



PLAN DE TRABAJO VIRTUAL

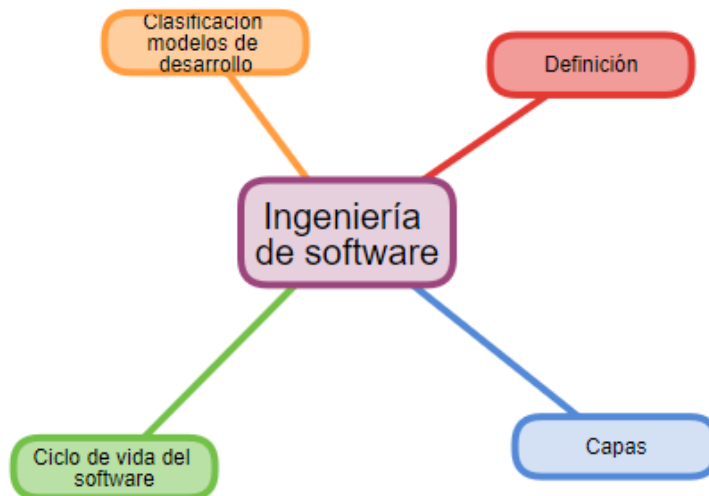
Área/Asignatura MEDIA TÉCNICA	Grupo: 10°	Fecha de entrega a estudiantes: 30 de abril	Fecha de devolución: 5 de mayo
---	----------------------	---	--

MÓDULO: Identificación del ciclo de vida del software

TEMA: Ingeniería del software

ACTIVIDADES:

1. Lea cuidadosamente el documento que se encuentra anexo
2. Realice un **RESUMEN CORTO**, de la lectura realizada en el cuaderno de Ciclo de vida del software. Recuerde que un resumen no implica copiar todo el texto, es un escrito con las ideas principales que den cuenta de la lectura.
3. En una hoja de block realiza un mapa mental con el resumen con los principales conceptos de Ingeniería del software. A continuación, se muestra la estructura principal del mapa, su tarea es completarlo de tal forma que dé cuenta del texto y siguiendo las características trabajadas en clase acerca mapas mentales.



PD: El mapa debe ir debidamente marcado con el nombre de cada estudiante, mapa que este sin marcar no será calificado. Esta hoja debe ir pegada en el cuaderno.

TENGA EN CUENTA:

- Resuma la información encontrada
- Por favor cuide su caligrafía, los textos deben ser muy legibles y entendibles, fácil para la lectura.
- Revise la ortografía y redacción de los textos
- Todos los mapas mentales deben estar bien marcados con el nombre de cada estudiante, el que no cumpla con este requisito **NO SERÁ EVALUADO**.
- Como el mapa mental evidencia el resumen de una información y cada uno tiene su propia forma de resumir información, **NO PUEDE HABER MAPAS IGUALES**.

MEDIO DE ENTREGA

- Escanear o enviar una imagen del cuadro legible a edmodo.
- Solo los estudiantes que no tengan acceso a edmodo podrán enviar su actividad al correo electrónico leidygiraldog@gmail.com o al Whatassap 3225134156, con previa autorización (Por favor me escribe quien tenga esta dificultad).

FECHA DE ENTREGA

Recibo trabajos hasta el 5 de mayo a las 12:00 de la noche.

INGENIERÍA DE SOFTWARE

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (software). Esta disciplina trasciende la actividad de programación, que es la actividad principal a la hora de crear un software. El ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto.

La ingeniería de software, por lo tanto, incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema.

CAPAS DE LA INGENIERÍA DE SOFTWARE



• Calidad

El control de calidad implica vigilar los procesos de desarrollo del software para asegurarse de que se siguen los procedimientos y los estándares de garantía de calidad. Existen algunos enfoques complementarios que se utilizan para comprobar la calidad de las entregas de un proyecto:

• Proceso

Un proceso de software es un conjunto de actividades y resultados asociados que producen un producto de software. Estas actividades se llevan a cabo por los ingenieros de software. Existen cuatro actividades de procesos las cuales son fundamentales y son comunes para todos los procesos de software. Estas actividades son:

- **Especificación del software:** Es donde los clientes e ingenieros definen el software a producir y las condiciones sobre su operador.
- **Desarrollo del software:** Donde el software es diseñado y programado.
- **Validación del software:** El software es validado para confirmar que es lo que el cliente requiere.
- **Evolución del software:** el software se modifica para adaptarlo a los cambios que se requieran por el cliente y el mercado.

• Método

Los métodos de la ingeniería de software indican como construir técnicamente el software. Abarcan un amplio espectro de tareas en las que se incluyen la planificación y estimación de proyectos, análisis de los requisitos del sistema y del software, diseño de estructuras de datos, arquitectura de programas y procedimientos algorítmicos, codificación, prueba, mantenimiento, documentación. En estos métodos se pueden incluir técnicas orientadas a objetos, estructuradas, de descomposición funcional. Aquí, el profesional debe escoger entre las técnicas que más le convienen para desarrollar su trabajo adecuadamente.

• Herramienta

Las herramientas de ingeniería de software proporcionan un soporte automático para los métodos. En la actualidad existen herramientas para soportar métodos como los que se han mencionado anteriormente.

A estas herramientas se les llama herramientas asistidas por computadora para la ingeniería de software CASE (Computer Assisted Software Engineering).

Cabe destacar que el proceso de desarrollo de software implica lo que se conoce como ciclo de vida del software:

CICLO DE VIDA DEL SOFTWARE

Es el proceso que se sigue para construir, entregar y hacer evolucionar el software, desde la concepción de una idea hasta la entrega y retiro del sistema; es decir, especifica el enfoque general del desarrollo, indicando los procesos, actividades y tareas que se van a realizar y en qué orden, y los productos que se van a generar, los que se van a entregar al cliente y en qué orden se van a entregar.

Etapas del desarrollo o ciclo de vida del software

Etapas de análisis: Es el proceso de investigar un problema que se quiere resolver. Definir claramente el Problema que se desea resolver o el sistema que se desea crear. Identificar los componentes principales que integrarán el producto.

- **Etapas de Diseño:** Es el proceso de utilizar la información recolectada en la etapa de análisis al diseño del producto. La principal tarea de la etapa de diseño es desarrollar un modelo o las especificaciones para el producto o Componentes del Sistema.
- **Etapas de Desarrollo:** Consiste en utilizar los modelos creados durante la etapa de diseño para crear los componentes del sistema.
- **Etapas de Pruebas o Verificación Prueba:** Consiste en asegurar que los componentes individuales que integran al sistema o producto, cumplen con los requerimientos de la especificación creada durante la etapa de diseño.
- **Etapas de Implementación o Entrega Implantación:** Consiste en poner a disposición del cliente el producto.
- **Etapas de Mantenimiento:** Consiste en corregir problemas del producto y re- liberar el producto como una nueva versión o revisión (producto mejorado).
- **Etapas final EOL (End-of-Life)** El fin del ciclo del producto consiste en realizar todas las tareas necesarias para asegurar que los clientes y los empleados están conscientes de que el producto ya no será vendido ni soportado.

En cada una de las etapas de un modelo de ciclo de vida, se pueden establecer una serie de objetivos, tareas y actividades que lo caracterizan.

Existen diferentes modelos de ciclo de vida y la elección de un modelo para un determinado tipo de proyecto es realmente importante; el orden de las etapas es uno de estos puntos importantes y de eso dependerá en gran medida el éxito o fracaso del proyecto de software.

CLASIFICACIÓN DE LA TECNOLOGÍA, PARADIGMAS O MODELOS DE DESARROLLO DE SOFTWARE

Los modelos de desarrollo de software son una representación abstracta de una manera en particular. Realmente no representa cómo se debe desarrollar el software, sino de un enfoque común. Puede ser modificado y adaptado de acuerdo a las necesidades del software en proceso de desarrollo. Hay varios modelos para perfilar el proceso de desarrollo, cada uno de las cuales cuenta con pros y contras. El proyecto debería escoger el más apropiado para sus necesidades. En ocasiones puede que una combinación de varios modelos sea apropiado. Existen tres paradigmas de los modelos de desarrollo de software:

1. Desarrollo estructurado o Tradicional:

Es uno de los paradigmas más antiguo, se inventó durante la creación del método estructurado. Si se elige un proyecto, el método varía en etapas. Como todo modelo, existen sus pros y contras al usar este paradigma:

CUADRO SOBRE LAS VENTAJAS Y DESVENTAJAS DEL USO DEL PARADIGMA TRADICIONAL	
VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none">• Mayor control en cuanto a la programación del desarrollo.• Al tener control, se reduce el riesgo de excesos de gastos.	<ul style="list-style-type: none">• El usuario no participa en el proceso de desarrollo.• El proceso no se hace de forma secuencial.• El tiempo de desarrollo excede al estimado.• Si el usuario olvida aclarar pautas, esto puede significar, sobrecostos en el proyecto.



Si se aplica este paradigma, unos de los principales problemas, es que las etapas realizadas no son autónomas de las siguientes, creando una dependencia estructural y en el caso de un error atrasaría todo el proyecto. Se tiene que tener pautas bien definidas, y que no se incurra a modificación porque implicaría en que el software no cumpla con su ciclo de vida. Tener en cuenta que el cliente no se vea afectado por la impaciencia.

2. Desarrollo Orientado a Objetos:

Estos modelos se basan en la Programación orientada a objetos; por lo tanto, se refiere al concepto de clase, el análisis de requisitos y el diseño. El modelo o paradigma orientado a objetos posee dos características principales, las cuales son:

- Permite la re-utilización de software.
- Facilita el desarrollo de herramientas informáticas de apoyo al desarrollo, el cual es simple al implementarla en una notación orientado a objetos llamado UML.

3. Desarrollo Ágil:

Es un paradigma de las Metodologías De Desarrollo basado en procesos ágiles. Estos intentan evitar los tediosos caminos de las metodologías tradicionales enfocándose en las personas y los resultados. Usa un enfoque basado en el Valor para construir software, colaborando con el cliente e incorporando los cambios continuamente.⁴

Información tomada de:

<http://www.lsi.us.es/docencia/get.php?id=7550>

<https://sites.google.com/site/ingenierialeosw/unidad-1-fundamentos-de-ingenieria-de-software/1-1-conceptos-basicos>

https://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_software

<http://cotana.informatica.edu.bo/downloads/ld->

[Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF](#)