SISTEMAS DE INFORMACION II

CICLOS DE VIDA Y MODELO DE PROCESOS DE SOFTWARE

Definición de Proceso:

Un **proceso** es un concepto manejado por el sistema operativo que consiste en el conjunto formado por:

- Las instrucciones de un programa destinadas a ser ejecutadas por el microprocesador.
- Su estado de ejecución en un momento dado, esto es, los valores de los registros de la CPU para dicho programa.
- Su memoria de trabajo, es decir, la memoria que ha reservado y sus contenidos.
- Otra información que permite al sistema operativo su planificación.

Esta definición varía ligeramente en el caso de sistemas operativos multihilo, donde un proceso consta de uno o más *hilos*, la memoria de trabajo (compartida por todos los hilos) y la información de planificación. Cada hilo consta de instrucciones y estado de ejecución.

Los procesos son creados y destruidos por el sistema operativo, así como también este se debe hacer cargo de la comunicación entre procesos, pero lo hace a petición de otros procesos. El mecanismo por el cual un proceso crea otro proceso se denomina bifurcación (*fork*). Los nuevos procesos son independientes y no comparten memoria (es decir, información) con el proceso que los ha creado.

En los sistemas operativos multihilo es posible crear tanto hilos como procesos. La diferencia estriba en que un proceso solamente puede crear hilos para sí mismo y en que dichos hilos comparten toda la memoria reservada para el proceso.

Ciclo de vida de un proyecto de Software:

Se entiende por *ciclo de vida de un proyecto software* a todas las etapas por las que pasa un proyecto, desde la concepción de la idea que hace surgir la necesidad de diseñar un sistema software, pasando por el análisis, desarrollo, implantación y mantenimiento del mismo y hasta que finalmente muere por ser sustituido por otro sistema.

Aunque UML es bastante independiente del proceso, para obtener el máximo rendimiento de UML se debería considerar un proceso que fuese:

Dirigido por los *casos de uso*, o sea, que los casos de uso sean un artefacto básico para establecer el comportamiento del deseado del sistema, para validar la arquitectura, para las pruebas y para la comunicación entre las personas involucradas en el proyecto.

Centrado en la *arquitectura* de modo que sea el artefacto básico para conceptuar, construir, gestionar y hacer evolucionar el sistema.

Un proceso *iterativo*, que es aquel que involucra la gestión del flujo de ejecutables del sistema, e *incremental*, que es aquel donde cada nueva versión corrige defectos de la anterior e incorpora nueva funcionalidad. Un proceso iterativo e incremental se denomina *dirigido por el riesgo*, lo que significa que cada nueva versión se ataca y reducen los riesgos más significativos para el éxito del proyecto.

Este proceso, dirigido a los casos de uso, centrado en la arquitectura, iterativo e incremental pude

descomponerse en fases, donde cada fase es el intervalo de tiempo entre dos hitos importantes del proceso, cuando se cumplen los objetivos bien definidos, se completan los artefactos y se toman decisiones sobre si pasar o no a la siguiente fase.

En el ciclo de vida de un proyecto software existen cuatro fases. La **iniciación**, que es cuando la idea inicial está lo suficientemente fundada para poder garantizar la entrada en la fase de **elaboración**, esta fase es cuando se produce la definición de la arquitectura y la visión del producto. En esta fase se deben determinar los requisitos del sistema y las pruebas sobre el mismo.

Posteriormente se pasa a la fase de **construcción**, que es cuando se pasa de la base arquitectónica ejecutable hasta su disponibilidad para los usuarios, en esta fase se reexaminan los requisitos y las pruebas que ha de soportar. La **transición**, cuarta fase del proceso, que es cuando el software se pone en mano de los usuarios. Raramente el proceso del software termina en la etapa de transición, incluso durante esta fase el proyecto es continuamente reexaminado y mejorado erradicando errores y añadiendo nuevas funcionalidades no contempladas.

Un elemento que distingue a este proceso y afecta a las cuatro fases es una ***iteración,*** que es un conjunto bien definido de actividades, con un plan y unos criterios de evaluación, que acaban en una versión del producto, bien interna o externa.

Ciclo de Vida:

Todo proyecto de ingeniería tiene unos fines ligados a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto. Al **conjunto de las fases empleadas se le denomina ciclo de vida**.

Sin embargo, la forma de agrupar las actividades, los objetivos de cada fase, los tipos de productos intermedios que se generan, etc. pueden ser muy diferentes dependiendo del tipo de producto o proceso a generar y de las tecnologías empleadas.

La complejidad de las relaciones entre las distintas actividades crece exponencialmente con el tamaño, con lo que rápidamente se haría inabordable si no fuera por la vieja táctica de divide y vencerás. De esta forma la división de los proyectos en fases sucesivas es un primer paso para la reducción de su complejidad, tratándose de escoger las partes de manera que sus relaciones entre sí sean lo más simples posibles.

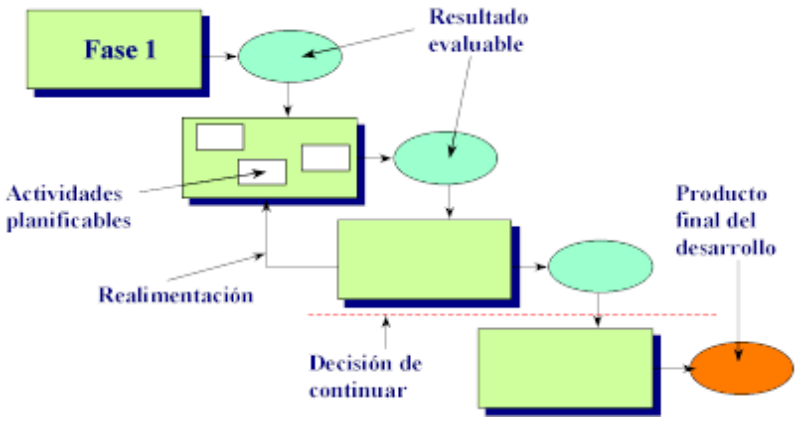
La definición de un ciclo de vida facilita el **control sobre los tiempos** en que es necesario aplicar recursos de todo tipo (personal, equipos, suministros, etc.) al proyecto. Si el proyecto incluye subcontratación de partes a otras organizaciones, el **control del trabajo subcontratado** se facilita en la medida en que esas partes encajen bien en la estructura de las fases. El **control de calidad** también se ve facilitado si la separación entre fases se hace corresponder con puntos en los que ésta deba verificarse (mediante comprobaciones sobre los productos parciales obtenidos).

De la misma forma, la práctica acumulada en el diseño de modelos de ciclo de vida para situaciones muy diversas permite que nos beneficiemos de la **experiencia adquirida** utilizando el enfoque que mejor de adapte a nuestros requerimientos.

Elementos de un Ciclo de Vida:

Un ciclo de vida para un proyecto se compone de **fases sucesivas** compuestas por tareas planificables. Según el modelo de ciclo de vida, la sucesión de fases puede ampliarse con **bucles de realimentación**, de manera

que lo que conceptualmente se considera una misma fase se pueda ejecutar más de una vez a lo largo de un proyecto, recibiendo en cada pasada de ejecución aportaciones de los resultados intermedios que se van produciendo (realimentación).

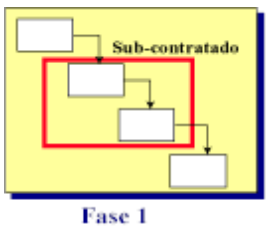


Para un adecuado control de la progresión de las fases de un proyecto se hace necesario especificar con suficiente precisión los resultados evaluables, o sea, productos intermedios que deben resultar de las tareas incluidas en cada fase. Normalmente estos productos marcan los hitos entre fases.

A continuación presentamos los distintos elementos que integran un ciclo de vida:

- **Fases**. Una fase es un conjunto de actividades relacionadas con un objetivo en el desarrollo del proyecto. Se construye agrupando tareas (actividades elementales) que pueden compartir un tramo determinado del tiempo de vida de un proyecto. La agrupación temporal de tareas impone requisitos temporales correspondientes a la asignación de recursos (humanos, financieros o materiales).

  Cuanto más grande y complejo sea un proyecto, mayor detalle se necesitará en la definición de las fases para que el contenido de cada una siga siendo manejable. De esta forma, cada fase de un proyecto puede considerarse un *micro−proyecto* en sí mismo, compuesto por un conjunto de micro−fases.



Otro motivo para descomponer una fase en subfases menores puede ser el interés de separar partes temporales del proyecto que se subcontraten a otras organizaciones, requiriendo distintos procesos de gestión.

Cada fase viene definida por un conjunto de elementos observables externamente, como son las **actividades** con las que se relaciona, los **datos de entrada** (resultados de la fase anterior, documentos o productos requeridos para la fase, experiencias de proyectos anteriores), los **datos de salida** (resultados a utilizar por la fase posterior, experiencia acumulada, pruebas o resultados efectuados) y la **estructura interna** de la fase.

*Esquema general de **operación** de una fase*

- **Entregables** ("*deliverables*"). Son los productos intermedios que generan las fases. Pueden ser materiales (componentes, equipos) o inmateriales (documentos, software). Los entregables permiten evaluar la marcha del proyecto mediante comprobaciones de su adecuación o no a los requisitos funcionales y de condiciones de realización previamente establecidos. Cada una de estas evaluaciones puede servir, además, para la toma de decisiones a lo largo del desarrollo del proyecto.

Tipos de Modelos de un Ciclo de Vida:

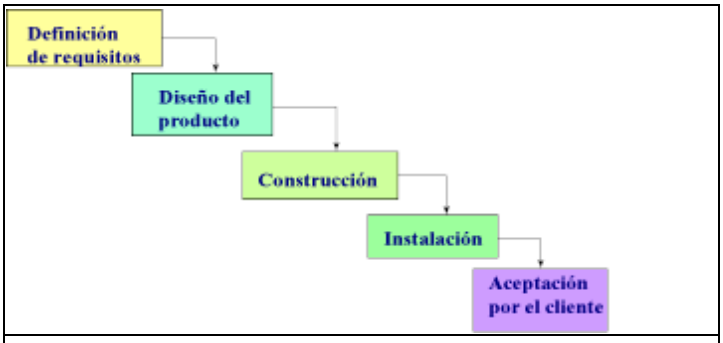
Las principales diferencias entre distintos modelos de ciclo de vida están en:

- El **alcance** del ciclo dependiendo de hasta dónde llegue el proyecto correspondiente. Un proyecto puede comprender un simple estudio de viabilidad del desarrollo de un producto, o su desarrollo completo o, llevando la cosa al extremo, toda la historia del producto con su desarrollo, fabricación, y modificaciones posteriores hasta su retirada del mercado.
- Las **características** (contenidos) de las fases en que dividen el ciclo. Esto puede depender del propio tema al que se refiere el proyecto (no son lo mismo las tareas que deben realizarse para proyectar un avión que un puente), o de la organización (interés de reflejar en la división en fases aspectos de la división interna o externa del trabajo).
- La **estructura** de la sucesión de las fases que puede ser lineal, con prototipado, o en espiral. Veámoslo con más detalle:

Ciclo de Vida Lineal:

Es el más utilizado, siempre que es posible, precisamente por ser el más sencillo. Consiste en descomponer la actividad global del proyecto en fases que se suceden de manera lineal, es decir, cada una **se realiza una sola vez**, cada una se realiza **tras la anterior y antes que la siguiente**. Con un ciclo lineal es fácil dividir las tareas entre equipos sucesivos, y prever los tiempos (sumando los de cada fase).

Requiere que la actividad del proyecto pueda descomponerse de manera que una fase no necesite resultados de las siguientes (realimentación), aunque pueden admitirse ciertos supuestos de realimentación correctiva. Desde el punto de vista de la gestión (para decisiones de planificación), requiere también que se sepa bien de antemano lo que va a ocurrir en cada fase antes de empezarla.



Ejemplo de ciclo lineal para un proyecto de construcción

Ciclo de vida con Prototipado:

A menudo ocurre en desarrollos de productos con innovaciones importantes, o cuando se prevé la utilización de tecnologías nuevas o poco probadas, que las incertidumbres sobre los resultados realmente alcanzables, o las ignorancias sobre el comportamiento de las tecnologías, impiden iniciar un proyecto lineal con especificaciones cerradas.

Si no se conoce exactamente cómo desarrollar un determinado producto o cuáles son las especificaciones de forma precisa, suele recurrirse a definir especificaciones iniciales para hacer un **prototipo**, o sea, un producto parcial (no hace falta que contenga funciones que se consideren triviales o suficientemente probadas) y provisional (no se va a fabricar realmente para clientes, por lo que tiene menos restricciones de coste y/o prestaciones). Este tipo de procedimiento es muy utilizado en desarrollo avanzado.

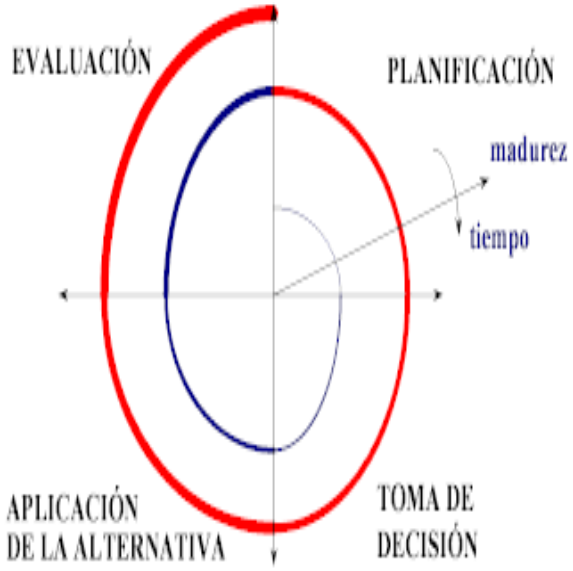
La experiencia del desarrollo del prototipo y su evaluación deben permitir la **definición de las especificaciones más completas** y seguras para el producto definitivo.

A diferencia del modelo lineal, puede decirse que el ciclo de vida con prototipado repite las fases de definición, diseño y construcción dos veces: para el prototipo y para el producto real.

Ciclo de Vida en Espiral:

El ciclo de vida en espiral puede considerarse como una generalización del anterior para los casos en que **no basta con una sola evaluación de un prototipo** para asegurar la desaparición de incertidumbres y/o ignorancias. El propio producto a lo largo de su desarrollo puede así considerarse como una sucesión de prototipos que progresan hasta llegar a alcanzar el estado deseado. En cada ciclo (espirales) las especificaciones del producto se van resolviendo paulatinamente.

A menudo la **fuente de incertidumbres es el propio cliente**, que aunque sepa en términos generales lo que quiere, no es capaz de definirlo en todos sus aspectos sin ver como unos influyen en otros. En estos casos la evaluación de los resultados por el cliente no puede esperar a la entrega final y puede ser necesaria repetidas veces.

	El esquema del ciclo de vida para estos casos puede representarse por un bucle en espiral, donde los cuadrantes son, habitualmente, fases de **especificación, diseño, realización y evaluación** (o conceptos y términos análogos). En cada vuelta el producto gana en *madurez* (aproximación al final deseado) hasta que en una vuelta la evaluación lo apruebe y el bucle pueda abandonarse.

Modelos de Procesos de Software:

Al día de hoy, ha aumentado la complejidad con la que se desarrollan sistemas de información para la industria, por lo que resulta difícil generar productos que cumplan cabalmente con las expectativas del cliente.

Para responder a esta situación, han surgido una serie de herramientas, técnicas y modelos que facilitan a las organizaciones, encargadas de las tecnologías de la información, generar productos que cumplan las expectativas del cliente e incluso las rebasen, herramientas que prometen ser la solución a los problemas de calidad, costo y tiempos de desarrollo; de éstas podemos mencionar a los modelos de calidad como la norma ISO 9000–2000, la ISO/IEC TR 15504 y el modelo CMM (Capability Maturity Model del Software Engineerig Institute SEI).

Aunque en el pasado se reconocía la necesidad de crear software de calidad, no se había hecho un esfuerzo serio para que nuestra industria generara productos que nos dieran la oportunidad de competir en el mercado internacional, con calidad equiparable o superior a la de países como la India o Irlanda. Afortunadamente, dicha situación ha cambiado; nuestro gobierno en conjunto con la industria, ha iniciado un esfuerzo serio para impulsar la industria del software a través del Programa para el Desarrollo de la Industria del Software (PROSOFT).

PROSOFT reconoce el estado incipiente de la industria mexicana de software, así como la necesidad de invertir cantidades crecientes de recursos en capital de tecnologías de información con objeto de contribuir de manera sostenible al crecimiento de la economía y la generación de empleos bien remunerados.

Con el programa, se pretende establecer una industria de software competitiva internacionalmente y asegurar su crecimiento a largo plazo, lo que situaría a México como líder de esta industria en Latinoamérica en 2012, además de convertirlo en líder desarrollador de soluciones de tecnologías de información de alta calidad y uso de software en Latinoamérica.

- Este programa tiene siete estrategias de donde emergen varios proyectos que ayudarán a que se alcancen las metas previstas en éste:
- Promover las exportaciones y la atracción de inversiones.
- Educar y formar personal competente en el desarrollo de software, en cantidad y calidad convenientes.

- Contar con un marco legal promotor de la industria.
- Desarrollar el mercado interno.
- Fortalecer a la industria local.
- Alcanzar niveles internacionales en capacidad de procesos.
- Promover acciones conjuntas con los gobiernos estatales y construir infraestructura.

Para el caso de la estrategia 6, la Asociación Mexicana para la Calidad en Ingeniería de Software (AMCIS), con el auspicio de la Secretaría de Economía propone un modelo concebido, diseñado y desarrollado por mentes mexicanas, adecuado para las necesidades específicas de México y con ventajas respecto de otros. El nuevo modelo, denominado MoProSoft, ofrece características que los otros no tienen de manera independiente; para su concepción, se tomaron las mejores prácticas de los otros modelos y se integraron y mejoraron otras; a continuación, mencionamos a qué se refiere cada modelo y algunas de sus ventajas y desventajas.

Norma ISO 9000–2000

Es una norma internacional destinada a evaluar la capacidad de la organización para cumplir los requisitos del cliente, los reglamentarios y los propios de la organización.

Ventajas

- Tiene un mecanismo de certificación bien establecido.
- Está disponible y es conocida.

Desventajas

- No es específica para la industria de software.
- No es fácil de entender.
- No está definida como un conjunto de procesos.
- No es fácil de aplicar.

Capability Maturity Model (CMM)

Es un marco evolutivo organizado en cinco niveles para lograr la mejora continua de procesos.

Ventajas

- Específico para el desarrollo y mantenimiento de software.
- Definido como un conjunto de áreas clave de procesos.
- Tiene un modelo de evaluación.
- Desde 1998 empezó a popularizarse en México.
- Existen organizaciones evaluadas.

Desventajas

- Es un modelo extranjero, no internacional.
- No es fácil de entender (inglés, 18 KPAs, 220 páginas).
- No es fácil de aplicar (pensado para organizaciones grandes).
- La mejora no está enfocada directamente a los objetivos de negocio.
- La evaluación es costosa y no tiene periodo de vigencia.
- Se está abandonando a favor de CMM–I (el SEI dejará de dar soporte a partir del 2005).

ISO/IEC TR 15504

Define el modelo de referencia de procesos de software y de capacidades de procesos que constituyen la base para la evaluación de procesos de software. Se compone de 9 partes de las cuales la 2, 3 y 9 son normativas y las demás informativas.

Ventajas

- Específico para el desarrollo y mantenimiento de software.
- Fácil de entender (24 procesos, 16 páginas).
- Definido como un conjunto de procesos.
- Orientado a mejorar los procesos para contribuir a los objetivos del negocio.

Desventajas

- No es práctico ni fácil de aplicar.
- Tiene solamente lineamientos para un mecanismo de evaluación.
- Todavía no es norma internacional.

MoProSoft

Es un Modelo de Procesos para la Industria de Software que fomenta la estandarización de su operación, a través de la incorporación de las mejores prácticas en gestión e ingeniería de software. La adopción del modelo permite elevar la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.

Ventajas

- Fácil de entender.
- Fácil de aplicar.
- No es costoso en su adopción.
- Sirve de base para alcanzar evaluaciones exitosas con otros modelos o normas, tales como ISO 9000:2000 [1] o CMM.1 V1.1[2].

A decir de sus creadores, el modelo está orientado a pequeñas y medianas empresas, hecho favorable si se considera que aproximadamente el 80% de las empresas desarrolladoras de software del país caen en esta categoría. Su principal fortaleza es que integra varias de las prácticas propuestas por los otros modelos y corrige algunas de sus desventajas, como son el hecho de que no ha sido liberado por completo o al menos falta el modelo de evaluación; además, está en proceso de convertirse en norma compitiendo con el proyecto de norma ISO/IEC TR 15504 y aunque no ha sido probado, se planea realizar pilotos en algunas organizaciones para evaluar qué tan fácil resulta su implantación determinando los recursos necesarios.

Lo interesante para nosotros como academia, es que tenemos la oportunidad de proponer productos concebidos y desarrollados en México, adecuados a nuestros requerimientos y realidad, lo que repercute de manera directa en la gama de soluciones que tiene una organización para resolver sus necesidades.

Modelo de Cascada:

En Ingeniería de software el **desarrollo en cascada**, también llamado **modelo en cascada**, es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

Un ejemplo de una metodología de desarrollo en cascada es:

- Análisis de requisitos
- Diseño
- Codificación
- Pruebas
- Implantación
- Mantenimiento

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costes del desarrollo. La palabra *cascada* sugiere, mediante la metáfora de la fuerza de la gravedad, el esfuerzo necesario para introducir un cambio en las fases más avanzadas de un proyecto.

Si bien ha sido ampliamente criticado desde el ámbito académico y la industria, sigue siendo el paradigma más seguido al día de hoy.

Fases del modelo

Análisis de requisitos

Se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (Documento de Especificación de Requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

Diseño

Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Codificación

Es la fase de programación propiamente dicha. Aquí se desarrolla el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores.

Pruebas

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente antes de ser puesto en explotación.

Implantación

El software obtenido se pone en producción.

Mantenimiento

Durante la explotación del sistema software pueden surgir cambios, bien para corregir errores o bien para introducir mejoras. Todo ello se recoge en los Documentos de Cambios.

Variantes

Existen variantes de este modelo; especialmente destacamos la que hace uso de prototipos y en la que se establece un ciclo antes de llegar a la fase de mantenimiento, verificando que el sistema final este libre de fallos

Modelo de Prototipo:

Modelo de Riesgo y Espiral:

Qué es el Modelo Espiral?

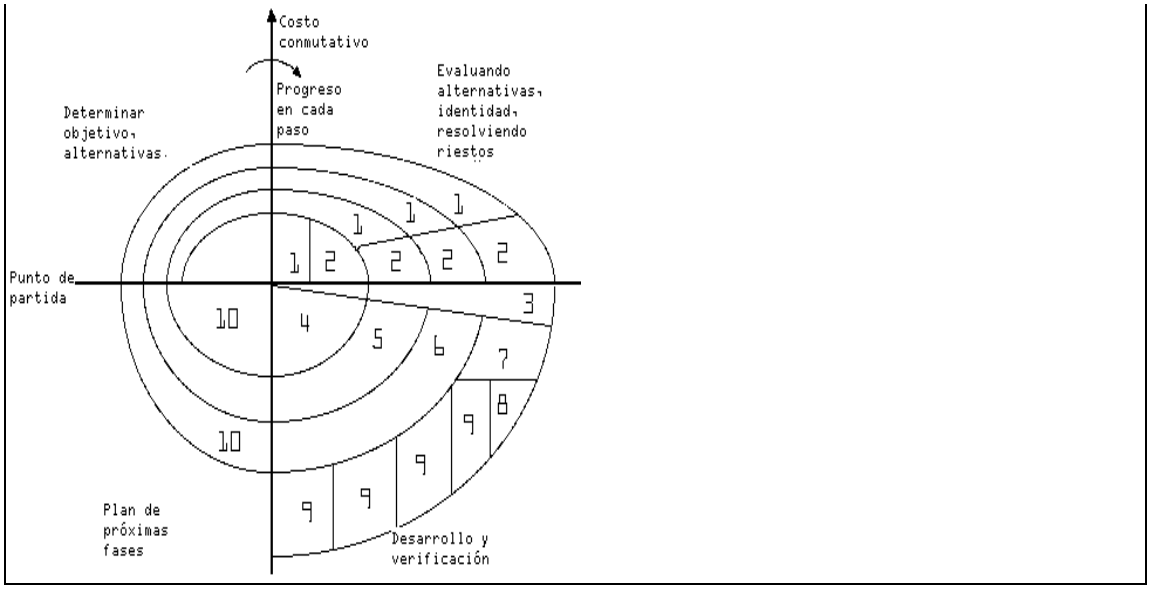
Desarrollado por B. Boehm, básicamente, la idea es Desarrollo Evolutivo, usando el Modelo de Cascada para cada etapa; esta orientado a evitar riesgos de trabajo. No define en detalle el sistema completo a la primera. Los desarrollares deberían solamente definir las mas altas prioridades. Definir e implementarlas y entonces obtener un feedback de los usuarios (tal y como feedback distingue desarrollo "evolutivo" de "incremental"). Con este conocimiento, deberían entonces retroceder o volver al punto de partida para definir e implementar más y mejores partes.

El Modelo Espiral mejora el Modelo de Cascada enfatizando la naturaleza iterativa del proceso de diseño. Eso introduce un ciclo de prototipo iterativo. En cada iteración, las nuevas expresiones que son obtenidas transformando otras dadas son examinadas para ver si representan progresos hacia el objetivo.

Este método esta basado en dos importantes principios:

- la practica de diseñó profesional es caracterizar en términos de conocer, actuar en situaciones, conversación con la situación y reflexión en acción. Hay un distinto medio de proceso – orientación en esta aproximación al diseño. Es raro que el diseñador tenga el diseño en su cabeza por adelantado y que después meramente lo transcriba. Gran parte del tiempo del diseñador esta inmiscuido en una progresiva relación con su entorno. Una buena metáfora para describirlo es "la conversación con el material", como un escultor, quien esta ocupado en una conversación con el medio. El escultor modela arcilla y luego mira y siente la escultura para ver lo que ha llegado a ser.
- la necesidad para diseñadores de tomar la practica de trabajo seriamente, de supervisar las formas en las que el trabajo se esta haciendo, en el sentido de una solución abierta y desplegada para aumentar la complejidad de una situación que el diseñador solo entiende parcialmente. El hecho por el cual se esta tratando con "actores humanos". Los sistemas necesitan tratar o estar en contacto con las preocupaciones del usuario. Es, definitiva, el reconocimiento de que el trabajo es fundamentalmente social, envolviendo cooperación y comunicación.

Una visión general del Modelo de Cascada puede ser la siguiente:



La Gerencia de Proyectos de Software:

En un contexto complejo, agudizado por la superficialidad generalizada con que se asumen las fases previas al inicio de los proyectos de software, emerge el gerente de proyecto. El líder cuyo objetivo consiste en mediar entre intereses y motivaciones diversas con el fin de llegar a una solución cuyos dividendos sean positivos tanto para el cliente como para el proveedor a quien representa. La gerencia de proyectos es un área que ha tenido un vasto desarrollo conceptual en las últimas 2 décadas. Iniciativas como la del Project Management Institute, PMI. Pretenden asentar las bases que conduzcan a la estandarización y clasificación de las actividades propias de la gerencia en diferentes áreas de conocimiento. Sin embargo, al igual que ocurre en otras ciencias relacionadas con el software, el desarrollo teórico y conceptual avanza vertiginosamente mientras los resultados de su aplicación en la ejecución de proyectos de software aún se encuentran en deuda.

El mercado de soluciones de TI. Ha impuesto una serie de condiciones a la Gerencia de los proyectos que la han hecho particularmente distante de la gerencia de proyectos en otras áreas.

Al iniciar un proyecto de software es normal que el alcance del proyecto no esté definido, no existen métricas que brinden una idea del tamaño del proyecto, no hay una metodología, buena o mala, a seguir durante la ejecución del proyecto. Resulta normal también que al inicio del proyecto se tenga un grupo de personas que nunca ha trabajado como equipo, un buen porcentaje de las personas desconoce la tecnología con la cuál debe trabajar y con una alta probabilidad es el primer proyecto que el gerente asume, cargo al que llegó gracias a que es un desarrollador destacado.

El cliente, factor determinante del éxito de un proyecto de software

Aunque a primera vista el proveedor de la solución es el directo responsable de los resultados del proyecto, una adecuada planeación, gestión y evaluación del avance constituyen herramientas fundamentales para la toma de decisiones del cliente.

Un alto porcentaje de los riesgos inducidos en los proyectos de tecnologías de información se deben a deficiencias en la planeación inicial. Definición pobre del alcance del proyecto y asignación de recursos, presupuesto y cronograma aislada de los objetivos que se quieren alcanzar son algunos de los riesgos comunes que inducen los clientes en las primeras instancias de los proyectos.

El mercado de soluciones de software demanda estrategias orientadas hacia la definición de cronogramas y costos de los proyectos en instancias incipientes, en las cuales no se ha efectuado una definición detallada de

requerimientos hecho que obliga a tener amplios márgenes de error en la estimación de tiempo y costo en los proveedores.

Algunos proveedores de soluciones de software optan por mitigar el riesgo elevando el costo de sus servicios con el fin poder asumir la responsabilidad en caso tal que el riesgo se concrete. Esta práctica disminuye significativamente la competitividad del proveedor y eleva innecesariamente los costos de desarrollo para los clientes, hecho que aumenta su inversión general en soluciones de software pero empobrece el alcance de los proyectos.

Resulta conveniente efectuar labores de planeación detallada al interior de las organizaciones cliente, previas al inicio de los proyectos y a la apertura de concursos y licitaciones. Dichas tareas permitirán establecer con un buen nivel de detalle el objetivo que se pretende alcanzar con el desarrollo e implantación de la solución, las prestaciones o funcionalidad necesarias en la solución, una adecuada priorización de los requerimientos y una asignación de recursos, no solo económicos sino humanos, responsables de manejar el flujo de información y requerimientos cliente–proveedor y proveedor–cliente.

La planeación detallada conducirá a obtener los máximos beneficios al invertir en soluciones de TI, adicionalmente le dará elementos para la toma de mejores decisiones en la ejecución del proyecto y será una herramienta para la evaluación de sus proveedores de tecnología.

La próxima vez que su proveedor de soluciones de software le pida profundizar en el análisis, previo a establecer compromisos de presupuesto y cronograma, seguramente se debe a que está interesado en ofrecerle una solución que se ajuste a su necesidad y de ofrecerle una perspectiva realista de planeación.

Funciones de gerencia de proyectos de software:

Partamos de la siguiente definición de administración de un proyecto:

"La administración de un proyecto, se define como un sistema de procedimientos, prácticas, tecnologías y conocimientos del tema que permiten la planificación, organización, designación de personal, dirección y control necesarios, para poder manejar con éxito un proyecto de ingeniería de software" 1

Por tanto Tahyer, es concreto en afirmar que el "concepto de la universalidad de la administración nos permite usar las funciones de administración y actividades básicas de las principales corrientes de administración, como las funciones y actividades de alto nivel de la administración de proyectos de ingeniería de software". En consecuencia, es necesario conocer cuales son las actividades y funciones de la administración:

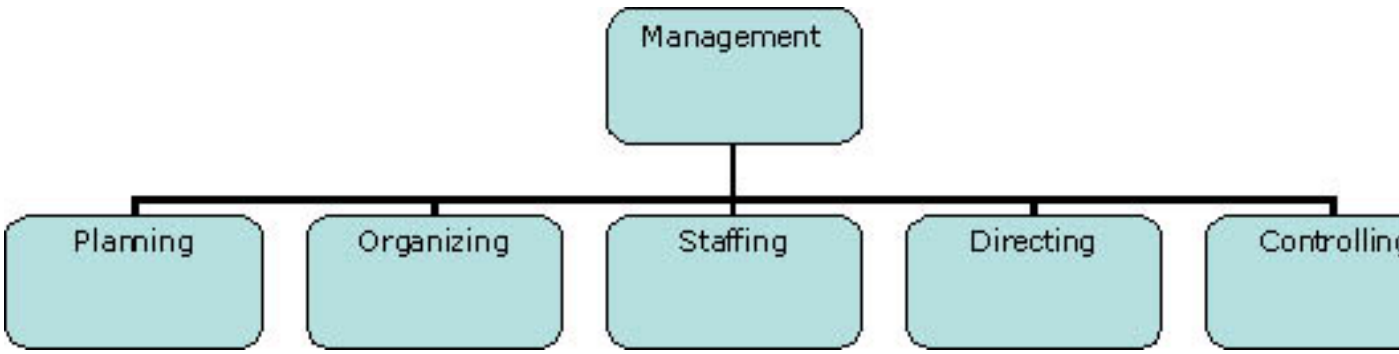


Figura 1: Modelo de funciones de gerencia

ACTIVIDAD	DEFINICIÓN
Planificación	Pre–determinar un curso de acción para lograr los objetivos de la

	organización
Organización	Arreglar y relacionar el trabajo para lograr los objetivos y para delegar responsabilidad y autoridad para lograr esos objetivos
Dotación de personal	Seleccionar y capacitar al personal para los puestos en la organización
Dirección	Crear una atmósfera que ayudará y motivará a la gente para lograr los resultados finales deseados
Control	Medir y corregir el rendimiento de las actividades que se dirigen hacia los objetivos, de acuerdo a lo planificado

Tabla 1: Principales funciones de la administración

Funciones de gerencia	**Actividades fundamentales de la gerencia**
Planificación	• Determinar los objetivos o metas • Desarrollar estrategias • Determinar cursos de acción • Tomar decisiones • Establecer procedimientos y reglas • Desarrollar programas • Predecir situaciones futuras • Preparar presupuestos • Documentar los planes del proyecto
Organización	• Identificar y agrupar las tareas requeridas • Seleccionar y establecer las estructuras organizativas • Definir responsabilidades y autoridad • Documentar estructuras organizativas
Dotación de personal	• Identificar y agrupar las tareas requeridas • Seleccionar y establecer las estructuras organizativas • Definir responsabilidades y autoridad • Documentar estructuras organizativas
Dotación de personal	• Llenar los puestos de la organización • Asimilar personal asignado recientemente • Educar o entrenar al personal • Hacer previsiones para el desarrollo general • Compensar • Finalizar las asignaciones • Documentar las decisiones referentes a la asignación de personal
Dirección	• Proporcionar liderazgo • Supervisar al personal • Delegar autoridad • Motivar al personal • Coordinar actividades • Facilitar las comunicaciones • Resolver conflictos

	• Manejar los cambios • Documentar las decisiones de dirección
Control	• Desarrollar estándares de rendimiento • Establecer sistemas de monitoreo e informes • Medir los resultados • Iniciar acciones correctivas • Premios y disciplina • Documentar métodos de control

Tabla 2: Principales actividades de gerencia

Por tanto un gerente de proyectos, planifica, dirige, dota de personal, y controla un proyecto de ingeniería de software.

Actividades de la administración de proyectos de ingeniería de software
Ahora nos enfocaremos a la segunda pregunta que nos hicimos: ¿Cómo administrar proyectos de ingeniería de software?

- **Planificación**

Imagine por un momento que usted es director técnico de un equipo de fútbol, y tiene un juego muy importante dentro de un par de semanas, ¿usted elaboraría su estrategia de juego y táctica antes o durante el juego? de esta respuesta dependerá el éxito o fracaso del encuentro. De manera análoga, si usted es Gerente de un Proyecto es vital que elabore un plan de proyecto. En este plan, usted definirá, de acuerdo a las necesidades de su cliente, los objetivos y metas del proyecto, factores críticos de éxito, estimará los tiempos de cada actividad y tarea del proyecto, analizará los riesgos del proyecto, elaborará el presupuesto del proyecto. Este plan es la base que le permitirá después, hacer una evaluación post–proyecto y determinar si el proyecto fue exitoso o no. En esta parte debe plantearse por ejemplo, cuales serán los entregables del proyecto, entregará documentación, subsistemas, etc. Además así como un director técnico, plantea una estrategia de juego para el encuentro, de manera análoga deberá elegir que metodología de desarrollo utilizará, ¿el ciclo de vida tradicional? ¿el modelo en espiral? ¿RUP? ¿Programación extrema?, etc., cual de todas las metodologías de desarrollo de software o de ingeniería de software existentes o que combinación de ellas, es la mas adecuada para lograr el éxito del proyecto. Es por eso mi estimado lector, que usted deberá de tener un sólido conocimiento de ingeniería de software como lo había mencionado antes, no solo basta ser un buen administrador, también debe conocer bien el terreno de acción.

- **Organización**

Un proyecto así como una empresa, tiene una estructura organizacional, esta estructura tenderá a ser compleja cuando el proyecto tenga más entes involucrados y el proyecto en sí sea más complejo. Esto se cumple cuando usted tiene que organizar un proyecto en donde no solo intervienen el proveedor y el cliente, sino, por ejemplo, otros proveedores de su cliente o proveedores del proveedor por ejemplo. Es necesario determinar las responsabilidades para las actividades y tareas, línea de mando de cada ente. Esta actividad consiste en organizar actividades y tareas, agruparlas, definir roles y relaciones de autoridad en el proyecto, es decir diseñar una estructura organizativa del proyecto. Esta estructura organizativa, solo tendrá validez durante la existencia del proyecto, una vez culminado este, esta organización desaparecerá. Por ejemplo, usted puede ser analista de sistemas para su empresa, pero en la organización del proyecto, usted puede tener el rol de jefe del equipo de desarrollo, es por eso que el proyecto tiene una organización propia, donde intervienen gerentes de

proyectos, ingenieros de software, analistas funcionales, arquitectos de software, desarrolladores, especialistas en redes y comunicaciones, soporte técnico, etc.

- **Dotación de personal**

Una vez que el proyecto fue aprobado y deberá ser ejecutado, usted tiene que solicitar y/o asignar al personal adecuado para el proyecto, así como es necesario para un director técnico de un equipo de fútbol, tener una táctica para asegurar el triunfo de su equipo, es necesario también formar el mejor equipo. Esto es parte de la estrategia y de las decisiones que usted deberá tomar antes de iniciar el proyecto. Usted tiene que tener claro, de acuerdo a la organización del proyecto diseñada, que especialistas necesita, con que conocimientos, años de experiencia, perfil profesional. En esta actividad usted deberá seleccionar, evaluar y asignar al personal idóneo para el proyecto. Y no solo deberá basarse en la parte profesional del personal, sino también deberá tomar en cuenta las fortalezas y debilidades personales de cada recurso, pues la parte personal es muy importante, es una variable a tomar en cuenta, es por eso que la productividad y el nivel de compromiso del proyecto varía de un individuo a otro.

- **Dirección**

Aquí llegamos a un punto crítico, usted tiene que tener claro que el título de Gerente de proyecto, es un título comercial y organizativo que le da su empresa, pero usted en realidad debe ser un líder para el personal involucrado en el proyecto. Un líder que dirija al personal por el camino del éxito del proyecto, un líder que trasmita al personal involucrado en el proyecto, el porque ellos deben poner el mejor empeño y dedicación en el proyecto. Usted como líder conoce cual es el camino, es decir los objetivos y metas del proyecto, lo que el cliente quiere del proyecto, eso tiene que hacer que sean los objetivos y metas del personal. Recuerde algo muy importante, "cuando un proyecto es exitoso el mérito es del equipo del proyecto, y cuando el proyecto fracasa, la responsabilidad es del Gerente del Proyecto". Así que usted debe proporcionar liderazgo, esto implica motivar al personal, guiar al personal, supervisar el rendimiento del personal y delegar responsabilidades cuando sea necesario.

- **Control**

Esta actividad se lleva a cabo durante la ejecución del proyecto y marca un punto de quiebre, entre lo planificado y lo real, solo en un mundo ideal se dan las cosas exactamente como usted las planificó, es por eso que un gerente de proyecto debe medir el progreso del proyecto, esto implica conocer si las tareas del proyecto se están cumpliendo en las fechas programadas, si el presupuesto que planificó esta dentro de lo esperado, si los riesgos y en general situaciones futuras que usted planificó podrían impactar en el proyecto se están dando o no. La única manera de controlar es a través de la medición. ¿Cómo usted controla que un determinado recurso este invirtiendo el tiempo planificado en una determinada tarea? La respuesta es a través de un sistema de medición, por ejemplo puede hacer que el personal durante el curso del proyecto registre las horas invertidas en cada tarea, de manera que usted pueda identificar retrasos e implemente estrategias y/o mecanismos de acción para retomar el deathline del proyecto. Es mucho mas importante, medir el avance real del proyecto, esto no se mide en base a el dinero que a la fecha se ha invertido en el proyecto, o a las horas que ya han sido consumidas por los recursos, sino a través de entregables e hitos claros que marquen la finalización de cada actividad del proyecto y finalmente el conjunto de la éxitosa finalización de estas actividades lleven a la exitosa finalización del proyecto. En esta parte de vital importancia el feedback del proyecto hacia su gerencia y su cliente, es decir, elaborar informes de avance, gráficos y todos los medios que permitan comunicar del estado del proyecto tanto a su cliente como a su gerencia, y esto debe hacerse periódicamente, incluso deben hacerse reuniones de revisión del estado del proyecto. Así mismo, para un adecuado control, usted debe implementar mecanismos y/o sistemas de monitoreo efectivos que le permitan recopilar la información del estado del proyecto y poder plasmarlo finalmente en un informe de avance del proyecto.

Con respecto al esfuerzo que un Gerente de Proyectos deberá desplegar a lo largo de un proyecto, podemos decir, que el esfuerzo desarrollado por un gerente de proyectos, es fuerte al comienzo, baja a la mitad del proyecto y vuelve a ser pesado hacia el final.

Planificación y Proyectos de Software:

TEMA I. PLANIFICACION DE UN PROYECTO DE SISTEMAS.

1.1. Qué es un proyecto de Sistema o Software?

Es el Proceso de gestión para la creación de un Sistema o software, la cual encierra un conjunto de actividades, una de las cuales es la estimación, estimar es echar un vistazo al futuro y aceptamos resignados cierto grado de incertidumbre. Aunque la estimación, es mas un arte que una Ciencia, es una actividad importante que no debe llevarse a cabo de forma descuidada. Existen técnicas útiles para la estimación de costes de tiempo. Y dado que la estimación es la base de todas las demás actividades de planificación del proyecto y sirve como guía para una buena Ingeniería Sistemas y Software.

Al estimar tomamos en cuenta no solo del procedimiento técnico a utilizar en el proyecto, sino que se toma en cuenta los recursos, costos y planificación. El Tamaño del proyecto es otro factor importante que puede afectar la precisión de las estimaciones. A medida que el tamaño aumenta, crece rápidamente la interdependencia entre varios elementos del Software.

La disponibilidad de información Histórica es otro elemento que determina el riesgo de la estimación.

1.2. Objetivos de la Planificación del Proyecto.

El objetivo de la Planificación del proyecto de Software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos costos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente medida que progresa el proyecto. Además las estimaciones deberían definir los escenarios del mejor caso, y peor caso, de modo que los resultados del proyecto pueden limitarse.

El Objetivo de la planificación se logra mediante un proceso de descubrimiento de la información que lleve a estimaciones razonables.

1.3 Actividades asociadas al proyecto de software.

1.3.1 Ámbito del Software.

Es la primera actividad de llevada a cabo durante la planificación del proyecto de Software.

En esta etapa se deben evaluar la función y el rendimiento que se asignaron al Software durante la Ingeniería del Sistema de Computadora para establecer un ámbito de proyecto que no sea ambiguo, e incomprensible para directivos y técnicos

Describe la función, el rendimiento, las restricciones, las interfaces y la fiabilidad, se evalúan las funciones del ámbito y en algunos casos se refinan para dar mas detalles antes del comienzo de la estimación. Las restricciones de rendimiento abarcan los requisitos de tiempo de respuesta y procesamiento, identifican los límites del software originados por el hardware externo, por la memoria disponible y por otros sistemas existentes.

El Ámbito se define como un pre–requisito para la estimación y existen algunos elementos que se debe tomar

en cuenta como es:

- *La Obtención de la Información necesaria para el software. Para esto el analista y el cliente se reúnen sobre las expectativas del proyecto y se ponen de acuerdo en los puntos de interés para su desarrollo.*

1.4 **RECURSOS:**

La Segunda tarea de la planificación del desarrollo de Software es la estimación de los recursos requeridos para acometer el esfuerzo de desarrollo de Software, esto simula a una pirámide donde las Herramientas (hardware y Software), son la base proporciona la infraestructura de soporte al esfuerzo de desarrollo, en segundo nivel de la pirámide se encuentran los Componentes reutilizables.

Y en la parte mas alta de la pirámide se encuentra el recurso primario, las personas (el recurso humano).

Cada recurso queda especificado mediante cuatro características:

- *Descripción del Recurso.*
- *Informes de disponibilidad.*
- *Fecha cronológica en la que se requiere el recurso.*
- *Tiempo durante el que será aplicado el recurso.*

1.4.1 Recursos Humanos.

La Cantidad de personas requeridas para el desarrollo de un proyecto de software solo puede ser determinado después de hacer una estimación del esfuerzo de desarrollo (por ejemplo personas mes o personas años), y seleccionar la posición dentro de la organización y la especialidad que desempeñara cada profesional.

1.4.2 Recursos o componentes de software reutilizables.

Cualquier estudio sobre recursos de software estaría incompleto sin estudiar la reutilización, esto es la creación y la reutilización de bloques de construcción de Software.

Tales bloques se deben establecer en catálogos para una consulta más fácil, estandarizarse para una fácil aplicación y validarse para la también fácil integración.

El Autor Bennatan sugiere cuatro categorías de recursos de software que se deberían tener en cuenta a medida que se avanza con la planificación:

- *Componentes ya desarrollados.*
- *Componentes ya experimentados.*
- *Componentes con experiencia Parcial.*
- *Componentes nuevos.*

1.4.3 Recursos de entorno.

El entorno es donde se apoya el proyecto de Software, llamado a menudo entorno de Ingeniería de Software, incorpora Hardware y Software.

El Hardware proporciona una plataforma con las herramientas (Software) requeridas para producir los productos que son el resultado de la buena practica de la Ingeniería del Software, un planificador de proyectos debe determinar la ventana temporal requerida para el Hardware y el Software, y verificar que estos recursos estén disponibles. Muchas veces el desarrollo de las pruebas de validación de un proyecto de software para la composición automatizada puede necesitar un compositor de fotografías en algún punto durante el desarrollo. Cada elemento de hardware debe ser especificado por el planificador del Proyecto de Software.

1.5. ESTIMACION DEL PROYECTO DE SOFTWARE.

En el principio el costo del Software constituía un pequeño porcentaje del costo total de los sistemas basados en Computadoras. Hoy en día el Software es el elemento mas caro de la mayoría de los sistemas informáticos.

Un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y perdidas, la estimación del costo y del esfuerzo del software nunca será una ciencia exacta, son demasiadas las variables: humanas, técnicas, de entorno, políticas, que pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo.

Para realizar estimaciones seguras de costos y esfuerzos tienen varias opciones posibles:

- Deje la estimación para mas adelante (obviamente podemos realizar una estimación al cien por cien fiable después de haber terminado el proyecto.
- Base las estimaciones en proyectos similares ya terminados.
- Utilice técnicas de descomposición relativamente sencillas para generar las estimaciones de costos y esfuerzo del proyecto.
- Desarrolle un modelo empírico para él cálculo de costos y esfuerzos del Software.

Desdichadamente la primera opción, aunque atractiva no es práctica.

La Segunda opción puede funcionar razonablemente bien si el proyecto actual es bastante similar a los esfuerzos pasados y si otras influencias del proyecto son similares. Las opciones restantes son métodos viables para la estimación del proyecto de software. Desde el punto de vista ideal, se deben aplicar conjuntamente las técnicas indicadas usando cada una de ellas como comprobación de las otras.

Antes de hacer una estimación, el planificador del proyecto debe comprender el ámbito del software a construir y generar una estimación de su tamaño.

1.5.1 Estimación basada en el Proceso.

Es la técnica más común para estimar un proyecto es basar la estimación en el proceso que se va a utilizar, es decir, el proceso se descompone en un conjunto relativamente pequeño de actividades o tareas, y en el esfuerzo requerido para llevar a cabo la estimación de cada tarea.

Al igual que las técnicas basadas en problemas, la estimación basada en el proceso comienza en una delineación de las funciones del software obtenidas a partir del ámbito del proyecto. Se mezclan las funciones del problema y las actividades del proceso. Como ultimo paso se calculan los costos y el esfuerzo de cada función y la actividad del proceso de software.

1.6. DIFERENTES MODELOS DE ESTIMACION.

Existen diferentes modelos de estimación como son:

1.6.1 Los Modelos Empíricos:

Donde los datos que soportan la mayoría de los modelos de estimación obtienen una muestra limitada de proyectos. Por esta razón, el modelo de estimación no es adecuado para todas las clases de software y en todos los entornos de desarrollo. Por lo tanto los resultados obtenidos de dichos modelos se deben utilizar con prudencia.

1.6.2 El Modelo COCOMO.

Barry Boehm, en su libro clásico sobre economía de la Ingeniería del Software, introduce una jerarquía de modelos de estimación de Software con el nombre de COCOMO, por su nombre en Ingles (Constructive, Cost, Model) modelo constructivo de costos. La jerarquía de modelos de Boehm esta constituida por los siguientes:

- **Modelo I.** *El Modelo COCOMO básico calcula el esfuerzo y el costo del desarrollo de Software en función del tamaño del programa, expresado en las líneas estimadas.*

- **Modelo II.** *El Modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de conductores de costos que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.*

Modelo III. *El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costos en cada caso (análisis, diseño, etc.) del proceso de ingeniería de Software.*

1.6.3 Herramientas Automáticas De Estimación.

Las herramientas automáticas de estimación permiten al planificador estimar costos y esfuerzos, así como llevar a cabo análisis del tipo, que pasa si, con importantes variables del proyecto, tales como la fecha de entrega o la selección del personal. Aunque existen muchas herramientas automáticas de estimación, todas exhiben las mismas características generales y todas requieren de una o más clases de datos.

A partir de estos datos, el modelo implementado por la herramienta automática de estimación proporciona estimaciones del esfuerzo requerido para llevar a cabo el proyecto, los costos, la carga de personal, la duración, y en algunos casos la planificación temporal de desarrollo y riesgos asociados.

En resumen el planificador del Proyecto de Software tiene que estimar tres cosas antes de que comience el proyecto: cuanto durara, cuanto esfuerzo requerirá y cuanta gente estará implicada. Además el planificador debe predecir los recursos de hardware y software que va a requerir y el riesgo implicado.

Para obtener estimaciones exactas para un proyecto, generalmente se utilizan al menos dos de las tres técnicas referidas anteriormente. Mediante la comparación y la conciliación de las estimaciones obtenidas con las diferentes técnicas, el planificador puede obtener una estimación más exacta. La estimación del proyecto de software nunca será una ciencia exacta, pero la combinación de buenos datos históricos y técnicas puede mejorar la precisión de la estimación.

TEMA II. Análisis de Sistemas de Computación.

DESARROLLO.

2.1 Conceptos y Análisis:

Es un conjunto o disposición de procedimientos o programas relacionados de manera que juntos forman una sola unidad. Un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada mostrando un plan lógico en la unión de las partes. Un método, plan o procedimiento de clasificación para hacer algo. También es un conjunto o arreglo de elementos para realizar un objetivo predefinido en el procesamiento de la Información. Esto se lleva a cabo teniendo en cuenta ciertos principios:

- *Debe presentarse y entenderse el dominio de la información de un problema.*
- *Defina las funciones que debe realizar el Software.*
- *Represente el comportamiento del software a consecuencias de acontecimientos externos.*
- *Divida en forma jerárquica los modelos que representan la información, funciones y comportamiento.*

El proceso debe partir desde la información esencial hasta el detalle de la Implementación.

La función del Análisis puede ser dar soporte a las actividades de un negocio, o desarrollar un producto que pueda venderse para generar beneficios. Para conseguir este objetivo, un Sistema basado en computadoras hace uso de seis (6) elementos fundamentales:

Software, que son Programas de computadora, con estructuras de datos y su documentación que hacen efectiva la logística metodología o controles de requerimientos del Programa.

- *Hardware, dispositivos electrónicos y electromecánicos, que proporcionan capacidad de cálculos y funciones rápidas, exactas y efectivas (Computadoras, Censores, maquinarias, bombas, lectores, etc.), que proporcionan una función externa dentro de los Sistemas.*
- *Personal, son los operadores o usuarios directos de las herramientas del Sistema.*
- *Base de Datos, una gran colección de informaciones organizadas y enlazadas al Sistema a las que se accede por medio del Software.*
- *Documentación, Manuales, formularios, y otra información descriptiva que detalla o da instrucciones sobre el empleo y operación del Programa.*
- *Procedimientos, o pasos que definen el uso especifico de cada uno de los elementos o componentes del Sistema y las reglas de su manejo y mantenimiento.*

Un Análisis de Sistema se lleva a cabo teniendo en cuenta los siguientes objetivos en mente:

- *Identifique las necesidades del Cliente.*
- *Evalúe que conceptos tiene el cliente del sistema para establecer su viabilidad.*
- *Realice un Análisis Técnico y económico.*
- *Asigne funciones al Hardware, Software, personal, base de datos, y otros elementos del Sistema.*

- *Establezca las restricciones de presupuestos y planificación temporal.*

- *Cree una definición del sistema que forme el fundamento de todo el trabajo de Ingeniería.*

2.2 Objetivos del Análisis.

2.2.1 Identificación de Necesidades.

Es el primer paso del análisis del sistema, en este proceso en Analista se reúne con el cliente y/o usuario (un representante institucional, departamental o cliente particular), e identifican las metas globales, se analizan las perspectivas del cliente, sus necesidades y requerimientos, sobre la planificación temporal y presupuestal, líneas de mercadeo y otros puntos que puedan ayudar a la identificación y desarrollo del proyecto.

Algunos autores suelen llamar a esta parte ***¨Análisis de Requisitos¨*** y lo dividen en cinco partes:

- *Reconocimiento del problema.*

- *Evaluación y Síntesis.*

- *Modelado.*

- *Especificación.*

- *Revisión.*

Antes de su reunión con el analista, el cliente prepara un documento conceptual del proyecto, aunque es recomendable que este se elabore durante la comunicación Cliente – analista, ya que de hacerlo el cliente solo de todas maneras tendría que ser modificado, durante la identificación de las necesidades.

2.2.2 Estudio de Viabilidad.

Muchas veces cuando se emprende el desarrollo de un proyecto de Sistemas los recursos y el tiempo no son realistas para su materialización sin tener perdidas económicas y frustración profesional. La viabilidad y el análisis de riesgos están relacionados de muchas maneras, si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce, sin embargo se deben tomar en cuenta cuatro áreas principales de interés:

Viabilidad económica.

Viabilidad Técnica.

Viabilidad Legal.

Es determinar cualquier posibilidad de infracción, violación o responsabilidad legal en que se podría incurrir al desarrollar el Sistema.

Alternativas. Una evaluación de los enfoques alternativos del desarrollo del producto o Sistema.

El estudio de la viabilidad puede documentarse como un informe aparte para la alta gerencia.

2.2.3 Análisis Económico y Técnico.

El análisis económico incluye lo que llamamos, el análisis de costos – beneficios, significa una valoración de la inversión económica comparado con los beneficios que se obtendrán en la comercialización y utilidad del producto o sistema.

Muchas veces en el desarrollo de Sistemas de Computación estos son intangibles y resulta un poco dificultoso evaluarlo, esto varia de acuerdo a la características del Sistema. El análisis de costos – beneficios es una fase muy importante de ella depende la posibilidad de desarrollo del Proyecto.

En el Análisis Técnico, el Analista evalúa los principios técnicos del Sistema y al mismo tiempo recoge información adicional sobre el rendimiento, fiabilidad, características de mantenimiento y productividad.

Los resultados obtenidos del análisis técnico son la base para determinar sobre si continuar o abandonar el proyecto, si hay riesgos de que no funcione, no tenga el rendimiento deseado, o si las piezas no encajan perfectamente unas con otras.

2.2.4 Modelado de la arquitectura del Sistema.

Cuando queremos dar a entender mejor lo que vamos a construir en el caso de edificios, Herramientas, Aviones, Maquinas, se crea un modelo idéntico, pero en menor escala (mas pequeño).

Sin embargo cuando aquello que construiremos es un Software, nuestro modelo debe tomar una forma diferente, deben representar todas las funciones y subfunciones de un Sistema. Los modelos se concentran en lo que debe hacer el sistema no en como lo hace, estos modelos pueden incluir notación gráfica, información y comportamiento del Sistema.

Todos los Sistemas basados en computadoras pueden modelarse como transformación de la información empleando una arquitectura del tipo entrada y salida.

2.2.5 Especificaciones del Sistema.

Es un Documento que sirve como fundamento para la Ingeniería Hardware, software, Base de datos, e ingeniería Humana. Describe la función y rendimiento de un Sistema basado en computadoras y las dificultades que estarán presente durante su desarrollo. Las Especificaciones de los requisitos del software se produce en la terminación de la tarea del análisis.

En Conclusión un proyecto de desarrollo de un Sistema de Información comprende varios componentes o pasos llevados a cabo durante la etapa del análisis, el cual ayuda a traducir las necesidades del cliente en un modelo de Sistema que utiliza uno mas de los componentes: Software, hardware, personas, base de datos, documentación y procedimientos.

TEMA III. DISEÑO DE SISTEMAS DE COMPUTACIÓN.

DESARROLLO.

3.1. Conceptos y principios:

El Diseño de Sistemas se define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un Sistema, con suficientes detalles como para permitir su interpretación y realización física.

La etapa del Diseño del Sistema encierra cuatro etapas:

ð ***El diseño de los datos.***

Trasforma el modelo de dominio de la información, creado durante el análisis, en las estructuras de datos necesarios para implementar el Software.

ð ***El Diseño Arquitectónico.***

Define la relación entre cada uno de los elementos estructurales del programa.

ð ***El Diseño de la Interfaz.***

Describe como se comunica el Software consigo mismo, con los sistemas que operan junto con el y con los operadores y usuarios que lo emplean.

ð ***El Diseño de procedimientos.***

Transforma elementos estructurales de la arquitectura del programa. La importancia del Diseño del Software se puede definir en una sola palabra ***Calidad***, dentro del diseño es donde se fomenta la calidad del Proyecto. El Diseño es la única manera de materializar con precisión los requerimientos del cliente.

El Diseño del Software es un proceso y un modelado a la vez. El proceso de Diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del Sistema a construir. A lo largo del diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas:

El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente.

Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el Software.

El Diseño debe proporcionar una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la Implementación.

Para evaluar la calidad de una presentación del diseño, se deben establecer criterios técnicos para un buen diseño como son:

- *Un diseño debe presentar una organización jerárquica que haga un uso inteligente del control entre los componentes del software.*
- *El diseño debe ser modular, es decir, se debe hacer una partición lógica del Software en elementos que realicen funciones y subfunciones especificas.*
- *Un diseño debe contener abstracciones de datos y procedimientos.*
- *Debe producir módulos que presenten características de funcionamiento independiente.*
- *Debe conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y el entorno exterior.*
- *Debe producir un diseño usando un método que pudiera repetirse según la información obtenida*

durante el análisis de requisitos de Software.

Estos criterios no se consiguen por casualidad. El proceso de Diseño del Software exige buena calidad a través de la aplicación de principios fundamentales de Diseño, Metodología sistemática y una revisión exhaustiva.

Cuando se va a diseñar un Sistema de Computadoras se debe tener presente que el proceso de un diseño incluye, concebir y planear algo en la mente, así como hacer un dibujo o modelo o croquis.

3.2. Diseño de la Salida.

En este caso salida se refiere a los resultados e informaciones generadas por el Sistema, Para la mayoría de los usuarios la salida es la única razón para el desarrollo de un Sistema y la base de evaluación de su utilidad. Sin embargo cuando se realiza un sistema, como analistas deben realizar lo siguiente:

- *Determine que información presentar. Decidir si la información será presentada en forma visual, verbal o impresora y seleccionar el medio de salida.*
- *Disponga la presentación de la información en un formato aceptable.*
- *Decida como distribuir la salida entre los posibles destinatarios.*

3.3. Diseño de Archivos.

Incluye decisiones con respecto a la naturaleza y contenido del propio archivo, como si se fuera a emplear para guardar detalles de las transacciones, datos históricos, o información de referencia. Entre las decisiones que se toman durante el diseño de archivos, se encuentran las siguientes:

- *Los datos que deben incluirse en el formato de registros contenidos en el archivo.*
- *La longitud de cada registro, con base en las características de los datos que contenga.*
- *La secuencia a disposición de los registros dentro del archivo (La estructura de almacenamiento que puede ser secuencial, indexada o relativa).*

No todos los sistemas requieren del diseño de todos los archivos, ya que la mayoría de ellos pueden utilizar los del viejo Sistema y solo tenga que enlazarse el nuevo Sistema al Archivo maestro donde se encuentran los registros.

3.4. Diseño de Interacciones con la Base de Datos.

La mayoría de los sistemas de información ya sean implantado en sistemas de cómputos grandes o pequeños, utilizan una base de datos que pueden abarcar varias aplicaciones, por esta razón estos sistemas utilizan u administrador de base de datos, en este caso el diseñador no construye la base de datos sino que consulta a su administrador para ponerse de acuerdo en el uso de esta en el sistema.

3.5 Herramientas para el Diseño de Sistemas.

Apoyan el proceso de formular las características que el sistema debe tener para satisfacer los requerimientos detectados durante las actividades del análisis:

3.5.1 Herramientas de especificación.

Apoyan el proceso de formular las características que debe tener una aplicación, tales como entradas, Salidas, procesamiento y especificaciones de control. Muchas incluyen herramientas para crear especificaciones de datos.

3.5.2 Herramientas para presentación.

Se utilizan para describir la posición de datos, mensajes y encabezados sobre las pantallas de las terminales, reportes y otros medios de entrada y salida.

3.5.3 Herramientas para el desarrollo de Sistemas.

Estas herramientas nos ayudan como analistas a trasladar diseños en aplicaciones funcionales.

3.5.4 Herramientas para Ingeniería de Software.

Apoyan el Proceso de formular diseños de Software, incluyendo procedimientos y controles, así como la documentación correspondiente.

3.5.5 Generadores de códigos.

Producen el código fuente y las aplicaciones a partir de especificaciones funcionales bien articuladas.

3.5.6 Herramientas para pruebas.

Apoyan la fase de la evaluación de un Sistema o de partes del mismo contra las especificaciones. Incluyen facilidades para examinar la correcta operación del Sistema así como el grado de perfección alcanzado en comparación con las expectativas.

La revolución del procesamiento de datos de manera computarizada, junto con las practicas de Diseño sofisticadas están cambiando de forma dramática la manera en que se trasladan las especificaciones de Diseño d Sistemas de Información funcionales.

En Conclusiones Generales. *En una organización o Empresa, el análisis y Diseño de Sistemas, es el proceso de estudiar su Situación con la finalidad de observar como trabaja y decidir si es necesario realizar una mejora; el encargado de llevar a cabo estas tareas es el analista de sistemas.*

Antes de comenzar con el desarrollo de cualquier proyecto, se conduce un estudio de Sistemas para detectar todos los detalles de la situación actual de la empresa. La información reunida con este estudio sirve como base para crear varias estrategias de Diseño. Los administradores deciden que estrategias seguir. Los Gerentes, empleados y otros usuarios finales que se familiarizan cada vez mas con el uso de computadoras están teniendo un papel muy importante en el desarrollo de sistemas.

Todas las organizaciones son Sistemas que actúan de manera reciproca con su medio ambiente recibiendo entradas y produciendo salidas. Los Sistemas que pueden estar formados por otros Sistemas de denominan Sub−sistemas y funcionan para alcanzar los fines de su Implantación.

TEMA IV. IMPLANTACION, EVALUACION Y PRUEBAS.

DESARROLLO.

4.1. IMPLANTACION. Concepto y Definición.

Es la ultima fase del desarrollo de Sistemas. Es el proceso instalar equipos o Software nuevo, como resultado de un análisis y diseño previo como resultado de la sustitución o mejoramiento de la forma de llevar a cavo un proceso automatizado.

Al Implantar un Sistema de Información lo primero que debemos hacer es asegurarnos que el Sistema sea operacional o sea que funcione de acuerdo a los requerimientos del análisis y permitir que los usuarios puedan operarlo.

Existen varios enfoques de Implementación:

- *Es darle responsabilidad a los grupos*.
- *Uso de diferentes estrategias para el entrenamiento de los usuarios*.
- *El Analista de Sistemas necesita ponderar la situación y proponer un plan de conversión que sea adecuado para la organización*.
- *El Analista necesita formular medidas de desempeño con las cuales evaluar a los Usuarios*.
- *Debe Convertir físicamente el sistema de información antiguo, al nuevo modificado*.

En la preparación de la Implantación, aunque el Sistema este bien diseñado y desarrollado correctamente su éxito dependerá de su implantación y ejecución por lo que es importante capacitar al usuario con respecto a su uso y mantenimiento.

4.2. Capacitación de Usuarios del Sistema:

Es enseñar a los usuarios que se relacionan u operan en un proceso de implantación.

La Responsabilidad de esta capacitación de los Usuarios primarios y secundarios es del Analista, desde el personal de captura de datos hasta aquellos que toman las decisiones sin usar una Computadora.

No se debe incluir a personas de diferentes niveles de habilidad e intereses de trabajo; debido a que si en una Empresa existen trabajadores inexpertos no se pueden incluir en la misma sección de los expertos ya que ambos grupos quedaran perdidos.

"Es como querer conducir dos Barcos con diferentes destinos con un mismo Mapa de rutas o con el mismo timón".

Aun y cuando la Empresa puede contratar los Servicios de Instructores externos, el analista es la persona que puede ofrecer la mejor capacitación debido a que conoce el personal y al Sistema mejor que cualquier otro. A la falta o imposibilidad del analista la organización puede contratar otros servicios de capacitación como son:

- *Vendedores: Son aquellos que proporcionan capacitación gratuita fuera de la Empresa de uno o dos días*.
- *Instructor pagado externamente: Son aquellos que pueden enseñar todo acerca de las computadoras pero para algunos usuarios esta no es una capacitación necesaria*.
- *Instructores en casa: Están familiarizados con el personal y pueden adecuar los materiales a sus necesidades, pero le faltaría experiencia en Sistemas de Información que es realmente la necesidad del usuario*.

En nuestro país existe una ley institucional (Ley 116 del 16 de Enero de 1980) creado durante el gobierno del Presidente Antonio Guzmán Fernández llamada INFOTEP, representante de los trabajadores y empresarios en el ámbito de Capacitación y entrenamiento, la cual Asesora y brinda Sus servicios a las Empresas y Sus trabajadores.

4.3.1 Objetivos de la Capacitación:

Es lograr que los usuarios tengan el Dominio necesario de las cosas básicas acerca de las maquinarias y procesos que se emplean para su operación de manera eficiente y segura.

4.4. La Evaluación del Sistema:

Se lleva a cabo para identificar puntos débiles y fuertes del Sistema implantado. La evaluación ocurre a lo largo de cualquiera de las siguientes cuatro dimensiones:

4.4.1 Evaluación operacional:

Es el Momento en que sé evalúa la manera en que funciona el Sistema, esto incluye su facilidad de uso, Tiempo de respuesta ante una necesidad o proceso, como se adecuan los formatos en que se presenta la Información, contabilidad global y su nivel de Utilidad.

4.4.2 Impacto Organizacional:

Identifica y mide los beneficios operacionales para la Empresa en áreas tales como, Finanzas (Costos, Ingresos y Ganancias), eficiencia en el desempeño laboral e impacto competitivo, Impacto, rapidez y organización en el flujo de Información interna y externa.

4.4.3 Desempeño del Desarrollo.

Es la evaluación del Proceso de desarrollo adecuado tomando en cuentas ciertos criterios como, Tiempo y esfuerzo en el desarrollo concuerden con presupuesto y estándares y otros criterios de Administración de Proyectos. Además se incluyen la valoración de los métodos y herramientas utilizados durante el desarrollo del Sistema.

4.5. Prueba de Sistemas.

Dependiendo del tamaño de la Empresa que usara el Sistema y el riesgo asociado a su uso, puede hacerse la elección de comenzar la operación del Sistema solo en un área de la Empresa (como una Prueba piloto), que puede llevarse a cabo en un Departamento o con una o dos personas. Cuando se implanta un nuevo sistema lo aconsejable es que el viejo y el nuevo funcionen de manera simultanea o paralela con la finalidad de comparar los resultados que ambos ofrecen en su operación, además dar tiempo al personal para su entrenamiento y adaptación al nuevo Sistema.

Durante el Proceso de Implantación y Prueba se deben implementar todas las estrategias posibles para garantizar que en el uso inicial del Sistema este se encuentre libre de problemas lo cual se puede descubrir durante este proceso y levar a cabo las correcciones de lugar para su buen funcionamiento.

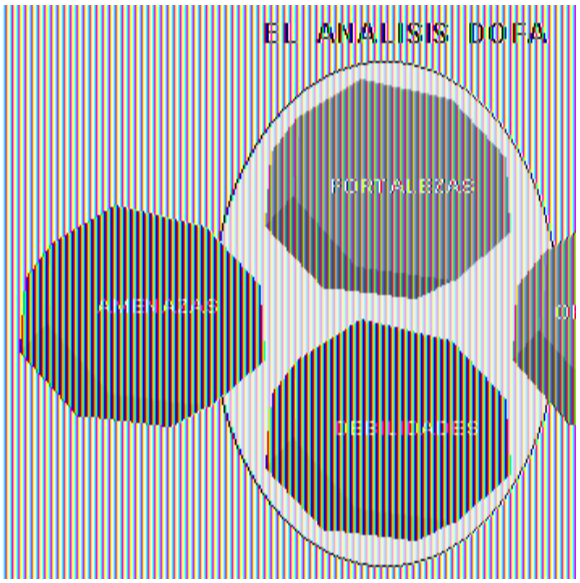
Desdichadamente la evaluación de Sistemas no siempre recibe la atención que merece, sin embargo cuando se lleva a cabo de manera adecuada proporciona muchas informaciones que pueden ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones futuras

El Análisis FODA:

FODA (en inglés *SWOT*), es la sigla usada para referirse a una herramienta analítica que le permitirá trabajar con toda la información que posea sobre su negocio, útil para examinar sus Fortalezas, Oportunidades, Debilidades y Amenazas.

Este tipo de análisis representa un esfuerzo para examinar la interacción entre las características particulares de su negocio y el entorno en el cual éste compite. El análisis FODA tiene múltiples aplicaciones y puede ser usado por todos los niveles de la corporación y en diferentes unidades de análisis tales como producto, mercado, producto–mercado, línea de productos, corporación, empresa, división, unidad estratégica de negocios, etc). Muchas de las conclusiones obtenidas como resultado del análisis FODA, podrán serle de gran utilidad en el análisis del mercado y en las estrategias de mercadeo que diseñé y que califiquen para ser incorporadas en el plan de negocios.

El análisis FODA debe enfocarse solamente hacia los factores claves para el éxito de su negocio. Debe resaltar las fortalezas y las debilidades diferenciales internas al compararlo de manera objetiva y realista con la competencia y con las oportunidades y amenazas claves del entorno.



Lo anterior significa que el análisis FODA consta de dos partes: una interna y otra externa.

- la parte interna tiene que ver con las fortalezas y las debilidades de su negocio, aspectos sobre los cuales usted tiene algún grado de control.
- la parte externa mira las oportunidades que ofrece el mercado y las amenazas que debe enfrentar su negocio en el mercado seleccionado. Aqui usted tiene que desarrollar toda su capacidad y habilidad para aprovechar esas oportunidades y para minimizar o anular esas amenazas, circunstancias sobre las cuales usted tiene poco o ningún control directo.

Fortalezas y Debilidades
Considere áreas como las siguientes:

- **Análisis de Recursos**
  Capital, recursos humanos, sistemas de información, activos fijos, activos no tangibles.
- **Análisis de Actividades**
  Recursos gerenciales, recursos estratégicos, creatividad
- **Análisis de Riesgos**
  Con relación a los recursos y a las actividades de la empresa.

- **Análisis de Portafolio**
  La contribución consolidada de las diferentes actividades de la organización.

Hágase preguntas como éstas:

- ¿Cuáles son aquellos cinco a siete aspectos donde usted cree que supera a sus principales competidores?
- ¿Cuáles son aquellos cinco a siete aspectos donde usted cree que sus competidores lo superan?

Al evaluar las fortalezas de una organización, tenga en cuenta que éstas se pueden clasificar así:

- **Fortalezas Organizacionales Comunes**
  Cuando una determinada fortaleza es poseida por un gran número de empresas competidoras. La paridad competitiva se da cuando un gran número de empresas competidoras están en capacidad de implementar la misma estrategia.
- **Fortalezas Distintivas**
  Cuando una determinada fortaleza es poseída solamente por un reducido número de empresas competidoras. Las empresas que saben explotar su fortaleza distintiva, generalmente logran una ventaja competitiva y obtienen utilidades económicas por encima del promedio de su industria. Las fortalezas distintivas podrían no ser imitables cuando:

  - Su adquisición o desarrollo pueden depender de una circunstancia histórica única que otras empresas no pueden copiar.
  - Su naturaleza y carácter podría no ser conocido o comprendido por las empresas competidoras. (Se basa en sistemas sociales complejos como la cultura empresarial o el trabajo en equipo).
- **Fortalezas de Imitación de las Fortalezas Distintivas**
  Es la capacidad de copiar la fortaleza distintiva de otra empresa y de convertirla en una estrategia que genere utilidad económica.

  La ventaja competitiva será temporalmente sostenible, cuando subsiste después que cesan todos los intentos de imitación estratégica por parte de la competencia.

  Al evaluar las debilidades de la organización, tenga en cuenta que se está refiriendo a aquellas que le impiden a la empresa seleccionar e implementar estrategias que le permitan desarrollar su misión. Una empresa tiene una desventaja competitiva cuando no está implementando estrategias que generen valor mientras otras firmas competidoras si lo están haciendo.

Oportunidades y Amenazas
Las oportunidades organizacionales se encuentran en aquellas áreas que podrían generar muy altos desempeños. Las amenazas organizacionales están en aquellas áreas donde la empresa encuentra dificultad para alcanzar altos niveles de desempeño.

Considere:

- ♦ **Análisis del Entorno**
  Estructura de su industria (Proveedores, canales de distribución, clientes, mercados, competidores).
- ♦ **Grupos de interés**
  Gobierno, instituciones públicas, sindicatos, gremios, accionistas, comunidad.
- ♦ **El entorno visto en forma más amplia**
  Aspectos demográficos, políticos, legislativos, etc.