



TECNOLÓGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO[®]
de Pabellón de Arteaga

TEC

28-4-2020

Ingeniería de software

DEFINICIONES UNIDAD 3

Profesor:
Eduardo Flores Gallegos

Alumno:
José Yovani Loera Lara

Carrera:
**Ingeniería en Tecnologías de la Información y
Comunicaciones**



Diagrama de Gantt

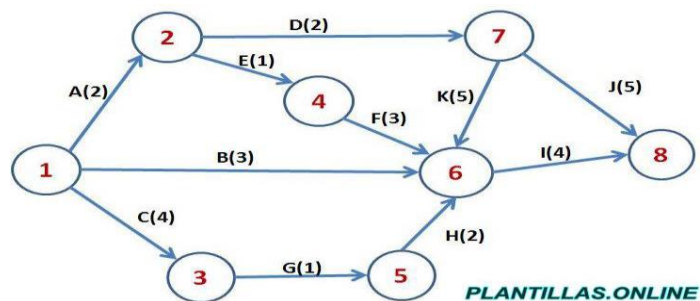
El diagrama de Gantt es una herramienta grafica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. A pesar de esto dicho diagrama no indica las relaciones entre dichas actividades.

Actividad	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8
Contratar a la Empresa DMFS para que realice el diseño de Mini fosas y Canales en las plantaciones de rubro de ciclo corto	x							
Solicitar y evaluar cotizaciones y presupuestos en las diferentes compañías de construcción agroindustrial de la zona		x	x					
Toma de decisiones y disposición de presupuesto para la construcción señalada			x					
Selección y contratación formal de la empresa encargada de la construcción				x				
Construcción de las Minifosas y Canales dentro de la plantación de rubros de ciclo corto					x	x	x	
Supervisar la Obra a construir					x	x	x	x
Realizar una prueba de drenaje para comprobar que la obra se realizó correctamente								x

Pert

El método pert o como se puede decir (Project Evaluation and Review Techniques) es un algoritmo basado en la teoría de redes que está diseñado para facilitar la planificación de proyectos

El algoritmo PERT se desarrolla mediante intervalos probabilísticos, considerando tiempos optimistas, probables y también pesimistas, lo cual lo hace diferente de método CMP que este supone tiempos determinísticos.





CPM

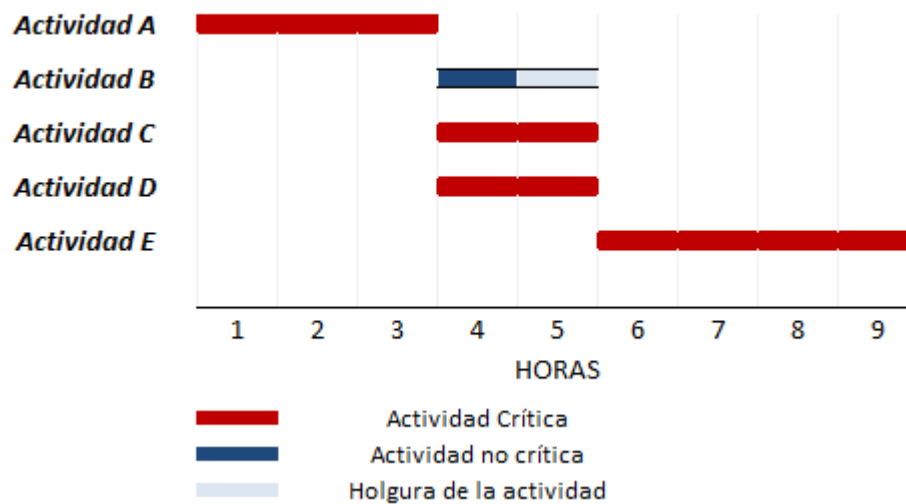
El método de la ruta crítica CPM (Critical Path Method), es un algoritmo basado en la teoría de redes diseñado para facilitar la planificación de proyectos. El resultado final del CPM será un cronograma para el proyecto, en el cual se podrá conocer la duración total del mismo, y la clasificación de las actividades según su criticidad. El algoritmo CPM se desarrolla mediante intervalos determinísticos, lo cual lo diferencia del método PERT que supone tiempos probabilísticos.

Paso 1: Actividades del proyecto

Paso 2: Diagrama de la red

Paso 3: Calcular la red

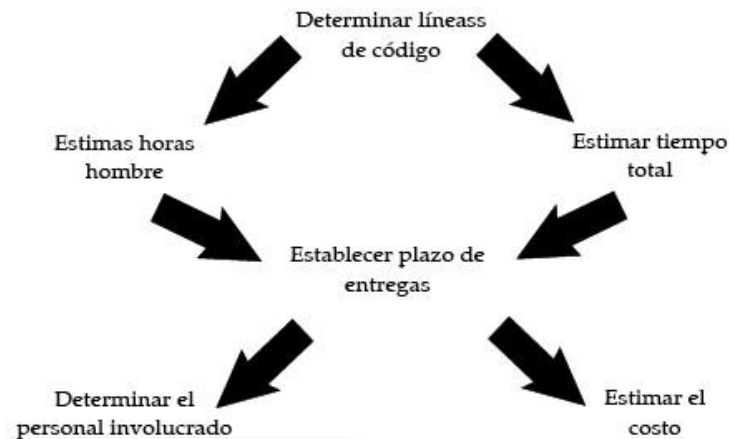
Paso 4: Establecer el cronograma





COCOMO

El Modelo Constructivo de Costes COCOMO -Constructive Cost Model- es utilizado en proyectos de software para estimar los costes del mismo en función de tres submodelos: básico, intermedio y detallado.



El modelo básico

Estima el coste del proyecto pequeño o mediano en función de número de líneas de código estimadas. En este modelo, el algoritmo COCOMO establece varios criterios de desarrollo, dependiendo el nivel de dificultad no del nivel de experiencia de los desarrolladores que se supone- sino de posibles dificultades que se pueden encontrar en el desarrollo o limitaciones del hardware usado en el desarrollo del software.

El modelo intermedio

Se utiliza para estimaciones más complejas. Éste incluye 15 atributos –dentro de 4 categorías- del software para determinar el coste del proyecto.

Atributos del producto: garantía de funcionamiento requerida para creación del software, tamaño de la BBDD, etc.

Atributos del ordenador usado: capacidad de almacenamiento, rapidez del ordenador, etc.

Atributos del personal: experiencia en el tipo de software a desarrollar, en el lenguaje usado, etc.

Atributos del proyecto: software usado para el desarrollo, lenguaje necesario para crear el software, etc.

Todos estos atributos son ponderado matemáticamente en atendiendo de su relevancia. De esta manera se intenta aproximar el coste estimado al real, lo máximo posible.



El modelo detallado

Incorpora las características del modelo intermedio y lleva a cabo una evaluación del impacto de los motivantes del coste en cada caso -análisis, diseño, etc.- del proceso de ingeniería del software.

Juicio Experto (En desarrollo de software)

Esta estrategia de estimación se lleva a cabo por una persona reconocida como un experto en la tarea, y una parte significativa del proceso de estimación se basa en un proceso de razonamiento no explícito y no recuperable, es decir, "la intuición". Según un estudio realizado por el método de estimación dominante es Juicio de Expertos.

Los autores sostienen que no hay suficiente evidencia disponible que sugiera que las estimaciones mejoran con el uso de métodos formales de estimación. Es el método de estimación dominante, más ampliamente usado no solo en el área de desarrollo de software sino también en otras áreas tales como negocios, salud, educación, etc.

Modelos Algorítmicos

Un algoritmo es un conjunto prescrito de instrucciones o reglas que permite llevar a cabo una actividad mediante pasos sucesivos. Este conjunto ordenado de operaciones sistemáticas permite hacer un cálculo y hallar la solución de un tipo de problemas.

Sin embargo, en Inteligencia Artificial es complicado definir algoritmos que hagan lo que queramos: ¿qué pasos hay que seguir para detectar un gato en una foto? ¿o para decidir si un cliente se va a cambiar de compañía telefónica? ¿o para estimar el volumen de negocio que tendrá una compañía en el próximo mes?

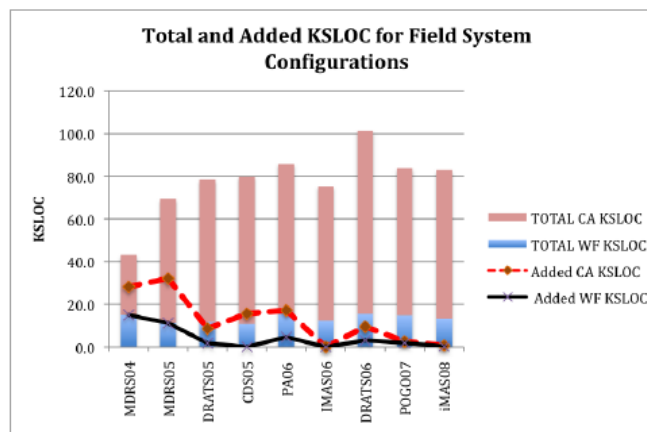
Estos problemas no se pueden resolver con instrucciones concretas dadas a un ordenador. En su lugar, se utilizan modelos matemáticos complejos que aprenden a base de ejemplos, es decir, a través de los datos. Este proceso de aprendizaje se denomina entrenamiento del modelo.



KSLOC

Líneas de código fuente (SLOC), también conocidas como líneas de código (LOC), es una métrica del software utilizado para medir el tamaño de un programa de ordenador , contando el número de líneas en el texto del programa del código fuente. SLOC se suele utilizar para predecir la cantidad de esfuerzo que se requiere para desarrollar un programa, así como para estimar la productividad de programación o de mantenimiento una vez que se produce el software.

Ejemplo:





Factibilidad Legal

Se recomienda adquirir las licencias para el software a emplearse en la implantación de un sistema informático de manera auténtica, con la finalidad de no tener inconvenientes legales a futuro.

La factibilidad legal nos permite determina los derechos que tienen los autores sobre la documentación realizada por estos en este proyecto, la cual es exclusividad de los desarrolladores de este sistema, por tal motivo queda prohibida la distribución y reproducción de este documento, tales como la publicación impresa o su grabación.

Factibilidad Operacional

Se ha visto mediante la matriz de estudio de problemas/oportunidades que las dificultades que se van a remediar mediante el sistema son de gran prioridad y urgencia, debido a que los procesos son realizados diariamente y de manera no equitativa.

Por este motivo, la creación de un sistema que automatice dicho problema será una de las mejores soluciones, además se ha obtenido la información necesaria mediante la investigación realizada, con lo que se creara un sistema que llene todas las expectativas y necesidades.

En cuanto a información, control y su eficacia se tiene como objetivo trabajar mutuamente con los socios, secretaria y el gerente, para que ellos vayan describiendo cuáles serán los requerimientos en el sistema y de esta manera desarrollar un sistema en donde se tendrá la completa satisfacción de los que lo van a utilizar.

Además, gracias a su plan de trabajo tiene gran aceptación entre los que lo van a utilizar por parte de la directiva y el resto de socios que piensan que el sistema es ideal para sus metas y que el problema será superado.



Factibilidad Económica

En el estudio de la Factibilidad Económica, determinamos el presupuesto de costos de los recursos técnicos, humanos y materiales tanto para el desarrollo como para la implantación del Sistema.

Además, nos ayudara a realizar el análisis costo-beneficio de nuestro sistema, el mismo que nos permitirá determinar si es factible a desarrollar económicamente el proyecto.

A continuación se describe los costos del recurso necesario para el desarrollo de nuestro Sistema de Información:

Recursos Humanos			
Nº	Cargo	Costo Individual	Costo Total
1	Ing. Sistema (Líder del Proyecto)	1200,00	1200,00
2	Analista/Diseñador	960,00	1920,00
1	Ingeniero del Software	840,00	840,00
1	Programador	720,00	720,00
		Total	4680,00



Recursos Tecnológicos			
Hardware			
Cantidad	Descripción	Costo/Hora	Total
2	140 horas Computadora	0,80	224,00
1	Impresora Lexmark X3350 (depreciación 240/16*1)		15,00
Software			
Cantidad	Descripción	Costo/Hora	Costo Total
1	Licencia Microsoft Office		400,00
Total			639,00



Recursos Materiales			
Cantidad	Descripción	Costo	Total
1	Resma de Papel A4	3,00	3,00
2	Cartuchos para Impresora	30,00	60,00
40	Transporte a la empresa	0,25	10,00
20	Viáticos	30	60,00
Total			133,00

Flujo de Pago	
Recursos	Costos
Recursos Humanos	4680,00
Recursos Tecnológicos	639,00
Recursos Materiales	133,30
Imprevistos (10%)	545,23
Total	5997,53



Costo de Operación	
Descripción	Costo Total
Suministros y Gastos	600,00
Diseñador Gráfico	850,00
Total	1450,00

Factibilidad Técnica

Debido a tratarse de un sistema pequeño, pero sumamente importante, sus costos son bajos y la necesidad de equipos potentes es innecesaria, motivo por el cual un equipo pequeño y una impresora común funcionarían adecuadamente con el sistema.

En la empresa Multioro se cuenta con estos requisitos, por lo que no habría necesidad de adquirirlos. En cuanto nuestros conocimientos son suficientes para el desarrollo y cumplimiento del software. Se debe de concluir indicando si es factible o no desarrollar técnicamente el proyecto.

Arquitecturas de software

Una arquitectura de software describe los componentes básicos de un sistema de software y su combinación interna.

Las **características** de la arquitectura de software es poder funcionar con éxito, la arquitectura de software debe ser sintonizada con los restantes factores del proyecto de software. Una arquitectura de software bien configurada facilita a los usuarios y desarrolladores la comprensión del sistema. Factores importantes que influyen la aptitud de la arquitectura de software son la planificación de proyectos, el análisis de riesgo, la organización, el proceso de desarrollo, los ciclos de trabajo, el hardware, la garantía de calidad y los requerimientos.



TCP/IP

Es una denominación que permite identificar al grupo de protocolos de red que respaldan a Internet y que hacen posible la transferencia de datos entre redes de ordenadores. En concreto, puede decirse que TCP/IP hace referencia a los dos protocolos más trascendentes de este grupo: el conocido como Protocolo de Control de Transmisión (o TCP) y el llamado Protocolo de Internet (presentado con la sigla IP).

En este sentido, es necesario subrayar que el primero de los protocolos citados lo que hace es proporcionar un transporte muy fiable de los datos dentro de lo que es el nivel de transporte de referencia OSI. Y mientras, el segundo, el protocolo IP se identifica y define especialmente por el hecho de que lo que hace, en el nivel de red, es ofrecernos la posibilidad de dirigir los citados a otras máquinas.

Existen cuatro niveles que son:

Nivel de aplicación:

Es el más alto dentro del protocolo que nos ocupa y en él se encuentran una serie de aplicaciones que tienen la capacidad de acceder a diversos servicios a los que se puede acceder vía Internet.

Nivel de transporte:

Es el encargado de ofrecer una comunicación entre extremos de programas de aplicación.

Nivel de red:

Se dedica a realizar una serie de acciones sobre la información que recibe del nivel anterior para luego acometer el envío al nivel que está por debajo de él.

Nivel de enlace:

Su misión más clara es transmitir la información que recibe al hardware.