de software. En la fase de Ingeniería de Requisitos, los objetivos del escenario son:

Capturar los requisitos.

Proveer un medio de comunicación entre los stakeholders.

Proveer un soporte para trazabilidad.

Los escenarios pueden ser escritos de diferentes maneras, pero deberían incluir al menos, la siguiente información:

Una descripción del estado del sistema antes de entrar al escenario.

El flujo normal de eventos en el escenario.

Excepciones al flujo normal de eventos.

Información acerca de otras actividades que podrían estar sucediendo al mismo tiempo) - Una descripción del estado del sistema después de completar el escenario.

Verificación del procedimiento

El procedimiento Verificación de Entrega Software engloba todos los servicios que son ejecutados en un entorno de pruebas. En el transcurso de este procedimiento se realizarán tanto pruebas técnicas, relacionadas con la existencia de una normativa, como pruebas funcionales, orientadas a asegurar que la aplicación está libre de errores relacionados con la percepción de calidad del usuario final.

DISEÑO Y DESARROLLO

Diseño

El diseño de un software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad. Los principios de diseño establecen una filosofía general que guía el trabajo de diseño que debe ejecutarse. Deben entenderse los conceptos de diseño antes de aplicar la mecánica de éste, y la práctica del diseño en sí lleva a la creación de distintas representaciones del software que sirve como guía para la actividad de construcción que siga

El diseño en el nivel de componentes tiene lugar una vez terminado el diseño de la arquitectura. En esta etapa se ha establecido la estructura general de los datos y del programa del software. El objetivo es traducir el modelo del diseño a software operativo. Pero el nivel de abstracción del modelo de diseño existente es relativamente alto y el del programa operativo es bajo. La traducción es difícil y está abierta a la introducción de errores sutiles que son difíciles de detectar y de corregir en las etapas posteriores del proceso del software

Durante el diseño arquitectónico, se define un conjunto completo de componentes de software. Pero las estructuras internas de datos y detalles de procesamiento de cada componente no están representadas en un nivel de abstracción cercano al código. El diseño en el nivel de componentes define las estructuras de datos, algoritmos, características de la interfaz y mecanismos de comunicación asignados a cada componente del software.

Antes de elaborar el software, se tiene que ser capaz de determinar si funcionará. El diseño en el nivel de componentes lo representa en forma tal que permite revisar los detalles del diseño para garantizar su corrección y su consistencia con otras representaciones del diseño (por ejemplo, los datos y el diseño de la arquitectura y la interfaz). Esto proporciona un medio para evaluar si funcionarán las estructuras de datos, interfaces y algoritmos

Las representaciones de diseño de datos, arquitectura e interfaces constituyen el fundamento para el diseño en el nivel de componentes. La definición de clase o narrativa de procesamiento de cada componente se traduce a un diseño detallado que utiliza formas de diagrama o basadas en texto que especifican las estructuras de datos internas, los detalles de la interfaz local y la lógica del procesamiento. La notación del diseño incluye diagramas UML y formatos complementarios. Se especifica el diseño del procedimiento con el empleo de construcciones de programación estructurada. Con frecuencia es posible obtener componentes de software reutilizable ya existentes, en lugar de construir nuevos.

Desarrollo

Desarrollar un [software](https://www.ecured.cu/Software) significa construirlo simplemente mediante su descripción. Esta es una muy buena razón para considerar la actividad de desarrollo de software como una [ingeniería](https://www.ecured.cu/Ingenier%C3%ADa_de_Software). En un nivel más general, la relación existente entre un software y su entorno es clara ya que el software es introducido en el mundo de modo de provocar ciertos efectos en el mismo.

Aquellas partes del mundo que afectarán al software y que serán afectadas por él será el [Dominio de Aplicación](https://www.ecured.cu/index.php?title=Dominio_de_Aplicaci%C3%B3n&action=edit&redlink=1). Es allí donde los [usuarios](https://www.ecured.cu/Usuarios) o [clientes](https://www.ecured.cu/Clientes) observarán si el desarrollo del software ha cumplido su propósito.

Una de las mayores deficiencias en la práctica de construcción de software es la poca atención que se presta a la discusión del problema. En general los desarrolladores se centran en la solución dejando el problema inexplorado. El problema a resolver debe ser deducido a partir de su solución.

Esta aproximación orientada a la solución puede funcionar en campos donde todos los problemas son bien conocidos, clasificados e investigados, donde la innovación se ve en la detección de nuevas soluciones a viejos problemas.

Pero el desarrollo de software no es un campo con tales características. La versatilidad de las [computadoras](https://www.ecured.cu/Computadora) y su rápida evolución hace que exista un repertorio de problemas en constante cambio y cuya solución software sea de enorme importancia.

IMPLATACION

La implantación es parte de la implementación, pero no son lo mismo.

La implantación es un proceso especial de inserción del sistema en la institución, el cual debe ocurrir en el primer periodo por parte de sus potenciales usuarios. No obstante que la herramienta adquirida o desarrollada es robusta y probada, no es suficiente para su inserción en la organización, pues requiere ser utilizada.

La implantación es un proceso, que dice relación con la gestión del cambio y es crucial para que los usuarios comiencen a apropiarse del sistema. Es decir, que los clientes conviertan la herramienta en una solución de su quehacer, explotando apropiadamente los atributos, multiplicando la socialización del uso y el conocimiento sobre la aplicación.

La implantación debe trabajarse con un plan muy bien diseñado, de modo que pueda integrarse sin dificultades con los otros sistemas de la organización y comunicar bien. A este efecto, es necesario diseñar medios de comunicación eficaces entre los actores involucrados, controlar y registrar evidencias del proceso. Identificar la metodología de coordinación que contemple las mejores prácticas y normas.

Validación del sistema

Las pruebas de validación en la [ingeniería de software](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software) son el proceso de revisión que verifica que el sistema de [software](https://es.wikipedia.org/wiki/Software) producido que cumple con las especificaciones y que logra su cometido. Es normalmente una parte del proceso de [pruebas de software](https://es.wikipedia.org/wiki/Pruebas_de_software) de un proyecto, que también utiliza técnicas tales como evaluaciones, inspecciones y [tutoriales](https://es.wikipedia.org/wiki/Cursillo). La validación es el proceso de comprobar que lo que se ha especificado es lo que el [usuario](https://es.wikipedia.org/wiki/Usuario) realmente quería.

Entrenamiento a usuarios finales

Un plan de capacitación se realiza mediante unos pasos los cuales guiaran e informaran al usuario final, un factor de gran importancia es que no se debe omitir es la capacitación, como un hecho que se da una sola vez para cumplir con un requisito. La mejor forma de capacitación es la que se obtiene de un proceso continuo, siempre buscando conocimientos y habilidades para estar al día.

Este proceso tiene unos pasos a seguir:

1. Analizar las necesidades. Identifica habilidades y necesidades de los conocimientos y desempeño.

2. Diseñar la forma de enseñanza: Aquí se elabora el contenido del programa folletos, libros, actividades. Etc.

3. Validación: Aquí se eliminan los defectos del programa y solo se presenta a unos cuantos pero que sean representativos.

4. Aplicación: Aquí se aplica el programa de capacitación.

5. Evaluación: Se determina el éxito o fracaso del programa.