



**TECNOLÓGICO
NACIONAL DE MÉXICO**



MATERIA:

INGENIERIA DE SOFTWARE

ALUMNO:

EDITH MARTINEZ MARTINEZ

DOCENTE:

M.S.C. EDUARDO FLORES GALLEGOS

CARRERA:

INGENIERIA EN TECNOLOGIAS DE LA INFORMACION Y LA CUMUNICACION

DEFINICIONES

DIAGRAMA GANTT:

El diagrama de Gantt es una herramienta gráfica que ilustra un cronograma del proyecto, llamado así por su inventor, Henry Gantt, quien diseñó este modelo a principios del siglo pasado. Los gráficos modernos de Gantt también muestran las relaciones de dependencia entre las actividades y el estado actual de la programación. Un diagrama de Gantt enumera las tareas que se realizarán en el eje vertical y los intervalos de tiempo en el eje horizontal. El ancho de las barras horizontales muestra la duración de cada actividad.

Los beneficios de usar Gantt son:

- Claridad
- Comunicación
- Motivación
- Coordinación
- Creatividad
- Tiempo de gestión
- Flexibilidad
- Capacidad de gestión
- Eficacia
- Rendición de cuentas

PERT:

este diagrama permite planificar el proyecto, identificar el camino crítico del proyecto, las tareas críticas por no tener holgura, y calcular su plazo mínimo de ejecución.

Aunque pueden existir diferentes formas de representar el diagrama PERT, aquí se muestra una donde las tareas se presentan mediante una casilla dividida en 4 áreas:

Nombre	
Duración	
Primera fecha de inicio	Última fecha de inicio

Nombre de la tarea

Duración de la tarea. Para ello puedes usar diferentes técnicas de estimación de plazos.

Primera fecha de inicio. Es la fecha más cercana al inicio en que la tarea puede empezar a ejecutarse, considerando las relaciones con las tareas precedentes.

Última fecha de inicio. Es la fecha más tarde en que la tarea puede empezar sin comprometer la duración mínima del proyecto

CPM:

Una herramienta que me permita estimar el tiempo más corto en el que es posible completar un proyecto es el método de la ruta crítica (CPM-Critical Path Method) o del camino crítico. Este es un algoritmo utilizado para el cálculo de tiempos y plazos en la planificación de proyectos. El objetivo principal es determinar la duración de un proyecto, donde cada una de las actividades del mismo tiene una duración estimada. La duración de las actividades que forman la ruta crítica determina la duración del proyecto entero y las diferencias con las otras rutas que no sean la crítica se denominan tiempos de holgura. Un proyecto puede tener más de una ruta crítica

COCOMO:

El Modelo Constructivo de Costes (Constructive Cost Model) fue desarrollado por B. W. Boehm a finales de los 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981). COCOMO es una jerarquía de modelos de estimación de costes software que incluye submodelos básico, intermedio y detallado.

Las ecuaciones de estimación del esfuerzo de desarrollo tienen la forma

$$E = a_1 S^{b_1} m(X)$$

con

- S el número de miles de líneas de código fuente
- $m(X)$ es un multiplicador que depende de 15 atributos
- en la siguiente tabla se muestran los coeficientes para los diferentes modos

Modo	Básico		Intermedio	
	a_i	b_i	a_i	b_i
Orgánico	2.4	1.05	3.2	1.05
Semiencajado	3.0	1.12	3.0	1.12
Empotrado	3.6	1.2	2.8	1.2

JUICIO EXPERTO (EN DESARROLLO DE SOFTWARE):

El juicio experto se basa en la experiencia, en el conocimiento anterior y en el sentido comercial de uno o más individuos dentro de la organización.

Desarrollar visiones a largo plazo.

Evaluar la validez o deficiencias de los elementos o etapas de un proyecto.

MODELOS ALGORITMICOS:

Para realizar un programa es conveniente el diseño o definición previa del algoritmo. El diseño de algoritmos requiere creatividad y conocimientos profundos de la técnica de programación.

Los algoritmos son independientes de los lenguajes de programación. En cada problema el algoritmo puede escribirse y luego ejecutarse en un lenguaje diferente de programación. El algoritmo es la infraestructura de cualquier solución, escrita luego en cualquier lenguaje de programación

KSLOC:

La definición de línea de código fuente es esencialmente ambigua para la mayor parte del software. Su significado varía de un lenguaje de programación a otro, pero también dentro de un mismo lenguaje de programación.

Una línea de código fuente es cada una de las líneas de un archivo de código fuente de un programa informático. Habitualmente en cada línea se ejecuta una instrucción que tiene que ejecutar el programa. También es habitual tabular las estructuras de control del programa en cuestión para una lectura más fácil. Viene a ser como la oración en libros y textos escritos en general.

FACTIBILIDAD TÉCNICA:

La factibilidad técnica "Indica si se dispone de los conocimientos y habilidades en el manejo métodos, procedimientos y funciones requeridas para el desarrollo e implantación del proyecto.

FACTIBILIDAD LEGAL:

Se refiere a que el desarrollo del proyecto o sistema no debe infringir alguna norma o ley establecida a nivel local, municipal, estatal, federal o Mundial.

FACTIBILIDAD OPERACIONAL:

La Factibilidad Operacional es parte de la formulación de un proyecto sobre la base de su estructuración metodológica, la identificación clara de relaciones causales entre problemas y soluciones, el diseño de estrategias de ejecución, las competencias profesionales de las personas participantes y la capacidad legal y administrativa de las organizaciones o áreas de una empresa, proponentes para desarrollar las actividades previstas en el proyecto.

FACTIBILIDAD ECONÓMICA:

Se refiere a que se dispone del capital en efectivo o de los créditos de financiamiento necesario para invertir en el desarrollo del proyecto, el cual deberá haber probado que sus beneficios a obtener son superiores a sus costos en que incurrirá al desarrollar e implementar el proyecto o sistema

ARQUITECTURAS DE SOFTWARE:

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de

la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software.

TCP/IP:

Este modelo es un protocolo para comunicación en redes que permite que un equipo pueda comunicarse dentro de una red. Está basado en el modelo teórico OSI de capas con la que comparte 4 de ellas; sin embargo, ofrece muchas más opciones y es un modelo práctico.

La comprensión de las características principales de la pila de protocolos de Internet TCP/IP posibilita la configuración de redes básicas por lo que un curso de TCP/IP será fundamental en la formación sobre redes e internet (se llama pila de protocolos a una colección ordenada de protocolos organizados por capas).

<https://rockcontent.com/es/blog/que-es-diagrama-de-gantt/>

<https://www.rekursosenprojectmanagement.com/creacion-del-cronograma-pert/>

<https://www.eoi.es/blogs/madeon/2013/04/14/metodo-de-ruta-critica-cpm-critical-path-method>

<https://www.phpfreelance.es/sloc.html>

https://www.uma.edu.ve/moodle_uma/course/info.php?id=28

<https://openwebinars.net/blog/que-es-tcpip/>