



UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ACATLÁN

METODOLOGÍA Y HERRAMIENTAS DE LA INGENIERÍA DE SOFTWARE. UN CASO PRÁCTICO

TESINA

QUE PARA OBTENER EL TÍTULO DE

MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA

FERNANDO HERNÁNDEZ MARTÍNEZ

ASESOR: Mtra. Georgina Eslava García

Octubre de 2010



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Para las personas creyentes, Dios esta al principio. Para los científicos está al final de todas sus reflexiones.

Max Planck (1858-1947) *Físico alemán.*

AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México, mi alma Mater.

A la Mtra. Georgina Eslava asesora de tesis y amiga.

A todos mis profesores.

DEDICATORIAS

Dedico este trabajo con todo mi cariño, respeto y admiración al pilar de lo que ha sido mi familia. A mi preciosa Mamá Lupe.

A mis mamás Lupita y Conchita porque siempre están presentes en mi vida.

A mis grandes amores:

Coco, mi linda esposa con quién comparto sueños y logros.

Fer, recuerda que todo lo podemos, pero hay que concentrarnos en lo que queremos.

Daniel, quién le dio un nuevo propósito a mi vida.

A Paula y Jimena

A Marcoantonio, Guillermo, Ramón y Ernesto

Al Sr. Mauro y la Sra. Dolores y a toda la familia Martínez José

Finalmente a mi padre, Fernando Hernández Ruíz (†).

Introducción

Propósito de este texto. Objetivo general

El objetivo del presente trabajo es brindar la metodología y las herramientas para producir sistemas de información de cómputo que cumpla con un propósito específico, en diversas modalidades y formas de distribución. De esta manera el ingeniero de software utilizará el paradigma adecuado para desarrollar, evaluar e implementar el sistema de cómputo que le servirá como fundamento teórico.

A quien va dirigido

Estudiantes de la Licenciatura en Matemáticas Aplicadas y Computación de quinto semestre e interesados en el desarrollo de software, como ingenieros de desarrollo de sistemas, administradores o líderes de proyectos de software

Justificación

Actualmente, el software ha superado al hardware como la clave del éxito de muchos sistemas basados en computadoras. Tanto que el uso de la computadora permite llevar un negocio, controlar un producto o capacitar un sistema.

Así, el lector podrá seguir cada uno de los pasos y procesos involucrados en el desarrollo de la documentación del Sistema de información, así como en el desarrollo del Software. Es decir, a partir del Capítulo 2. Procesos y Administración de Software, Tema 2.3 Planeación y calendarización de proyectos, se hará referencia a un CD, el cual contiene un CASO PRACTICO, que le permitirá al lector llevar a cabo la metodología en conjunto con la PRACTICA, ya que se indican mediante notas al pie, la página y lo que se espera ver de acuerdo al tema tratado en esta metodología con el fin de hacer más claro y concreto la comprensión de la misma.

Índice

Unidad 1. Introducción a la Ingeniería de Software	1
1.1 Visión general	1
1.2 Introducción a la Ingeniería de Software	2
1.3 Problemas relacionados con el software	15
1.4 Procesos de producción de software	20
Unidad 2. Procesos y administración de software	26
2.1 Visión general	26
2.2 Administración de Software	27
2.3 Planeación y calendarización de proyectos	31
2.4 Modelos de capacidad de madurez del proceso (CMM)	41
Unidad 3. Ingeniería de requerimientos	45
3.1 Visión general	45
3.2 Análisis de problemas: objetivo del análisis y técnicas para obtener información	46
3.3 Ingeniería de requerimientos: tipos de requerimientos y notaciones de especificación	51
3.4 Inicio del proceso	57
3.5 Análisis de factibilidad	63
Unidad 4. Modelado y especificación formal	70
4.1 Visión general	70
4.2 Modelos de contexto y comportamiento	71
4.3 Modelos de datos	79
4.4 La especificación formal en el proceso de software	83
Unidad 5. Diseño de software	85
5.1 Visión general	85
5.2 Diagramación del diseño	86
5.3 Importancia de una interfaz en el desarrollo de aplicaciones	91
5.4 Criterios ergonómicos en las aplicaciones	103
Unidad 6. Verificación, validación y pruebas del software	108
6.1 Visión general	108
6.3 Pruebas del sistema	109
6.4 Métodos de prueba	115

6.5 Verificación formal	123
Unidad 7. Cambios en el software y mejora de procesos	128
7.1 Visión general	128
7.2 Mantenimiento de software	129
7.3 Calidad del proceso y del producto	133
7.4 Modelo de mejora de procesos	136
7.5 Modelo de procesos para la industria de software (MoProSoft)	140
Conclusiones	143
Bibliografía	144

Unidad 1. Introducción a la Ingeniería de Software

Tema 1.1 Visión general

Objetivo Particular

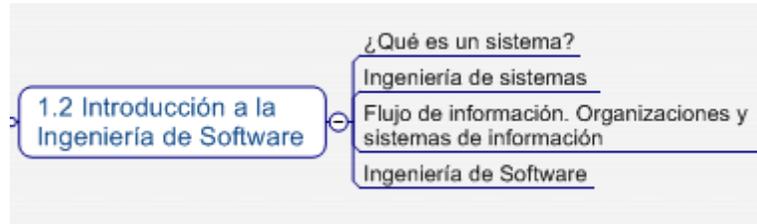
Al término de la unidad el alumno reconocerá la importancia del rol que juega la Ingeniería de software y su metodología en la administración de información, mediante el planteamiento de diversos problemas relacionados con el desarrollo de software y los diversos modelos de procesos de producción de software.

Mapa conceptual



Tema 1.2 Introducción a la Ingeniería de Software

Mapa conceptual



Objetivo específico

Al final del tema, el alumno identificará los diversos componentes y el flujo de información de un sistema de información, mediante diversos esquemas de comportamiento y algunas definiciones.

¿Qué es un sistema?

La palabra sistema, como es usada en análisis de sistemas de información, no necesariamente significa una computadora o un grupo de programas de computadora.

“Un sistema puede ser definido como una serie de elementos interrelacionados que realizan alguna actividad, función u operación. También puede ser definido como un conjunto de componentes que interactúan para alcanzar un objetivo”

La Teoría General de Sistemas (TGS) surgió con los trabajos del alemán Ludwig von Bertalanffy, publicados entre 1950 y 1968. La TGS no busca solucionar problemas o intentar soluciones prácticas, pero sí producir teorías y formulaciones conceptuales que pueden crear condiciones de aplicación en la realidad.

La TGS se fundamenta en tres premisas básicas:

- Los sistemas existen dentro de sistemas, es decir, cada sistema existe dentro de otro más grande.
- De acuerdo al punto anterior, los sistemas son abiertos. Cada sistema que se examine, excepto el menor o mayor, recibe y descarga algo en los otros sistemas, generalmente en los contiguos. Los sistemas abiertos se caracterizan por un proceso de cambio infinito con su entorno, que son los otros sistemas. Cuando el intercambio cesa, el sistema se desintegra, esto es, pierde sus fuentes de energía.
- Las funciones de un sistema dependen de su estructura, es decir, para los sistemas biológicos y mecánicos esta afirmación es intuitiva. Los tejidos musculares por ejemplo, se contraen porque están constituidos por una estructura celular que permite contracciones.

Existen diversas clases de sistemas, por ejemplo: biológicos, educacionales, sociales, administrativos, etc. En cada uno de ellos, siempre se podrá identificar su medio ambiente, sus entradas, procesos y salidas, así como la información utilizada para la retroalimentación, como se ilustra en la figura 1.2.1

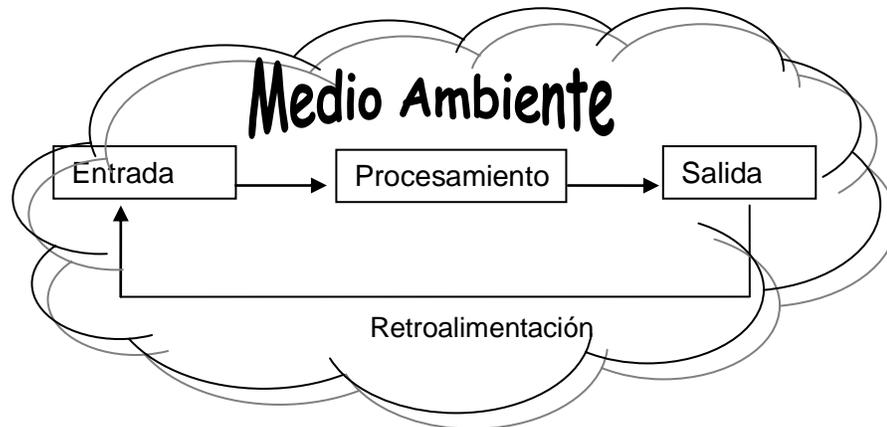


Fig. 1.2.1 Componentes de un sistema

- **Entrada.** Recopilación y captura de datos. Puede adoptar muchas formas: lector de código de barras (tarjetas de crédito, registro de productos, etc.), respuesta de clientes generados por una encuesta, lectoras de óvalos (actas de calificaciones), etc. Independientemente del método de entrada, la exactitud de la misma es decisiva para obtener la salida.
- **Procesamiento.** Conversión o transformación de datos en salidas útiles, lo cual puede implicar ejecutar cálculos, realizar comparaciones y adoptar acciones alternas y el almacenamiento de datos para su uso posterior.
- **Salida.** Información útil por lo general bajo la modalidad de documento y/o consulta, cheques de pago, ticket de compra.
- **Retroalimentación.** Es la salida que sirve para hacer cambios en actividades de entrada o procesamiento, por ejemplo en el cálculo de nómina si se capturó mal el número de horas trabajadas, 400 en lugar de 40, el sistema debe ser congruente con escalas predeterminadas. Es decir, es improbable que una persona labore más de 100 horas a la semana lo cual la retroalimentación será un mensaje de error indicando que la captura está mal.

El interés de la Teoría General de Sistema, son las características y parámetros que establece para todos los sistemas. Aplicada a la administración la TGS, la empresa se ve como una estructura que se reproduce y se visualiza a través de un sistema de toma de decisiones, tanto individual como colectivamente.

Las teorías tradicionales han visto la organización humana como un sistema cerrado. Eso ha llevado a no tener en **cuenta el ambiente**, provocando poco desarrollo y comprensión de la retroalimentación (feedback), básica para sobrevivir.

En algunos textos de administración de negocios o sistemas de información se pueden encontrar algunas clasificaciones de acuerdo al propósito o necesidad de la empresa, como se muestra en el Cuadro 1.2.1

Cuadro 1.2.1 Clasificación de sistemas

Nivel	Tipo de sistema	Propósito
Operativo	TPS. Sistema de procesamiento de transacciones	Son importantes para las operaciones cotidianas de un negocio.
	OAS. Sistema de automatización de oficina	Ayudan a manipular, transformar o compartir la información con otros departamentos (Star Office, Microsoft Office, SmartSuite, etc.)
	KWS. Sistema de trabajo de conocimiento	Sirven de apoyo a profesionales: científicos, ingenieros, médicos, para crear nuevo conocimiento.
Alto	DSS. Sistema de apoyo a decisiones	Depende de una base de datos y se ajustan al gusto del grupo o persona que toma decisiones.(PROMIS: Decisiones médicas)
	MIS. Sistema de información gerencial	Producen información que se emplea en la toma de decisiones, contribuyen a unificar algunas de las funciones de información a través de informes o reportes.
Estratégico	ESS. Sistema de apoyo a ejecutivos	Ayudan a organizar actividades para comprender mejor su entorno externo mediante herramientas gráficas y de comunicaciones.
	GDSS. Sistema de apoyo a decisiones de grupo	Son sistemas que reúnen las opiniones de un grupo de personas. Se diseñan para minimizar conductas negativas o toma de decisiones conformistas y mas objetivas.

En otras ocasiones el sistema es clasificado por el tipo de información que administrará y la relación de esta con otros sistemas o subsistemas de la empresa o diferentes empresas, como se muestra en el Cuadro 1.2.2

Cuadro 1.2.2 Otra clasificación de sistemas de información

Tipo de sistema	Descripción
Simple.	Pocos componentes y su relación o interacción es sencilla y directo.
Compleja.	Muchos componentes estrechamente relacionados y conectados.
Abierto.	Presentan intercambio con el ambiente, a través de entradas y salidas. Intercambian energía y materia con el ambiente. Son adaptativos para sobrevivir. Su estructura es óptima cuando el conjunto de elementos del sistema se organiza,

aproximándose a una operación adaptativa. La adaptabilidad es un continuo proceso de aprendizaje y de auto-organización. Interactúa con su entorno, por ejemplo la legislación y normativas, barreras no arancelarias, el medio ambiente y la evolución de su ciudad, la cultura, política, sociología.

Cerrado. Son sistemas cuyo comportamiento es determinístico y programado y opera con muy pequeño intercambio de energía y materia con el ambiente. Se aplica el término a los sistemas completamente estructurados, donde los elementos y relaciones se combinan de una manera peculiar y rígida produciendo una salida invariable, como las máquinas.

Estable. Sufre escasos cambios conforme el paso del tiempo.

Dinámico. Sufre rápidos y constantes cambios al paso del tiempo.

Adaptable. Es capaz de recibir cambios de acuerdo a su entorno.

No adaptable. Es imposible hacer cambios.

Permanente. Diseñado para existir durante un periodo relativamente largo.

Temporal. Diseñado para existir durante un periodo relativamente corto.

Ingeniería de sistemas

Desde las últimas cinco décadas los sistemas basados en computadora han introducido un nuevo orden, aunque la tecnología ha conseguido grandes avances, las palabras de Maquiavelo aún son ciertas:

“No hay nada más difícil de llevar a cabo, más peligroso de realizar o de éxito más incierto que tomar el liderazgo en la introducción de un nuevo orden de cosas”, Maquiavelo

La ingeniería de sistemas es un proceso de modelado a nivel global o detallado. Sus objetivos son:

- Definir procesos
- Representar el comportamiento de los procesos
- Definir explícitamente entradas exógenas y endógenas de la información
- Representar interfases para entender la visión global.

Jerarquía de la Ingeniería de Sistemas

La ingeniería de sistemas nos da otra perspectiva de este concepto, ya que antes de que sea posible construir el software, se debe entender el “sistema” en que éste reside, es decir, visualizar las estructuras de los sistemas y subsistemas en donde residirá el software a desarrollar, figura 1.2.2. Para lograrlo es necesario determinar el objetivo general del sistema, así como el papel del software, hardware, personas, procedimientos y bases de datos.

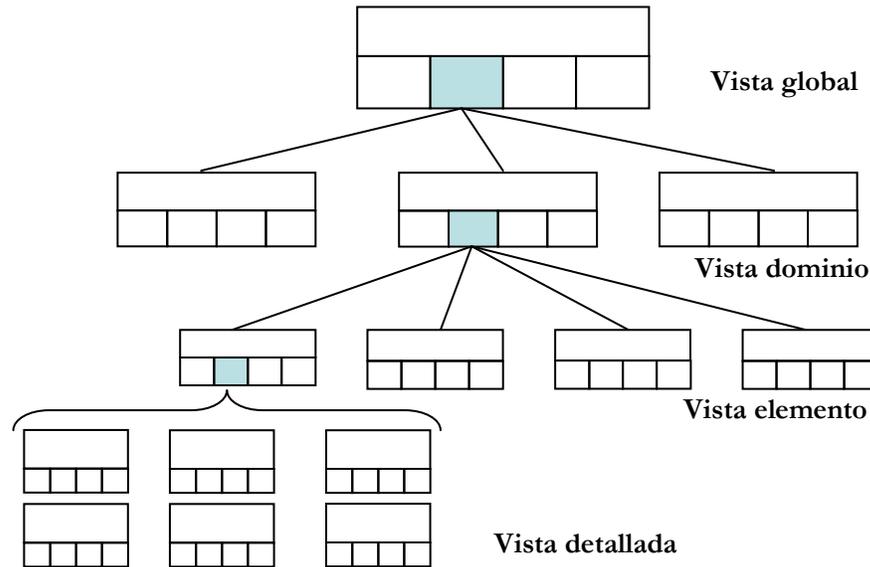


Fig. 1.2.2 Jerarquía de la Ingeniería de Sistemas

De esta manera es importante identificar los diversos niveles jerárquicos del sistema como se señala en la figura 1:

1. Vista global. Se compone de un conjunto de sistemas o dominios
2. Vista dominio. Se compone de elementos específicos los cuales realizan un papel para lograr objetivos y metas de los dominios
3. Vista elemento. Se implementa con componentes técnicos que le dan funcionalidad necesaria a un elemento.
4. Vista Detallada. Elemento: programa, módulo clase u objeto, sentencia del lenguaje de programación

Con frecuencia no se advierte, pero un negocio también es un sistema, sus partes son conocidas como mercadotecnia, producción, ventas, investigación, embarque, contabilidad y personal, como se muestra en la figura 1.2.3. Así los sistemas de información son el lazo que mantiene unidos a diferentes componentes en forma total para trabajar en forma efectiva hacia el mismo objetivo.

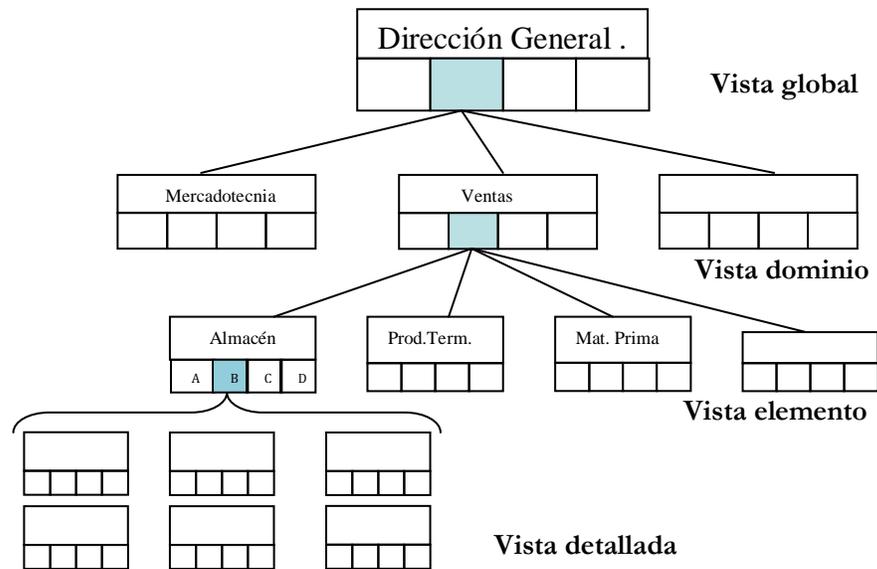


Fig. 1.2.3 Ejemplo de un negocio

Sin embargo, dichos sistemas tienen elementos comunes:

- Interacción con el medio ambiente: entradas, proceso (sistema de información) y salidas
- Tienen un propósito
- Posibilidad de autorregulación (regulación), es decir se mantienen en un estado estable.
- Posibilidad de corrección (control), lo cual se ajusta a las necesidades.

Flujo de información. Organizaciones y sistemas de información

Una organización, es un conjunto formal personas y otros recursos establecidos para cumplir una serie de metas. La meta por lo general es lucrativa: máxima utilidad, mínimo costo, en otros casos, también la meta puede ser no lucrativa: agrupaciones sociales, religiosas, universidades y otros.

Por otra parte, un ingeniero en sistemas realiza actividades interdisciplinarias con grupos de personas con base en conocimientos amplios y requeridos para considerar todas las implicaciones de las decisiones en el diseño de sistemas.

Todas las partes deben involucrarse para definir los requerimientos operacionales para ser identificados, analizados, validados y gestionados, las cuales son actividades propias de la Ingeniería de Software. En la figura 1.2.4, se pueden identificar los componentes principales de un sistema, su medio ambiente y los requerimientos operacionales. Así mismo, se puede apreciar el elemento flujo de valor, el cual da sentido a los objetivos, misión y visión de la empresa u organización, en donde se encuentra implementado el sistema de información o software.

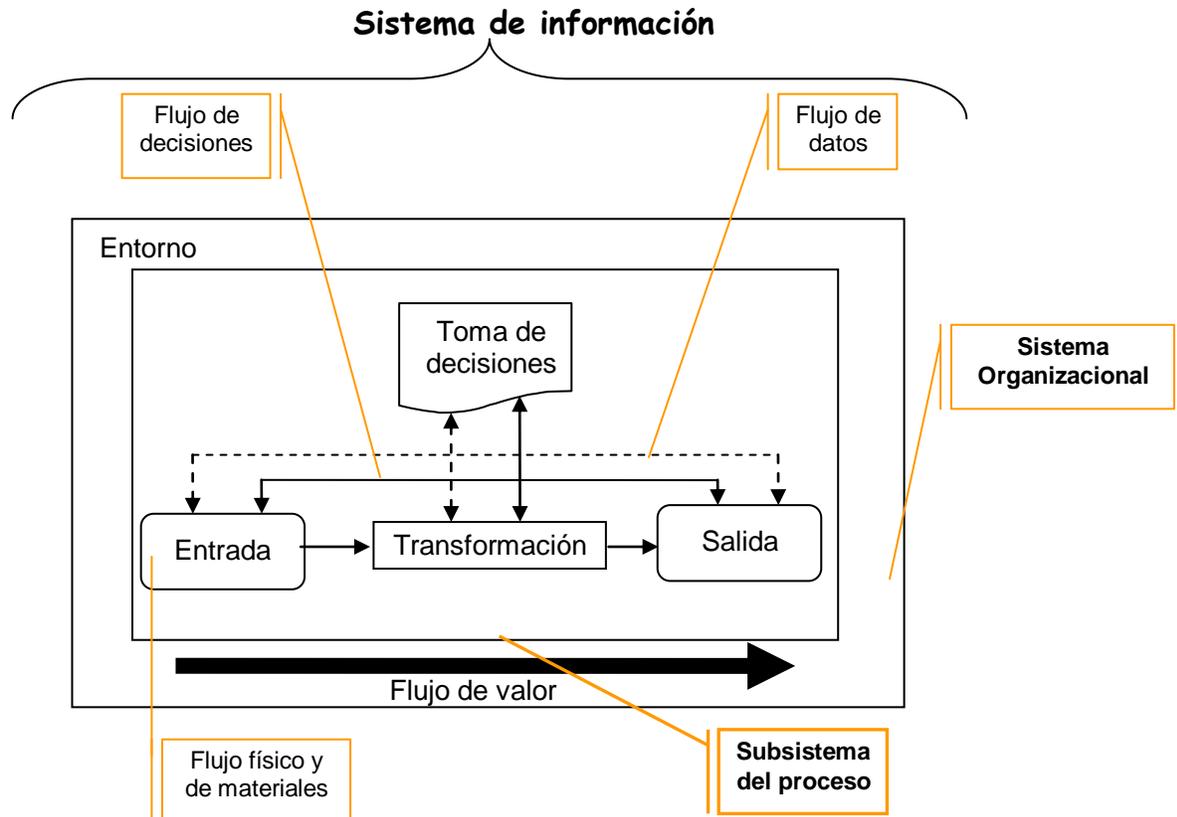


Fig.1.2.4 Modelo general de una organización

Es importante tomar en cuenta que un sistema de información involucra datos, la transformación de éstos en información para definir decisiones que convengan a la empresa. Por tal motivo es importante conocer las características principales de los elementos básicos de un sistema de información:

Información. Es un conjunto de datos organizados de tal modo que adquieren un valor adicional más allá del propio para un usuario de los mismos. Por ejemplo, es más valioso saber las ventas mensuales totales que conocerlas por vendedor individual.

Datos. Son realidades concretas en su estado primario, por ejemplo: nombre de un empleado, número de horas laboradas, número de pedidos levantados. Sus clasificaciones se muestran en el siguiente cuadro

Cuadro 1.2.3 Clasificaciones de datos

Tipo de dato	Descripción
Alfanumérico	Números, letras y otros caracteres
De imágenes	Imágenes, gráficas
Audio	Sonidos o tonos
Video	Imágenes en movimiento

Proceso. Es la conversión de datos en información, lo cual consiste en una serie de tareas relacionadas entre sí y ejecutadas con el fin de producir un resultado definido

Conocimiento. Para definir relaciones entre datos se requiere de conocimiento, es decir de apreciar y comprender un conjunto de información y de la utilidad que puede extraerse de ella en beneficio de una tarea específica.

Base de conocimiento. Conjunto de datos, reglas, procedimientos y relaciones de consideración indispensable para adquirir valor u obtener el resultado apropiado o buscado.

¿Cómo ocurre el incremento o flujo de valor? En el mecanismo de transformación, varios subsistemas contienen procesos que contribuyen a convertir entradas específicas en bienes o servicios de mayor valor. Estos procesos de valor agregado incrementan el valor relativo de las entradas combinadas en su camino, hacia la conversión en salidas finales de una organización, lo que va sin duda relacionado con información valiosa.

Las características propias, que hacen que la información sea valiosa se describen a continuación:

Exacta. Carece de errores, en algunos casos se genera información inexacta porque se insertan datos erróneos en el proceso de transformación, lo que comúnmente se conoce como “si entra basura, sale basura = GIGO Garbage IN Garbage OUT”

Completa. Un informe de inversión que no incluya costos importantes no estaría completo. Debe contener todos los datos importantes

Económica. La producción de la información debe ser relativamente barata, siempre se debe evaluar el valor de la información y el costo de producirla.

Flexible. Es útil para muchos propósitos. Por ejemplo, la información de existencias de un almacén (inventario) para cerrar la venta, a un gerente de producción para determinar el artículo más vendido, y aun ejecutivo de finanzas para conocer el valor total de la inversión en el inventario de la empresa.

Confiable. La confiabilidad depende de la forma en recolectar los datos o de la fuente de información. Un rumor anónimo no es nada confiable.

Pertinente. Es la realmente importante para el tomador de decisiones. Por ejemplo, si la madera decrece su precio no tiene mayor importancia para un fabricante de microprocesadores.

Simple. Un exceso de información puede provocar sobre carga de información o confusiones.

Oportuna. Es cuando se conoce la información justo cuando se solicita o necesita.

Verificable. Tener la posibilidad de comprobar que es correcta, quizá mediante la consulta de muchas fuentes de información al respecto.

Accesible. Debe ser de fácil acceso a usuarios autorizados en un formato adecuado

Segura. Debe estar protegido contra usuarios no autorizados.

Ingeniería de software

La Ingeniería de Software no son sólo programas, es todo aquello que se involucra en el proceso del desarrollo de una sistema de información, tales como los diagramas de flujo manuales de operación, estudios de factibilidad, y demás elementos relacionados con la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta.

En 1970, menos del 1% de las personas podrían haber definido lo que significaba “software de computadora”. En la actualidad, la mayoría de los profesionales y muchos miembros del público creen que entienden el software ¿pero en realidad lo entienden?

Para entender el software y la Ingeniería de Software, es importante examinar las características que lo hacen diferente de otras cosas que construye el ser humano. De esta manera y apoyándonos en la figura 1.2.5, podemos ver que el software es un elemento lógico, en lugar de físico, de un sistema, ya que el software se desarrolla o construye; no se manufactura en el sentido clásico. Por otra parte el software no se desgasta, ya que para el hardware, la tasa de fallas alta representa la “mortalidad infantil”, conforme pasa el tiempo el hardware se estabiliza pero el avance de la tecnología hará en su momento que el hardware se estropea.

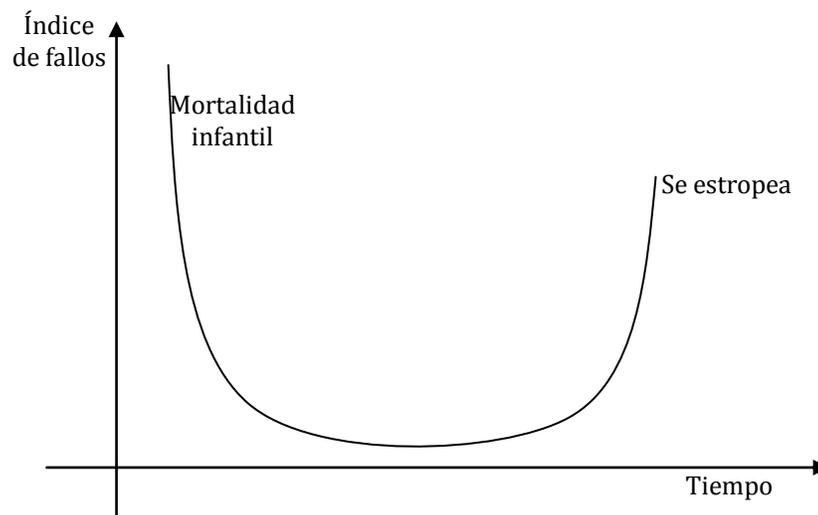


Fig.1.2.5 Curva de fallas para el hardware

La relación, llamada a menudo “curva de la bañera”, indica que el hardware tiene un número considerablemente alto de fallas al inicio de su vida. Estas se atribuyen a defectos de manufactura, después los defectos se corrigen y la tasa de fallas baja hasta un nivel estable, por algún periodo de tiempo. Sin embargo, la tasa de fallas se eleva de nuevo cuando los componentes del hardware sufren los efectos acumulativos del polvo, la vibración, el abuso, temperaturas extremas y otros males ambientales.

De acuerdo a lo anterior, la implicación es clara para el software, es decir, el software no se

desgasta pero se deteriora, esta relación se puede ver claramente en la figura 1.2.6.

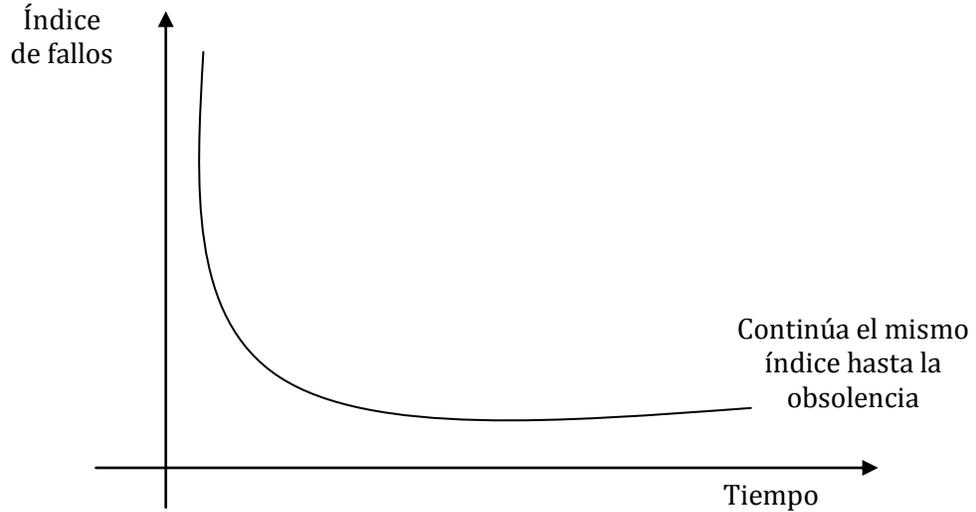


Fig.1.2.6 Curva ideal de fallas para el software

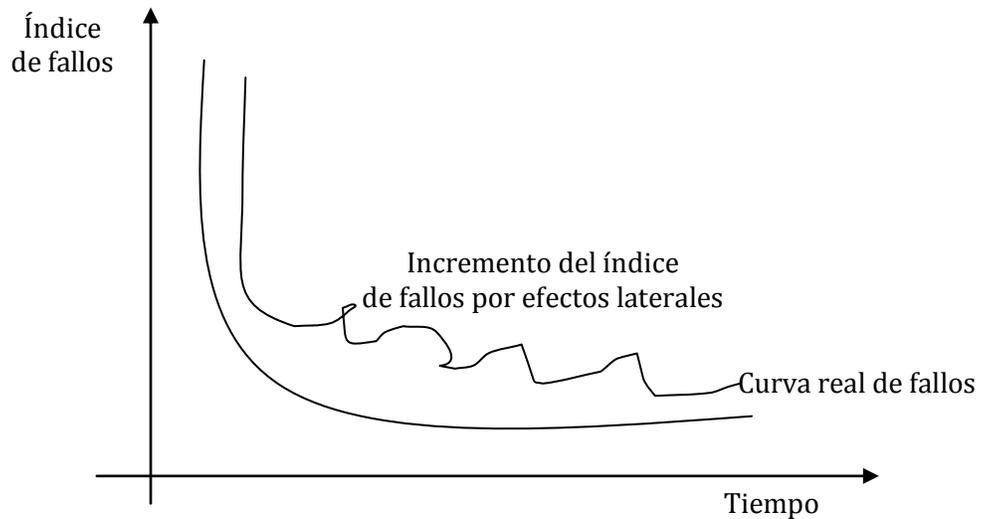


Fig.1.2.7 Curva real de fallas para el software

El software experimenta cambios durante su vida útil, y cuando estos ocurren se presenta la posibilidad de introducir errores, lo que ocasiona que la curva de fallas tenga picos. Antes de que la curva pueda regresar a su estado original con una tasa de fallos estable, se requiere de otro cambio, lo que ocasiona que la curva tenga otro pico. De esta manera, el nivel de fallas mínimo se comienza a elevar, por tal motivo el software se deteriora, como se muestra en la figura 1.2.7..

A pesar de que la industria tiene una tendencia hacia la construcción por componentes, la mayoría del software aún se construye a la medida. Cuando una disciplina de ingeniería evoluciona, se crea una colección de diseños estándar de componentes. Los componentes reutilizables se han creado para que el ingeniero se pueda concentrar en los elementos que en

realidad son innovadores en el diseño; es decir, en las partes que representa algo nuevo. Un componente de software se debe diseñar e implementar de forma que pueda utilizarse en muchos programas diferentes. Los componentes reutilizables modernos encapsulan tanto los datos como el proceso que se aplica a éstos, lo que permite al ingeniero de software crear aplicaciones nuevas a partir de las partes reutilizables.

“Las ideas son los bloques de construcción de ideas”, Jason Zebehazy

Los ingenieros de software se concentran en el desarrollo de productos de software, que se vende a un cliente:

- Productos genéricos. Son sistemas aislados producidos por una organización de desarrollo y que se venden al mercado abierto a cualquier cliente que le sea posible comprarlos, algunas veces éstos sistemas se denominan software empaquetado.
- Productos personalizados. Son sistemas requeridos por un cliente en particular. Un contratista de software desarrolla el software especialmente para ese cliente.

De estas clasificaciones, todavía podemos encontrar otras, de acuerdo a su uso:

- Software de sistemas. El usuario interactúa con el hardware, puede haber múltiples usuarios al mismo tiempo, se dosifican los recursos y su estructura de datos es compleja. Por ejemplo, Compiladores, editores, sistemas operativos (UNIX, MSDOS, Windows, etc.) Administrador de redes (Novell, SUN, etc.)
- Software de tiempo real. Por ejemplo, Registro de vuelos de aviones, miden, analizan y controlan sucesos en el momento que ocurren. Bolsa mexicana de valores, responden típicamente en el rango de un milisegundo al minuto.
- Software de gestión. Permite el control de las entradas, procesamientos y salidas de información, facilitan operaciones comerciales, gestionan las tomas de decisión. Por ejemplo, nómina, contabilidad, inventarios, transacciones en punto de venta.
- Software de ingeniería y científico. Son algoritmos de manejo de números. Por ejemplo, astronomía, vulcanología, análisis de presión de motores, lanzamientos espaciales, biología molecular, fabricación automática.
- Software empotrado. Reside en la memoria del hardware, ejecuta funciones digitales muy limitadas y curiosas. Teclas de microondas, funciones digitales de un automóvil o de juguetes.
- Software de computadoras personales. Tales como procesadores de datos, hojas de cálculo, multimedia, juegos de entretenimiento.
- Software de inteligencia artificial. Involucran el uso de algoritmos no numéricos (lógica), para el desarrollo de sistemas expertos, software de reconocimiento de patrones (imágenes y voz), redes neuronales artificiales (simulación de la estructura de procesos del cerebro).

Definición

Así pues podemos decir que la Ingeniería de software es la rama de la ingeniería que crea y mantiene las aplicaciones de software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos.

En esta definición existen dos fases clave:

- a) Disciplina de la ingeniería. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas.
- b) Todos los aspectos de producción de software. La ingeniería de software no solo comprende los procesos técnicos sino también la administración de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

Algunas otras definiciones de esta rama de la ingeniería nacieron a través del concepto de análisis y diseño de sistemas, en donde este concepto en los años 80's se definía como:

- El análisis y diseño de sistemas se refiere al proceso de examinar una situación de la empresa con la intención de mejorarla mediante nuevos procedimientos y métodos.
- El diseño de sistemas es el proceso de planeación de un nuevo sistema dentro de la empresa para reemplazar o complementar el existente
- El análisis de sistemas es el proceso que sirve para recopilar e interpretar los hechos, diagnosticar problemas y utilizar estos hechos a fin de mejorar el sistema"

Estos conceptos, que en la actualidad se siguen ocupando, permiten dar pie a la conceptualización de la ciencia Ingeniería de Software.

La Ingeniería del Software, es el término utilizado por primera vez por Fritz Bauer en la primera conferencia sobre desarrollo de software (crisis del software) patrocinada por el Comité de Ciencia de la OTAN celebrada en Garmisch, Alemania, en octubre de 1968. Puede definirse según Alan Davis como:

"la aplicación inteligente de principios probados, técnicas, lenguajes y herramientas para la creación y mantenimiento, dentro de un coste razonable, de software que satisfaga las necesidades de los usuarios".

Importancia de la Ingeniería de Software

Actualmente casi todos los países dependen de sistemas complejos basados en computadora. El software en estos sistemas representa una proporción grande y creciente del costo total del sistema.

La experiencia previa en la construcción de estos sistemas mostró que un enfoque informal para el desarrollo de software no era muy bueno, debido a que los grandes proyectos a menudo tenían años de atraso, costaban mucho más de lo presupuestado, eran difíciles de mantener y con un desempeño pobre.

Nuevas técnicas y métodos eran necesarios para controlar la complejidad inherente en los sistemas de software grandes. Éstas técnicas han llegado a ser parte de la ingeniería de software y aunque se utilizan ampliamente no se ha logrado que esto sea universal. Sin embargo aún existen problemas en la producción de software complejo que cumplan con las expectativas del cliente, que se libere en tiempo y que se encuentre con un presupuesto aceptable.

La Ingeniería de Software se encuentra en un estado de enfermedad crónica, de acuerdo con Pressman, ya que mientras crezca nuestra habilidad en la producción del software también lo hará la complejidad de los sistemas de software solicitados

Las nuevas tecnologías de hardware y software impusieron nuevas demandas a los ingenieros de software, debido a que muchas compañías no aplican de forma efectiva las técnicas de la Ingeniería de Software. Sin embargo, la batalla no está perdida, aunque no estamos tan mal todavía existen muchas cosas por mejorar.

¿Porque existen sistemas de información tan complejos? en una palabra, por el hombre. El conocimiento humano y las grandes capacidades computacionales se aprovechan para realizar la tarea encomendada. Un sistema de información debe:

- Funcionar adecuadamente
- Fácil de manejar
- Adecuarse a las necesidades para lo cual fue diseñado

Los ingenieros de software pueden estar orgullosos de sus logros, porque sin software complejo no existiría la Internet y telecomunicaciones modernas y todas las formas de viajar serían peligrosas y caras.

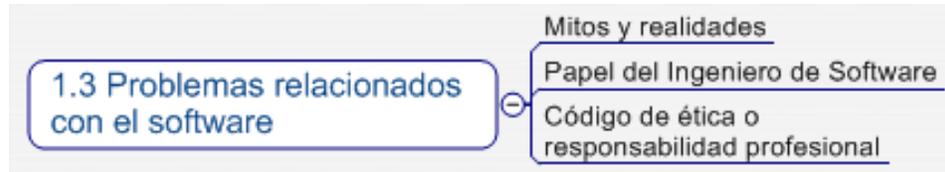
Diferencia entre Ingeniería de Software y Ciencias de la Computación. La Ciencia de la computación son teorías y métodos subyacentes a las computadoras y a los sistemas de software, en cambio la Ingeniería de software se refiere a los problemas prácticos de producir software.

Diferencia entre Ingeniería de Software e Ingeniería de Sistemas. La Ingeniería de Sistemas basados en computadoras se refiere a todos los aspectos del desarrollo y de la evolución de sistemas complejos donde el software juega un papel principal. Por lo tanto la ingeniería de sistemas comprende el desarrollo de hardware, políticas y procesos de diseño y distribución de sistemas, así como a la ingeniería de software.

La Ingeniería de sistemas es más antigua que la de Software, sin embargo, puesto que se ha incrementado el porcentaje de software en los sistemas, las técnicas de ingeniería de software (modelado de casos de uso, administración de la configuración, entre otros) se utilizan en el proceso de ingeniería de sistemas.

Tema 1.3 Problemas relacionados con el software.

Mapa conceptual



Objetivo específico

Al término del tema, el alumno expresará con sus palabras el papel del ingeniero de software, desechando los mitos que existen actualmente alrededor del proceso de producción de software y aumentará su sensibilidad hacia el código de ética de un ingeniero de software.

Mitos y realidades

Los administradores con responsabilidades sobre el software, al igual que sus pares en la mayoría de las disciplinas, a menudo están bajo presión por mantener los presupuestos, evitar que los itinerarios se extiendan y mejorar la calidad. De la misma forma que una persona a punto de ahogarse se aferra a un tronco, con frecuencia el administrador de software se aferra a un mito (mitos de la administración), si siente que esta creencia reducirá la presión, aún en forma temporal.

De esta manera, se entiende por mito a las creencias acerca del software y de los procesos empleados para construirlo. Los mitos del software se pueden rastrear hasta los primeros días de la computación. Los mitos tienen ciertos atributos que los convierten en insidiosos. Por ejemplo algunos mitos parecen una relación de hechos razonables (algunas veces contienen elementos verdaderos):

- Mito. Ya se tiene un libro de estándares y procedimientos para la construcción de software ¿esto proporcionará a mi gente todo el conocimiento necesario?
- Realidad. Tal vez sea verdad que el libro de estándares existe, pero ¿se usa?, ¿los encargados de la construcción de software saben de su existencia? ¿el libro refleja la práctica moderna de la ingeniería de software? ¿está completo? ¿es adaptable? ¿está dirigido al mejoramiento de tiempo de entrega sin dejar de enfocarse a la calidad? En muchos casos la respuesta a todas estas preguntas es no.
- Mito. Si se está atrasado en el itinerario es posible contratar más programadores para así terminar a tiempo (algunas veces llamado el concepto de la horda mongola)
- Realidad. El desarrollo de software no es un proceso mecánico como la manufactura. En palabras de Brooks “Agregar gente a un proyecto de software atrasado lo atrasa más”. De inicio este enunciado podría parecer contrario a la intuición. Sin embargo, cuando se agregan nuevos integrantes a un equipo, la gente que ya estaba trabajando debe invertir tiempo en la enseñanza a los recién llegados lo cual reduce el tiempo

dedicado al esfuerzo para el desarrollo del productivo. Se puede agregar gente pero solo de una manera apropiada.

- Mito. Si decido subcontratar el proyecto de software a un tercero, puedo relajarme y dejar que esa compañía lo construya.
- Realidad. Si una organización no entiende cómo administrar y controlar internamente los proyectos de software de manera invariable entrará en conflicto al subcontratar ese tipo de proyectos.

El cliente que solicita un software de computadora puede ser la persona del escritorio de al lado. Un grupo técnico en el piso de abajo, el departamento de ventas o de mercadotecnia, o una compañía externa que ha solicitado el software bajo contrato. En muchos casos, el cliente cree en mitos acerca del software porque los profesionales y administradores del software hacen muy poco para corregir la desinformación. Los mitos conducen a expectativas falsas (del cliente) y en definitiva a insatisfacción con el desarrollador.

- Mito. Un enunciado general de los objetivos es suficiente para comenzar a escribir programas, los detalles se pueden afinar después.
- Realidad. A pesar de que no siempre es factible que el enunciado de los requerimientos sea comprensible y estable, un enunciado ambiguo de los objetivos es la receta perfecta para el desastre. Los requerimientos precisos (los cuales derivan usualmente en forma iterativa) se desarrollan solo mediante la comunicación continua y efectiva entre el cliente y el desarrollador.
- Mito. Los requerimientos del proyecto cambian de manera continua, pero el cambio puede ajustarse con facilidad porque el software es flexible.
- Realidad. Es verdad que los requerimientos del software cambian, pero el impacto del cambio varía de acuerdo con el movimiento con que éste se introduce. Cuando los cambios en los requerimientos se solicitan en etapas tempranas (antes de iniciar con el diseño o el código), el impacto en el costo es relativamente pequeño. Sin embargo conforme pasa el tiempo, el impacto en el costo crece con rapidez y el cambio puede provocar una convulsión que requiera recursos adicionales y una modificación significativa en el diseño.

Los mitos que aún subsisten entre los desarrolladores del software han permanecido a través de 50 años de cultura de programación. Durante los primeros años del software, la programación era vista como una forma de arte, por ello, las viejas formas y actitudes son difíciles de eliminar.

- Mito. Una vez que el programa ha sido escrito y puesto a funcionar, el trabajo está terminado.
- Realidad. Alguien dijo alguna vez que entre mas rápido se comience a escribir código, más tiempo pasará para que el programa esté terminado. Los datos de la industria indican que entre 60 y 80 por ciento de todo el esfuerzo aplicado en el software se realizará después de que el sistema haya sido entregado al cliente por primera vez.
- Mito. Mientras el programa se esté ejecutando no hay forma de evaluar su calidad.
- Realidad. Uno de los mecanismos mas efectivos para el aseguramiento de la calidad del software se puede aplicar desde el inicio de un proyecto; la revisión técnica formal. Las revisiones del software son “un filtro de calidad” que han probado ser más efectivas que las pruebas para encontrar ciertas clases de errores en el software.
- Mito. El único producto del trabajo que puede entregarse para tener un proyecto exitoso es el programa en funcionamiento.

- Realidad. Un programa en funcionamiento es sólo una parte de la configuración del software que incluye muchos elementos. La documentación proporciona un fundamento para la ingeniería exitosa y, aún más importante, representa una guía para el mantenimiento del software.
- Mito. La ingeniería del software obligará a emprender la creación de una documentación voluminosa e innecesaria y de manera invariable tornará más lento el proceso.
- Realidad. La ingeniería del software no se refiere a la elaboración de documentos. Está relacionada con la creación de calidad. Una mejor calidad conduce a la reducción de los trabajos redundantes. Y una menor cantidad de trabajos redundantes resulta en menores tiempos de entrega.

Muchos profesionales de los sistemas reconocen la falacia de los mitos del software. Por el contrario, las actitudes y los métodos habituales conducen a adoptar malas prácticas administrativas y técnica, a pesar de que la realidad exige un mejor enfoque. El reconocimiento de las realidades del software es el primer paso hacia la formulación de soluciones prácticas para la ingeniería del software.

Papel del ingeniero de software

En el ámbito de la ingeniería de software es importante identificar a los diversos usuarios relacionados con el proceso de desarrollo del mismo, ya que la posición ética apropiada depende del tipo de vista de los individuos que estén involucrados.

Así se puede decir que los usuarios directos operan el sistema e interactúan directamente, por ejemplo los capturistas de datos, administradores de bases de datos, entre otros. Los usuarios indirectos utilizan la información producida por el sistema pero no opera la computadora, por ejemplo los gerentes, jefes del departamento, directores o subdirectores. Y finalmente, los usuarios responsables administrativamente, inspeccionan la inversión en desarrollo o uso del sistema, tienen la responsabilidad de organizar, planear y ejecutar los planes para el control de las actividades de los sistemas de información. Por ejemplo los administradores de redes (local, internet, intranet, extranet), administradores de bases de datos, entre otros.

El rol de un ingeniero de software tiene que ir más allá de sus capacidades operativas, es decir, más allá de:

- Capacidad de análisis para examinar alguna petición y de ser posible encontrar la oportunidad de resolverla mediante el uso de la computadora
- Conocimiento para recopilar e interpretar los datos y relacionarlos con la solución computacional
- Entendimiento para identificar las áreas donde las actividades se deben sistematizar y en donde realizarlos manualmente
- Conocimientos para elegir los mejores métodos de entrada, almacenamiento, salidas, desarrollo y puesta en marcha del sistema
- Capacidad para comunicarse efectivamente con otros

Así pues es necesario que las responsabilidades de los encargados de la administración y desarrollo de software tomen conciencia de los siguientes puntos con el fin de mejorar en su desempeño, no solo como especialista en su ramo, sino abarcando otras áreas de conocimiento

que le permitan darle un valor agregado a su trabajo.

- Los desarrolladores de software se podían dar el lujo de ser expertos en el área de computación, sin embargo hoy día cualquier diseñador o líder de proyecto que involucre software deberá ser conocedor en la administración de la información en los diversos departamentos o áreas de interés.
- Por otra parte, continuamente surgen mejores herramientas de desarrollo de sistemas, lo cual implica que el ingeniero de software, también tendrá la tarea de conocer, menjar y recomendar las mejores herramientas tecnológicas, que se ajusten a las necesidades de los usuarios, identificando costos y beneficios.
- Los ingenieros de software analizarán todas las peticiones para el uso de sistemas en computadora, y de ser posible recomendará la forma o formas de resolverla mediante el desarrollo de algún software informático.

Código de ética o responsabilidad profesional

Los ingenieros de software deben aceptar que su trabajo comprende amplias responsabilidades no solo la aplicación de habilidades técnicas. Su trabajo se lleva a cabo dentro de un marco legal y social. La ingeniería de Software está acotada por las leyes locales nacionales e internacionales, no basta con decir que los ingenieros deben poseer estándares normales de honestidad e integridad, sin embargo existen áreas donde los estándares de comportamiento aceptable no están acotados por las leyes, sino por la noción de responsabilidad profesional:

- **Confidencialidad.** Los ingenieros normalmente deben respetar la confidencialidad de sus empleadores o clientes independiente de que se haya firmado un acuerdo formal de confidencialidad.
- **Competencia.** Los ingenieros no deben falsificar su nivel de competencia. No deben aceptar conscientemente trabajos que esten fuera de su capacidad.
- **Derechos de propiedad intelectual.** Los ingenieros de software deben estar pendientes de las leyes locales que gobiernan el uso de la propiedad intelectual como las patentes, el copyright, entre otros. Debe asegurarse de que la propiedad intelectual de los empleadores y clientes este protegida.
- **Uso inapropiado de las computadoras.** Los ingenieros de software no deben utilizar sus habilidades técnicas de forma inapropiadas con las computadoras de otras personas. El uso impropio puede ir desde lo relativamente trivial (uso de juegos en una máquina de algún empleador) hasta los extremadamente serios (diseminación de virus).

Organizaciones como la ACM, el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) y la British Computer Society publican un código de conducta profesional y ética. Este código existe tanto en forma reducida como en forma extendida.

En cualquier situación donde diferentes personas tienen diferentes puntos de vista y objetivos, es posible encontrar dilemas éticos, los cuales aparecen en nuestra vida profesional y, afortunadamente en muchos casos son relativamente menores o se pueden resolver sin mucha dificultad, sin embargo nadie se encuentra exento de encontrarse en una situación difícil de resolver.

El área de ética y responsabilidad profesional ha recibido creciente atención en últimos años. Los principios de ética se pueden considerar desde un punto de vista filosófico, y la ética de la Ingeniería de software se debe discutir con referencia a estos principios básicos. Laudon (1995) y Huff y Martin (1995)

Tema 1.4 Procesos de producción de software

Mapa Conceptual



Objetivo específico

Al término del tema, el alumno identificará las características principales de los diferentes procesos de producción de software: cascada, prototipo, incremental, evolutivo y espiral, así como las principales ventajas y desventajas de cada modelo esquemático.

Proceso de Software

En este tema se identificarán las principales actividades realizadas en la Ingeniería de Software que permiten la producción de software. Empezaremos por definir el concepto de proceso de software.

Cuando se trabaja para construir un producto o sistema es importante seguir una serie de pasos predecibles, tal como un mapa de carreteras que siempre nos indica en qué dirección vamos o bien si ya llegamos a nuestro objetivo. Entonces podemos decir que el proceso de software es un conjunto de actividades o resultados asociados que producen un producto de software. Existen cuatro actividades fundamentales de procesos:

- Especificación de software. La funcionalidad del software y las restricciones sobre su operación deben quedar definidas.
- Desarrollo del software. Debe producirse software que cumpla con la especificación.
- Validación de software. El software debe validarse para asegurar que cumpla los requisitos del cliente.
- Evolución del software. El software debe evolucionar para cumplir con los cambios requeridos por el cliente.

El proceso de software ofrece estabilidad, control y organización a una actividad que puede volverse caótica si no se controla. Sin embargo un enfoque de ingeniería de software debe requerir solo aquellas actividades, controles y documentaciones apropiadas para el equipo del proyecto y el producto que ha de producirse.

Junto a esta definición también se encuentran los conceptos de eficiencia y eficacia que de una u otra forma también contribuyen al producto final del software.

La eficiencia y eficacia son medidas relativas usadas para comparar sistemas. La eficiencia es la medida de lo que se produce dividido entre lo que se consume. Por lo general se trata de una proporción o porcentaje: $\text{Producción/Consumo} = \%$

Por ejemplo, cuando hablamos de la eficiencia de la funcionalidad de un motor de alguna máquina, su productividad se medirá a partir de la energía producida y la consumida:

$$\begin{aligned} \text{Motor} &\Rightarrow \text{Energía producida} \Rightarrow \text{trabajo realizado} \\ \text{Electricidad o Combustible} &\Rightarrow \text{Energía consumida} \\ &(\text{trabajo realizado}) / \text{energía consumida (electricidad o combustible)} \end{aligned}$$

El resultado casi siempre es del 50% o menos, debido a la pérdida de energía por causas de fricción y generación de calor.

Por otra parte, la eficacia es una medida en grado en el que un sistema cumple sus metas. De esta manera se puede concebir un nuevo concepto estándar de desempeño de sistemas, el cual tiene que ver tanto de eficiencia como de eficacia. El estándar de desempeño es un objetivo específico del sistema, por ejemplo, una campaña podría:

- En una campaña de mercadeo, como se muestra en la siguiente gráfica, conseguir que cada representante de ventas venda una cantidad de cierto tipo de producto equivalente a 100,000 dólares cada año, por lo que se espera un desempeño alto de ventas.

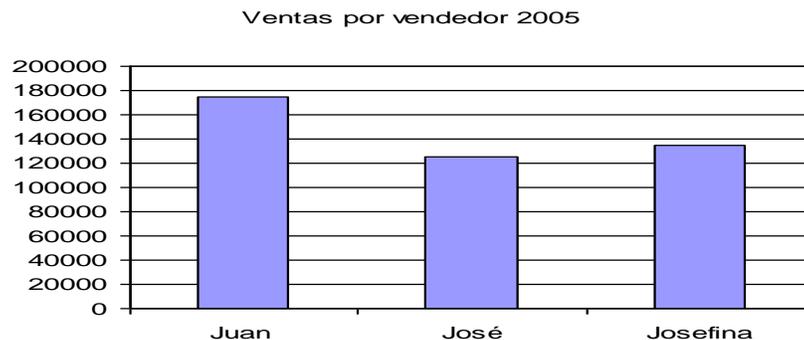


Fig.1.4.1 Estándar de desempeño (ventas)

- En otro caso, de un proceso de manufactura, que la producción no rebase más de un por ciento de partes defectuosas, se preferirá que su comportamiento sea descendente.

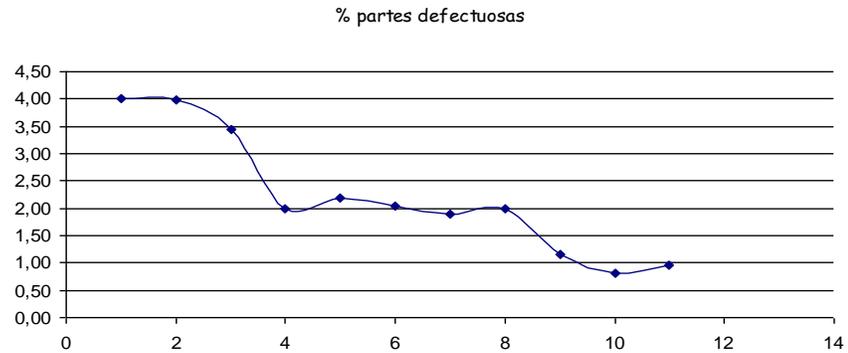


Fig.1.4.2 Estándar de desempeño (manufactura)

Etapas del proceso

El proceso de la ingeniería del software es el elemento que mantiene juntos los estratos de la tecnología y que permite el desarrollo racional y a tiempo del software de computadora. El proceso define un marco de trabajo que debe establecerse para la entrega del producto de software. El proceso de software forma la base para el control de la gestión de los proyectos de software y establece el contexto en el cual se aplican los métodos técnicos, se generan los productos de trabajo (modelos, documentos, datos, reportes, formatos, etc.) se establecen los fundamentos y se asegura la calidad.

De esta manera, se puede decir que la ingeniería de software es una tecnología estratificada, es decir, como se muestra en la siguiente figura 1.4.3, cualquier enfoque de ingeniería debe estar sustentado en un compromiso de calidad que deben fomentar la cultura de mejora continua del proceso, y es esta cultura la que al final conduce al desarrollo de enfoques muy efectivos para la ingeniería del software.



Fig. 1.4.3 Proceso de Software

Los métodos abarcan un amplio espectro de tareas que incluyen la comunicación, el análisis de requisitos, el modelado del diseño, la construcción del programa, la realización de pruebas y el soporte. Incluyen actividades de modelado y otras técnicas descriptivas.

Finalmente las herramientas de la ingeniería del software proporcionan el soporte automatizado o semiatomizado para el proceso y los métodos. Cuando las herramientas se integran de forma que la información que genere una de ellas pueda usarla otra, se dice que se ha establecido un sistema para el soporte del desarrollo de software, que con frecuencia se denomina ingeniería del software asistida por computadora.

Modelo del proceso de software

Los modelos son simplificaciones, por lo tanto un modelo de procesos del software es una abstracción de un proceso real. Estos modelos incluyen actividades que son parte de los procesos y productos del software y el papel de la gente involucrada en la ingeniería del software, por ejemplo:

- Modelo de flujo de trabajo: Muestra la secuencia de actividades en el proceso en conjunto con las entradas, salidas y dependencias, representan actividades o acciones humanas.
- Modelo de flujo de datos o de actividad: Representa el proceso como una transformación de los datos, muestra como la entrada en el proceso se transforma en una salida como un diseño
- Modelo de rol o acción: Representa los roles de la gente involucrada en el proceso del software y las actividades de las que son responsables.

Existe una gran variedad de modelos o paradigmas de desarrollo de software:

- Enfoque de cascada: Considera las actividades anteriores y las representa como fases de procesos separados, después de que cada etapa queda definida se firma y el desarrollo continúa con la siguiente etapa.

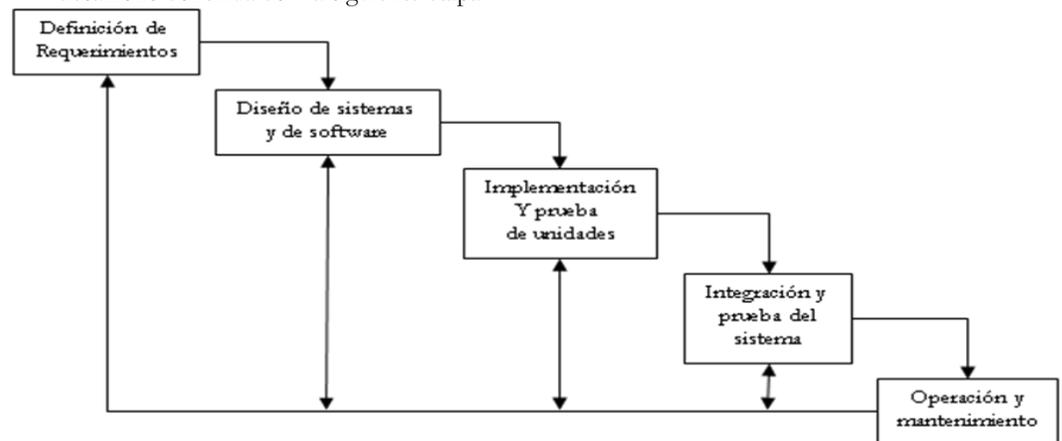


Fig. 1.4.4 Modelo Cascada

- Desarrollo evolutivo: Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones muy abstractas, este se refina basándose en las especificaciones del cliente para producir un sistema que satisfaga sus necesidades.

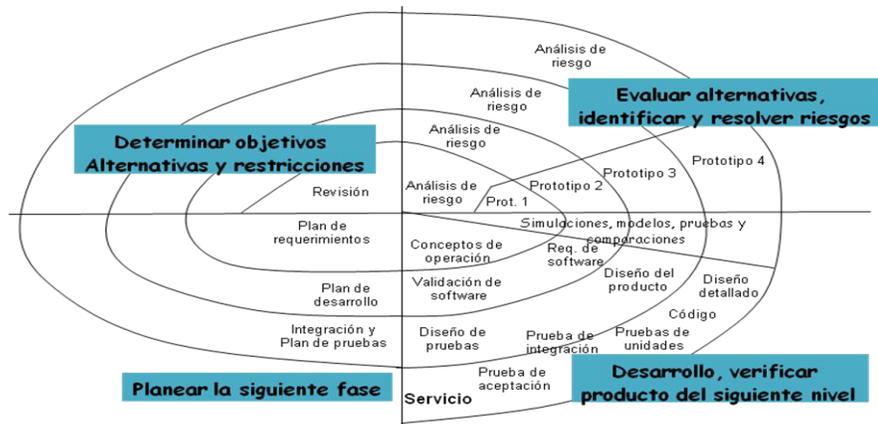


Fig. 1.4.5 Modelo Espiral

- Transformación formal: Este enfoque se basa en la producción de la especificación matemática formal del sistema y la transformación de esta especificación a un programa utilizando métodos matemáticos.
- Sistema de ensamblaje de componentes reutilizables: Esta técnica supone que las partes del sistema existen. El proceso de desarrollo del sistema se enfoca a la integración de esas partes más que a construirlo desde sus inicios.

Costos de la Ingeniería del Software

La distribución precisa del costo del software depende del proceso utilizado y del tipo de software que se vaya a desarrollar, sin embargo la integración y prueba del sistema pueden ser las actividades de desarrollo más caras. Si el software se desarrolla utilizando un enfoque evolutivo, no existe división entre la especificación, el diseño y el desarrollo.

Además de los costos de desarrollo, existen costos asociados a cambios que se le hacen al software una vez que está en uso los cuales se cumplen para el software personalizado. Para productos de software que se venden integrado en PC's el perfil del costo es diferente. Estos productos comúnmente se desarrollan a partir de una especificación inicial utilizando un enfoque de desarrollo evolutivo.

Cuadro 1.4.1 Avance porcentual de tiempo en cada fase

Especificación	Diseño	Desarrollo	Pruebas	Integración
0	25	50	75	100

Atributos de un buen software

Así como los servicios que proveen los productos de software tienen un cierto número de atributos asociados que reflejan la calidad de ese software. Estos atributos reflejan su comportamiento durante su ejecución y en la estructura y organización del programa fuente y en la documentación asociada, por ejemplo, el tiempo de respuesta y la comprensión del programa fuente.

Las características esenciales de un sistema de software bien diseñado son:

- **Mantenibilidad:** El software debe escribirse de tal forma que pueda evolucionar para cumplir las necesidades de cambio de los clientes. Este es un atributo crítico debido a que el cambio en el software es una consecuencia inevitable de un cambio en el entorno de negocios.
- **Confiabilidad:** La confiabilidad del software tiene un gran número de características, incluyendo la fiabilidad, seguridad y protección, el software confiable no debe causar daños físicos o económicos en el caso de una falla del sistema.
- **Eficiencia:** El software no debe hacer que se malgasten los recursos del sistema, como la memoria y los ciclos de procesamiento por lo tanto la eficiencia incluye tiempos de respuesta y de procesamiento, utilización de la memoria, etc.
- **Usabilidad:** El software debe ser fácil de utilizar, sin esfuerzo adicional, por el usuario para quien está diseñado. Esto significa que debe tener una interfaz de usuario apropiada y una documentación adecuada.

Retos fundamentales de la Ingeniería de Software

Sin embargo existen varios retos que la ingeniería de software tiene que considerar a lo largo del proceso de desarrollo de software, los cuales se explican a continuación.

- **El reto de lo heredado:** La mayoría de sistemas de software grandes que actualmente están en uso fueron desarrollados hace muchos años y aun llevan acabo funciones críticas de negocios, el reto de lo heredado es mantener y actualizar este software de tal forma que se eviten los costos excesivos y que los servicios esenciales del negocio continúen funcionando.
- **El reto de la heterogeneidad:** Cada vez más se requiere que los sistemas operen como sistemas distribuidos en redes que incluyen diferentes tipos de computadoras y con diferentes clases de sistemas de soporte. El reto de la heterogeneidad es desarrollar técnicas para construir software confiable que sea lo suficientemente flexible para adecuarse a esta heterogeneidad.
- **El reto de la entrega:** Muchas técnicas tradicionales de la ingeniería de software consumen tiempo para producir un software de calidad, sin embargo hoy día los negocios deben tener una gran capacidad de respuesta y cambiar con mucha rapidez, por lo que el software de soporte también debe cambiar con la misma rapidez. El reto de la entrega es reducir los tiempos de entrega para sistemas grandes y complejos sin comprometer la calidad del sistema.

Unidad 2. Procesos y administración de software

Tema 2.1 Visión general

Objetivo Particular

Al término de la unidad, el alumno reconocerá la importancia de los procesos involucrados en el desarrollo de software y su administración, realizando la planeación de proyectos de software.

Mapa conceptual

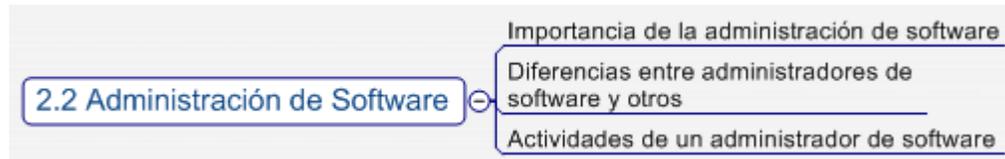


Tema 2.2 Administración de software

Objetivo específico

Al final del tema, el alumno identificará las actividades principales que competen a un administrador de software, ejemplificando y diferenciándolas con respecto a otros administradores.

Mapa conceptual



Importancia de la administración de software

La necesidad de administrar es una distinción importante entre un desarrollador profesional de software y el programador no profesional. La administración de software es necesaria debido a que la Ingeniería de Software siempre está sujeta a restricciones de presupuesto y calendarización a las que debe ajustar de la organización que desarrolla el software. El trabajo del administrador de software es asegurar que éstos cumplan dichas restricciones y entregar software que contribuya a las metas del negocio.

Los administradores de software:

- Supervisan el trabajo para asegurar que se lleva a cabo de acuerdo a los estándares requeridos.
- Supervisan el progreso para comprobar que el desarrollo va de acuerdo al presupuesto y al calendario
- Una buena administración no garantiza el éxito del proyecto, sin embargo la mala siempre asegura el fracaso del mismo.

Diferencias entre administradores de software y otros administradores

Algunas diferencias entre los administradores de software y otros tipos de administradores son:

El producto es intangible. El administrador en la construcción de un edificio, un barco o la apertura de un negocio puede ver si hay algún desfase en tiempo ya que el producto es visible. Los administradores de software deben confiar en otros para producir la documentación necesaria y revisar el progreso.

No existen procesos de software estándar. En las disciplinas de ingeniería con larga historia el proceso se prueba y se verifica, La comprensión del proceso de software se ha desarrollado en forma significativa, sin embargo aún no se puede predecir con certeza cuando un proceso particular tiende a desarrollar problemas.

A menudo los proyectos grandes de software son únicos. En general los proyectos grandes son diferentes a proyectos previos, aún cuando los administradores cuenten con amplia experiencia los cambios tecnológicos rápidos en las computadoras y telecomunicaciones hacen parecerla obsoleta. Las lecciones aprendidas pueden no ser transferidas a los nuevos proyectos.

Debido a estos problemas, no es sorprendente que algunos proyectos de software se retrasen, sobrepasen el presupuesto y estén fuera de tiempo.

En la tabla 1, se comparan las actividades diarias de un administrador de software contra otro tipo de administradores.

Cuadro 2.2.1 Comparativo de administradores

Administrador de software	Otros administradores
Producto intangible	Producto tangible
Procesos de software sin estándares o normas previas. Cuando un proceso en particular va a desarrollar problemas	Procesos con normas o estándares
Proyectos grandes son únicos y se vuelven obsoletos	Proyectos grandes se pueden adaptar más rápido al cambio

Actividades de un administrador de software

Aunque es imposible describir todas las actividades de un administrador de software, muchos administradores son responsables de algunas o todas de las siguientes actividades:

- Redacción de la propuesta
- Planeación y calendarización del proyecto
- Costeo del proyecto.
- Supervisión y revisión del proyecto
- Selección y evaluación del personal
- Redacción y presentación de informes

Redacción de la propuesta. Se describen objetivos del proyecto y como se llevarán a cabo, incluye estimado de costo y calendarización, justifica porque el proyecto debe darse a una organización o un equipo en particular.

Planeación y calendarización del proyecto. Se refiere a la identificación de actividades, hitos y entregas producidas por un proyecto, por lo tanto se debe bosquejar un plan para guiar el desarrollo hacia las metas del proyecto

Costeo del proyecto. Se refiere al estimado de los recursos requeridos para llevar a cabo el plan del proyecto.

Supervisión y revisión del proyecto. La supervisión de un proyecto es una actividad continua, el administrador debe tener conocimiento del progreso del proyecto y comparara los costos del proyecto contra los costos reales. Por otra parte el tiempo de desarrollo para un proyecto grande de software puede ser de varios años, durante ese tiempo los objetivos organizacionales tienden, obviamente a cambiar, estos cambios pueden significar que el software ya no se requiere mas o que los requerimientos originales del proyecto son inapropiados. La administración puede decidir parar el desarrollo del software o cambiar el proyecto para adecuarlo a los cambios de los objetivos de la organización.

Selección y evaluación del personal. Los administradores del proyecto tienen que seleccionar a las personas que trabajarán en su proyecto, sin embargo en muchos casos los administradores tienen que establecer un equipo ideal mínimo, debido a:

- El presupuesto del proyecto no cubre la contratación de personal con sueldos altos.
- El personal con experiencia apropiada no está disponible dentro o fuera de la organización, es imposible reclutar nuevo personal para el proyecto.
- La organización desea desarrollar las habilidades de sus empleados. El personal inexperto puede ser asignado al proyecto para aprender y adquirir experiencia.
- Comunicarse de forma oral y escrita es una habilidad esencial que un administrador de proyectos debe tener.

Redacción y presentación de informes. Por lo general el administrador del proyecto es responsable de dar informes tanto a clientes como a las organizaciones contratantes.. Dichos administradores deben redactar documentos concisos y coherentes que resuman la información crítica de los informes detallados del proyecto.

Tema 2.3 Planeación y calendarización del proyecto

Objetivo específico

Al final del tema, el alumno reconocerá la importancia de planear y calendarizar los proyectos de software a fin de identificar las actividades principales que competen a un administrador de software, ejemplificando y diferenciándolas con respecto a otros administradores.

Mapa Conceptual



Planeación de proyectos

La planeación es un proceso iterativo que solamente se completa cuando el proyecto mismo se termina. Conforme la información se hace disponible el plan debe revisarse regularmente. Las metas globales del desarrollo son un factor importante que debe considerarse cuando se formula el plan del proyecto conforme esto cambia, los cambios en el proyecto serán necesarios.

Existen varios tipos de planes, entre los que destacan los siguientes:

- Plan de calidad
- Plan de validación
- Plan de administración
- Plan de mantenimiento
- Plan de desarrollo personal

El plan del proyecto fija los recursos disponibles, divide el trabajo y crea un calendario de trabajo, en algunas organizaciones el plan del proyecto es un único documento que incluye todos los diferentes tipos de planes mencionados anteriormente. En otros casos este plan solo se refiere al proceso de desarrollo, otros pueden estar referenciados pero son proporcionados por separado.

Algunos planes incluyen las siguientes secciones:

- Organización del proyecto. Describe la forma en que el equipo de desarrollo está organizado, la gente involucrada y sus tareas asignadas.
- Análisis de riesgos. Describe los riesgos del proyecto, su probabilidad y las estrategias de reducción de riesgo propuestas.
- Requerimientos de recursos de hardware y software. Se incluye la tecnología adecuada, la estimación de costos y fechas de entrega.
- División de trabajo. Describe la distribución del proyecto en actividades e identifica los hitos y productos a entregar asociados con cada actividad.
- Programas de proyecto. Describe las dependencias entre actividades, el tiempo estimado para alcanzar cada hito y la asignación de la gente a las actividades.
- Mecanismos de supervisión e informe. Describe la administración de informes, cuando deben producirse y los mecanismos de supervisión del proyecto.

Dado que el software es intangible los administradores necesitan información (documentos que describan el estado del software que se está desarrollando), sin ésta es imposible juzgar el proceso y no se pueden actualizar los costos y calendarios. De esta manera se involucran los “Hitos”, que son puntos finales de una actividad del proceso del software.

En cada hito debe existir una salida formal, como un informe que se debe presentar al administrador. Los informes de hitos no deben ser documentos amplios, deben ser informes de los logros en una actividad del proyecto. Los hitos deben representar el fin de una etapa lógica en el proyecto.

Los hitos indefinidos como “80% de código completo” son imposibles de validar y carecen de utilidad para la administración del proyecto. Como regla general los productos a entregar son hitos, pero estos no necesariamente productos a entregar, es decir, pueden ser resultados internos del proyecto.

Análisis de Riesgos

Durante el desarrollo de software se corren diversos tipos de riesgos, por lo que el administrador de software debe estar preparado para atacar cualquier contingencia que se llegara a presentar. Estos se esquematizan en el cuadro 2.3.1:

Cuadro 2.3.1. Riesgos en el desarrollo de software

Riesgos	Tipo de riesgo	Descripción
Rotación de personal	Proyecto	Personal con experiencia abandona el proyecto antes de que finalice.
Cambio de administración	Proyecto	Habrà un cambio de administración organizacional con diferentes prioridades
No disponibilidad del hardware	Proyecto	El hardware esencial del proyecto no será entregado a tiempo.
Cambio de requerimientos	Proyecto y producto	Habrà mas cambios en los requerimientos que lo anticipado
Retrasos en la especificación	Proyecto y Producto	Las especificaciones de las interfaces esenciales no estarán a tiempo.
Subestimación del tamaño	Proyecto y Producto	El tamaño del sistema se ha subestimado
Cambio de tecnología	Negocio	La tecnología fundamental sobre la que se construirà el sistema se sustituye por nueva tecnología.
Competencia del producto	Negocio	Un producto competitivo se pone en venta antes de que el sistema se complete.

Calendarización de actividades

Usted ha seleccionado un modelo de proceso adecuado, ha identificado las tareas de ingeniería del software que hay que llevar a cabo, ha estimado la cantidad de trabajo y el número de personas necesarias, conoce las fechas límites de entrega e incluso ha considerado los riesgos. Ahora, es el momento de unir todos los puntos. Esto es, tiene que crear una red de tareas de ingeniería de software que le permitan conseguir el trabajo realizado a tiempo. Una vez creada la red, tiene que asignar la responsabilidad para cada tarea, asegurarse de hacerlo, y de adaptar la red antes de que los riesgos se conviertan en realidad. Lo anterior es la planeación temporal.

La planificación temporal es importante para construir sistemas complejos, muchas tareas de la ingeniería de software se realizan en paralelo y el resultado del trabajo desarrollado durante una tarea puede tener un gran efecto en el trabajo a realizar en otra tarea. Estas interdependencias son muy difíciles de comprender sin una planificación. También es virtualmente imposible evaluar el progreso en un proyecto de software normal o grande sin una planificación detallada.

Compartimentación. Divide el proyecto en un número de tareas manejables, es decir, se debe descomponer tanto el producto como el proceso.

Interdependencia. Existen actividades dependientes o independientes, es decir, se puede dar la situación en que una actividad no procede hasta que el resultado de otras no esté disponible.

Asignación de tiempo. Se debe identificar el número de unidades de tiempo (día, semana, mes, año, etc.) mediante la consideración de una fecha de inicio y una fecha de término.

Validación de esfuerzo. Se indicará el número de personas asignadas a una tarea específica.

Responsabilidades definidas. Cada tarea o actividad programada debe asignarse a un miembro o grupo de trabajo específico.

Resultados definidos. Cada tarea debe generar un resultado específico.

Hitos definidos. Se consigue un hito cuando se ha revisado la calidad de uno o más productos y se ha aceptado.

Técnicas de estimación de proyectos

Una planificación adecuada requiere de todas las tareas que aparecen en la red, el esfuerzo y tiempo se debe asignar de forma inteligente a cada tarea, además de que estén indicadas correctamente. Los recursos deben ser asignados al trabajo a realizar y los hitos deben situarse rigurosamente espaciados para que se pueda seguir el progreso.

Diagrama de Gantt.¹

Los cronogramas de barras o “gráficos de GANTT” fueron concebidos por el ingeniero norteamericano Henry L. Gantt, uno de los precursores de la ingeniería industrial contemporánea de Taylor. Gantt procuró resolver el problema de la programación de actividades, es decir, su distribución conforme a un calendario de manera tal que se pudiese visualizar el periodo de duración de cada actividad, sus fechas de iniciación y terminación e igualmente el tiempo total requerido para la ejecución de un trabajo. El instrumento que se desarrolló permite también que se siga el curso de cada actividad, al proporcionar información del porcentaje ejecutado de cada una de ellas, así como el grado de adelanto o atraso con respecto al plazo previsto.

¹El lector podrá consultar el caso práctico anexo (CD), con el fin de disipar las diversas dudas que puedan surgir. p. 57.

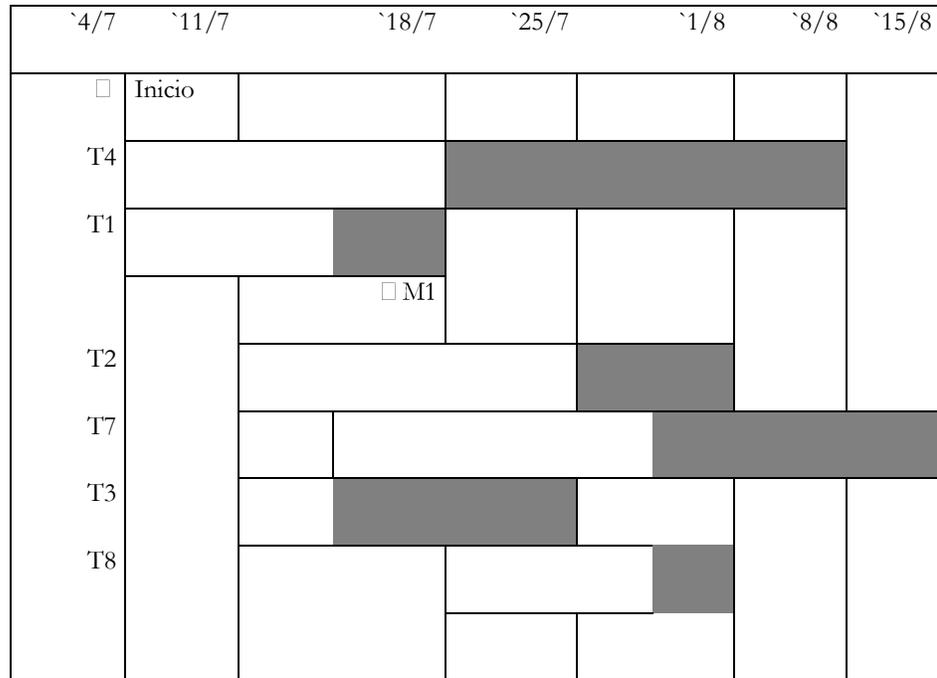


Figura 2.3.1 Diagrama de GANTT

Cuadro 2.3.2. Asignación de actividades

Tarea	Duración (días)	Dependencias	Ingeniero
T1	8		Salvador
T2	15		Ana
T3	15	T1 (M1)	Lorenzo
T4	10		Alfredo
T5	5	T2,T4 (M2)	Josefina
T6	20	T3 (M4)	Ana
T7	25	T6	Lorenzo
T8	15	T5, T7 (M3)	Alfredo
T9	15	T4	Salvador
T10	7	T8	Ana
T11	10	T10 (M5)	Lorenzo

La ventaja principal del gráfico radica en que su trazado requiere de un nivel mínimo de planificación, es decir, es necesario que haya un plan que ha de representarse en forma de gráfico. Los gráficos de Gantt se revelan muy eficaces en las etapas iniciales de la planificación. Sin embargo después de iniciada la ejecución de la actividad y cuando comienza a efectuarse modificaciones, el gráfico tiende a volverse confuso. Por eso se utiliza mucho la representación gráfica del plan en tanto que los ajustes (replanificación) requieren por lo general de la formulación de un nuevo gráfico. Para superar esa deficiencia se crearon dispositivos mecánicos, tales como cuadros magnéticos, fichas, cuerdas, etc. Que permite una mayor flexibilidad en las actualizaciones.

Aún en términos de planificación, existe todavía una limitación bastante grande en lo que se refiere a la representación de planes de cierta complejidad. El gráfico de Gantt no ofrece condiciones para el análisis de opciones, ni toma en cuenta factores como el costo. Es fundamentalmente una técnica de pruebas y errores. No permite, tampoco, la visualización de la relación entre las actividades cuando el número de éstas es grande.

Diagrama de PERT/CPM²

Los diagramas de PERT/CPM ayudan en la planeación, administración y control de un proyecto, tales como:

- Proyectos de construcción (Ingeniería Civil)
- Construcción de aviones
- Ruta de transporte económica
- Sistemas de armas militares
- Sistemas de gasoductos
- Sistemas de computación

Su objetivo es estimar fechas de inicio y término de las tareas que conforman un proyecto, sus siglas significan Program Evaluations and Review Technique, PERT (Técnicas de evaluación y revisión de programas)

En cambio los diagramas CPM, su objetivo es reducir el tiempo de terminación de un proyecto contando con más recursos. Sus siglas significan Critical Path Method CPM (Método de la Ruta Crítica).

La terminología de estos diagramas se describe en los siguientes párrafos. Los nodos o círculos son puntos de referencia, son puntos lógicos de conexión para diversas actividades. Las flechas implican tiempo y consumo de recursos. Las actividades ficticias no consumen tiempo ni recursos, además de que evitan conectar en forma directa dos nodos a través de dos flechas.

² El lector puede consultar el CD anexo, donde se ejemplifica un caso sobre la elaboración y uso del diagrama de PERT/CPM. p.8

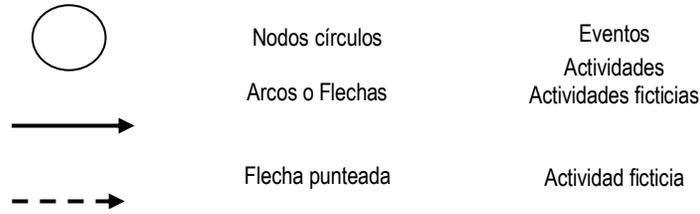


Figura 2.3.2 Simbología PERT

Antes de graficar un diagrama PERT debe considerar:

- Identificar todas las tareas o actividades asociadas a los proyectos y sus interrelaciones
- Ordenarlas en forma lógica
- Antes de que pueda comenzar una actividad, todas las actividades precedentes deben haber terminado
- Las flechas indican solo procedencia, ni su longitud ni su dirección tiene significado alguno
- Cada flecha (actividad) debe comenzar y terminar en un nodo de evento
- Ningún par de nodos de la red puede estar directamente conectados por más de una flecha

Cuadro 2.3.3. Actividades

Clave de La actividad	Descripción de la actividad	Predecesores de la actividad	Tiempo asignado
A	Identificación de necesidades	----	1
B	Diseño y modelado del sistema	A	2
C	Desarrollo del sistema	A	2
D	Pruebas del sistema	A	3
E	Instalación y mantenimiento del sistema	B,C,D	2

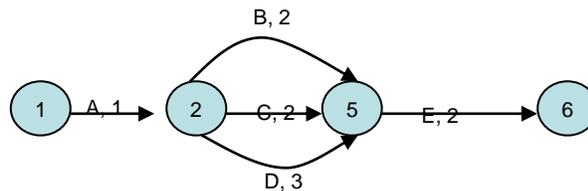


Figura 2.3.3 Diagrama de PERT erróneo

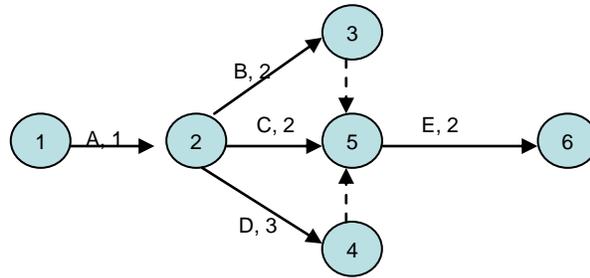


Figura 2.3.4 Diagrama de PERT correcto

- Cuando se numeran los nodos es aconsejable, y en particular en una red grande, utilizar múltiplos de 10 para que sea fácil incorporar cualesquiera cambios o adiciones futuras
- Todas las flechas de la red deben estar dirigidas, de izquierda a derecha
- La clasificación de las actividades (o sea, el listado de las actividades del proyecto), no deben ser más detallado que lo que se requiera para representar un plan de acción lógico y claramente definido

La ruta crítica es la ruta o camino que se lleva más tiempo para acabar el proyecto, y las actividades dentro de ella se llaman actividades críticas. Esta gráfica tiene por objetivo encontrar el tiempo esperado para terminar una actividad, la cuál se puede dar en semanas, días, meses, etc. de acuerdo al tipo de proyecto, y ésta se denota como D_{ij}

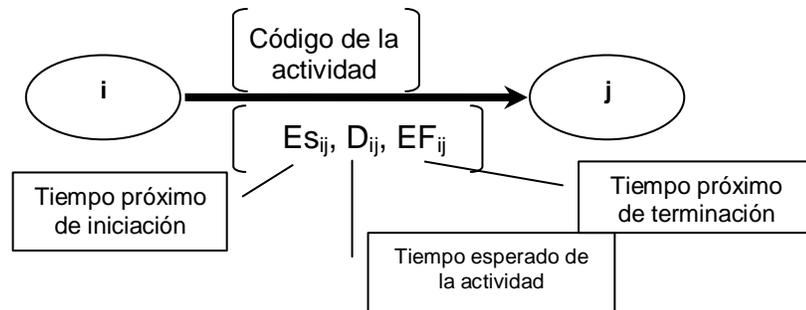


Figura 2.3.5. Simbología CPM, para el tiempo esperado

Si hay varias actividades precediendo a otra, el tiempo más próximo de iniciación para esta actividad es igual al mayor valor de los tiempos próximos de terminación para todas las actividades precedentes. Por otra parte, también es importante conocer el tiempo de holgura que en determinado momento se pueda tener sin que esto signifique un atraso en el proyecto. Dicho dato lo podemos calcular identificando el tiempo más lejano de inicio y término de una actividad, como se muestra en la figura

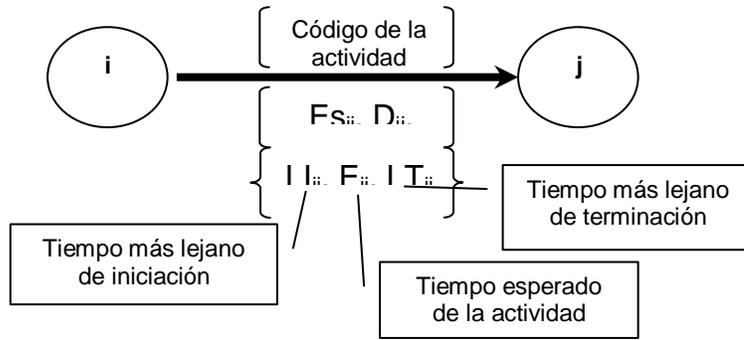


Figura 2.3.6. Simbología CPM, para el tiempo más lejano

Ejemplo 1:

Clave de la actividad	Descripción de la actividad	Predecesores de la actividad	Tiempo asignado
A	Identificación de necesidades	—	1
B	Diseño y modelado del sistema	A	2
C	Desarrollo del sistema	A	2
D	Pruebas del sistema	A	3
E	Instalación y mantenimiento del sistema	B,C,D	2

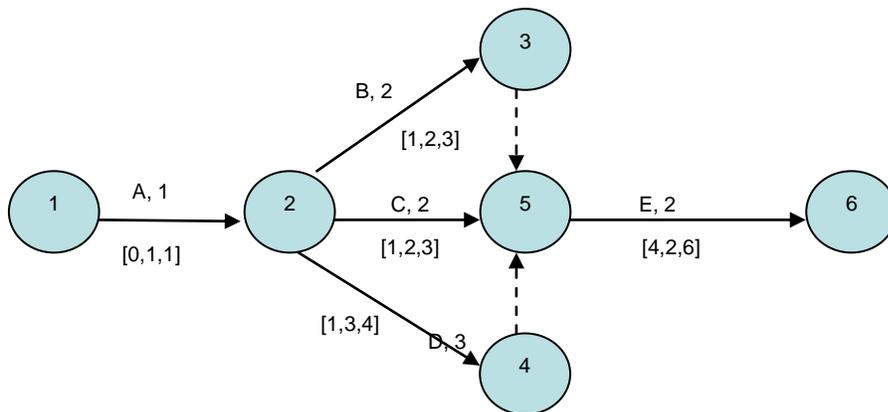


Figura2.3.7 Ejemplo de diagrama de PERT

Ejemplo 2:

Clave de la actividad	Descripción de la actividad	Predecesores de la actividad	Tiempo asignado
A	Diseñar el producto	—	6
B	Diseñar el empaque	—	2
C	Ordenar y recibir los materiales para el producto	A	3
D	Ordenar y recibir los materiales para el producto	B	3
E	Fabricar el producto	C	4
F	Fabricar el empaque	D	3
G	Empacar el producto	E	6
H	Prueba de mercado del producto	F	4
I	Prueba de mercado del empaque	G,H	1
J	Entregar a los distribuidores el producto	I	2

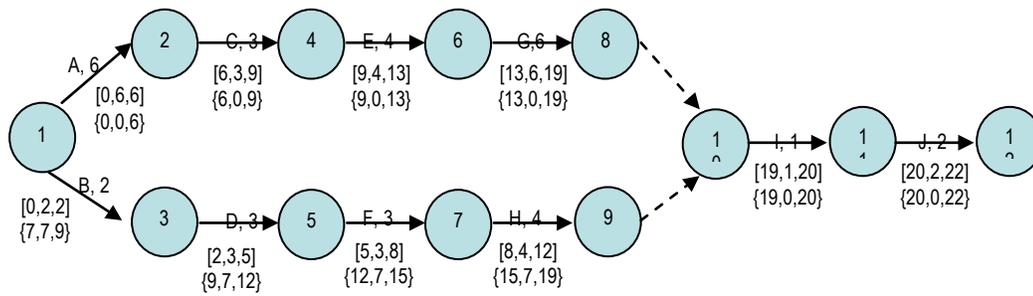


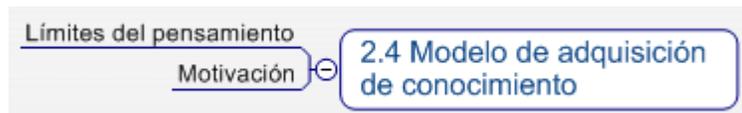
Figura2.3.8 Otro ejemplo de diagrama de PERT

Tema 2.4 Modelo de adquisición de conocimiento

Objetivo específico

Al final del tema, el alumno identificará los límites del pensamiento para conocer el proceso que lleva cualquier ser humano para procesar un nuevo conocimiento, así como las técnicas adecuadas de motivación, para aplicarlas en las personas que integran su equipo de trabajo y éste exprese sus capacidades adecuadamente.

Mapa Conceptual



Límites del pensamiento

Existen diferentes formas de demostrar nuestra inteligencia, educación o experiencia, sin embargo no todos pensamos igual o imaginamos lo mismo, por esto es importante estar conciente de los límites de la forma en que pensamos con el fin de ayudar al administrador a detectar las diferentes formas de pensar de su equipo de trabajo.

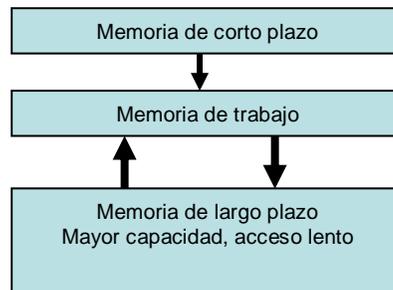


Figura 2.4.1 Proceso de adquisición de conocimiento

Es importante reconocer, que existen varios niveles de retención, por ejemplo los sistemas de software son entidades abstractas y los ingenieros tienen que recordar sus características para el proceso de desarrollo.

Memoria de corto plazo. Es de rápido acceso y capacidad limitada como las entradas de los sentidos que se reciben para un procesamiento inicial. Es comparable con los registros de una computadora en donde se utiliza para procesar información y no almacenarla.

Memoria de trabajo. Tiene mayor capacidad que la anterior y se usa para realizar procesamiento de información y poder retener la información por más tiempo. Análogamente se puede interpretar como la memoria RAM de una computadora.

Memoria de largo plazo. Tiene mayor capacidad, sin embargo su tiempo de acceso es relativamente lento y los mecanismos de recuperación de información son poco confiables, debido a que se olvidan cosas.

La información de un problema que entra en la memoria de corto plazo de los documentos

que se leen y de las pláticas con las personas se integra con otra información relevante de la memoria de largo plazo en la memoria de trabajo. El resultado de esta integración forma la base para la resolución de problemas, dicho resultado se almacena en la memoria de largo plazo para su uso futuro. Por supuesto, la solución puede ser incorrecta debido a que entre más información se tenga disponible se deberá modificar la memoria de largo plazo. Sin embargo, la información incorrecta no se descarta completamente sino que se retiene de alguna forma, esto es necesario debido a que aprendemos de los errores.

En 1980 Schneiderman comentó que los lectores de programas extraen la información en el programa y crean pedazos que forman una estructura semántica interna. Los programas no se comprenden bajo una base de instrucción por instrucción a menos que una instrucción represente un pedazo lógico.

El conocimiento adquirido durante el desarrollo de software y almacenado en la memoria de largo plazo se divide en dos clases:

Conocimiento semántico (Que pertenece a un mismo grupo). Este es el conocimiento de los conceptos como la operación de una instrucción de asignación, la noción de una clase de objetos la manera en que opera una técnica de búsqueda, etc. Este conocimiento se adquiere a través de la experiencia y el aprendizaje, y se retiene en una forma independiente de la representación.

Conocimiento sintáctico (coherencia). Este es el conocimiento de representación detallada por ejemplo escribir una descripción de un objeto en UML, funciones estándar de un lenguaje de programación, etc. Este conocimiento se retiene como un formulario sin procesar.

Los diferentes modos de adquisición del conocimiento sintáctico y semántico ayudan a explicar como aprenden los programadores experimentados un nuevo lenguaje de programación. Sin embargo la sintaxis del lenguaje tiende a mezclarse con la sintaxis de los lenguajes con los que están familiarizados.

En cuanto se desarrolla la comprensión de un concepto, se almacena en memoria como conocimiento semántico, donde éste se almacena en una forma conceptual abstracta.

Éste modelo explica porque para muchas personas aprender a programar es una habilidad que parece llegar de golpe después de períodos de dificultades. Las habilidades de programación requieren una comprensión de los conceptos semánticos y una separación de los conceptos semánticos y sintácticos.

Los instructores algunas veces tienen dificultades para comprender los problemas de los estudiantes, debido a que dichos instructores comprenden y procesan de forma exitosa la información semántica por lo que sólo tiene que considerar la información sintáctica, entonces a veces encuentran difícil explicar los conceptos semánticos a los programadores novatos de tal forma que ellos los puedan comprender al mismo nivel.

Cuando los administradores tienen que tomar decisiones sobre quién debe estar involucrado en un proyecto de largo plazo, deben considerar la habilidad para resolver el problema completo y la experiencia en el dominio, en lugar de las habilidades específicas como puede ser un lenguaje de programación.

Las habilidades en el lenguaje de programación son necesarias y lleva tiempo desarrollarlas, sin embargo es mucho más fácil aprender un lenguaje de programación específico que desarrollar habilidades en la resolución de problemas.

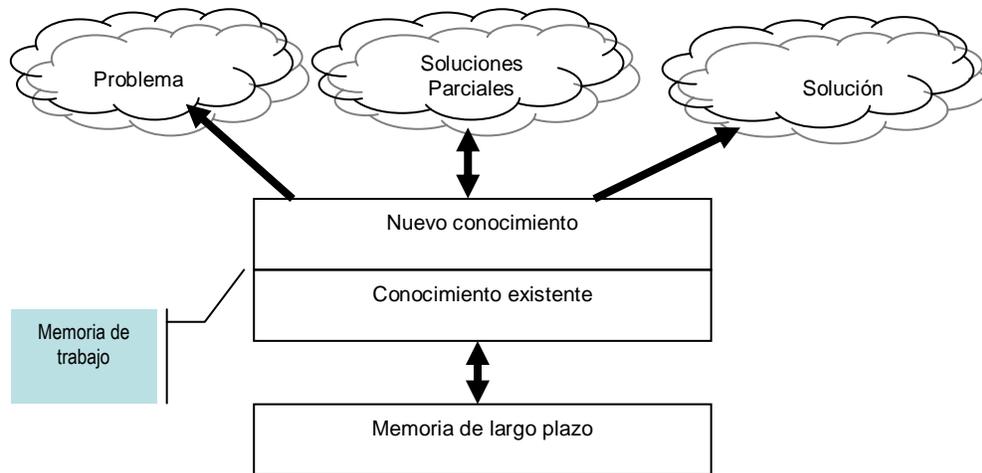


Figura 2.4.2 Proceso de adquisición de conocimiento

Motivación

Maslow señaló que las personas se motivan al satisfacer sus necesidades y éstas se ordenan en una serie de niveles.

Las necesidades de estima se refieren a la necesidades de sentirse respetados por los otros y las necesidades de autorrealización se refieren al desarrollo personal. Las necesidades sociales se refieren a la necesidad de sentirse parte de un grupo, las prioridades humanas son satisfacer las necesidades de los niveles inferiores, como estar hambriento, antes de satisfacer las necesidades más abstractas de los niveles superiores.

En una organización de desarrollo de software es más importante satisfacer necesidades sociales, de estima y de autorrealización desde un punto de vista administrativo. Satisfacer necesidades sociales significa permitir que la gente tenga tiempo para conocer a sus compañeros de trabajo y proveer lugares para que se conozcan, por ejemplo canales de comunicación informales como el correo electrónico o el Chat.

Para satisfacer las necesidades de estima se necesita mostrar a las personas que son de gran valor para la organización. El reconocimiento público de los logros es una forma sencilla pero efectiva de hacer esto. Obviamente, las personas también deben sentir que se les paga de acuerdo con el nivel que reflejan sus habilidades y experiencia.

A partir de un estudio psicológico creado por Bass y Dunteman existen tres tipos de profesionales:

- a) Orientados a tareas, ellos trabajan de forma individual y están motivados por el trabajo que hacen y por el reto intelectual del desarrollo del software.
- b) Orientados a sí mismos. Realizan trabajo individual y están motivados por los éxitos y el reconocimiento personal, se valen de la ingeniería del software para cumplir sus propios propósitos.
- c) Orientados a la interacción. Trabajan en grupo y están motivados por la presencia y acciones de sus compañeros de trabajo, debido a que el desarrollo del software está cada vez más centrado en el usuario, los individuos orientados a la interacción están cada vez más involucrados en la ingeniería del software.

El trabajo en grupo por regla general no debe tener más de ocho personas, la motivación se alimenta de los éxitos del grupo y debe tener el balance correcto de habilidad y experiencia técnica así como un balance de personalidades. Un buen grupo tiene un espíritu de equipo de tal forma que las personas involucradas buscan siempre el éxito en la tarea encomendada.

Existen varios factores que influyen en el trabajo en grupo: composición, cohesión, comunicación y organización.

Las personalidades del grupo deben ser complementarias, más que tomar en cuenta sólo una actividad, por lo que el administrador de proyectos tiene que medir o controlar al equipo de tal forma que los objetivos individuales no trasciendan sobre los objetivos organizacionales.

Dicho control es mejor hacerlo cuando se hace partícipe a todos los miembros del equipo en el proyecto, por lo que es importante designar un líder, aunque éste no sea el más carismático dentro del grupo, sino que cuente con la capacidad de proveer dirección técnica y administración del proyecto. Es posible que internamente dentro del grupo exista otro líder con carisma que apoye al grupo y sea el portavoz de inconformidades del mismo.

Es importante cuidar el no imponer un líder de proyecto, dado que se produce tensión, rechazo y no habrá lealtad del grupo a favor de las metas individuales.

En un grupo cohesivo los miembros creen que el grupo es más importante que los individuos que lo conforman. Los miembros de un grupo cohesivo son leales, se identifican con las metas, protegen al grupo como una entidad de las interferencias exteriores, pueden desarrollar un estándar de calidad, pueden conocer el trabajo de los demás y practicar la programación sin egocentrismos.

Sin embargo los grupos fuertes y cohesivos padecen de dos problemas importantes: resistencia irracional al cambio del liderazgo y pensamiento de grupo mayoritario es decir, lo que diga la mayoría se hará.

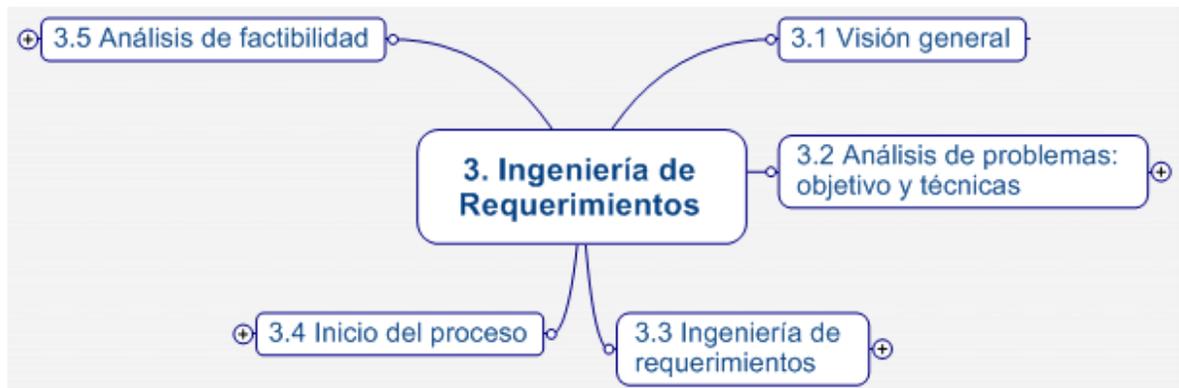
Unidad 3. Ingeniería de requerimientos

Tema 3.1 Visión general

Objetivo Particular

Al término de la unidad, el alumno reconocerá la importancia de realizar un análisis apropiado del problema para clasificar y delimitar los requerimientos del usuario, mediante el uso de diferentes técnicas de recopilación de información.

Mapa conceptual

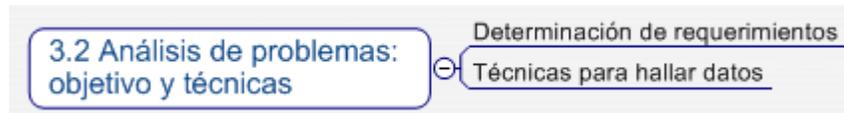


Tema 3.2 Análisis de problemas: objetivo del análisis y técnicas para obtener información

Objetivo específico

Al final del tema, el alumno utilizará herramientas para recopilar información, con el fin de investigar la forma apropiada de dar solución a las diversas necesidades que un usuario tiene, y establecer una o varias propuestas factibles.

Mapa conceptual



¿Qué es la determinación de requerimientos?

Cuando se solicitan formalmente los proyectos, los analistas bajo la dirección de la gerencia llevan a cabo una investigación preliminar para conocer la razón de la petición y recopilar suficiente información que responda a la solicitud en forma apropiada. Algunos proyectos son factibles, mientras que otros no lo son por varias razones. Sin embargo algunas de las razones más comunes para iniciar un proyecto son:

- Mayor velocidad en el proceso
- Mayor exactitud y mejor consistencia
- Consulta más rápida de la información
- Integración de las áreas de negocio
- Reducción de costos
- Mayor seguridad
- Dichas razones son solicitadas por diversas instancias o entidades:
- Gerentes de departamento
- Ejecutivos de alto nivel
- Analistas de sistemas
- Grupos externos

Sin embargo la decisión de aceptar o rechazar una petición no la realizan en su totalidad un analista de sistemas, sino un grupo de personas, generalmente denominado Comité directivo o comité de sistemas de información o comité de grupo de usuarios.

Recuerde que el ciclo de vida del desarrollo de sistemas, consiste en las siguientes actividades o fases con los siguientes recursos:

Cuadro 3.2.1 Porcentaje de recursos utilizados de forma ideal

Fases	Recursos (%)
Análisis	24
Diseño	26
Construcción	30
Implantación	13
Soporte del sistema en producción (Mantenimiento)	7

- **Análisis.** Se intensifica en el software, entendiendo la necesidad de programas, comportamiento, rendimiento e interconexión. Los datos que el analista recaba durante las investigaciones preliminares se recopilan por medio de dos métodos principales: la revisión de documentos y la entrevista de personal escogido de la compañía. Ésta recopilación de datos inicial, también permite ver la factibilidad del proyecto, es decir la posibilidad de que el sistema sea o no benéfico, tanto en el aspecto técnico, operativo y financiero.
- **Diseño.** Es la estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental (algoritmos).
- **Generación de código.** En base al diseño detallado la generación de código se realiza mecánicamente.
- **Pruebas.** Se centra en procesos lógicos internos y procesos externos funcionales. Las entradas definidas producen los resultados esperados.

Por lo tanto podemos decir que un **requerimiento** es una característica que debe incluirse en un nuevo sistema y puede consistir en una forma de capturar o procesar datos, producir información, controlar una actividad de negocio o dar apoyo a la gerencia; por lo tanto la determinación de requerimientos significa estudiar un sistema existente y recopilar los datos en relación con éste para encontrar cuáles son los requerimientos.

Técnicas para hallar datos³

Entrevista. La entrevista se utiliza para recabar información en forma verbal, a través de preguntas que propone el analista. La entrevista es una forma de conversación, no de interrogatorio. La estructura de las entrevistas varía. Si el objetivo de la entrevista radica en adquirir información general, es conveniente elaborar una serie de preguntas sin estructura, con una sesión de preguntas y respuestas libres, con el fin de crear una atmósfera abierta y de fácil flujo de información y que de una u otra forma se conozca actitudes, creencias o ideas de quien responde. La entrevista estructurada utiliza preguntas estandarizadas, y el formato de las respuestas puede ser abierto o cerrado.

³ Algunas de las técnicas aquí mostradas para hallar datos, se ejemplifican en el CD (Caso Práctico), p.10

Cuadro 3.2.2 Ventajas en el uso de entrevistas

Entrevista estructurada	Entrevista no estructurada
Asegura la elaboración uniforme de las preguntas para todo los que van a responder	El entrevistador tiene mayor flexibilidad al realizar las preguntas adecuadas a quien responde
Fácil de administrar y evaluar	El entrevistador puede explotar áreas que surgen espontáneamente durante la entrevista
Evaluación más objetiva tanto de quienes responden como de las respuestas a preguntas	Puede producir información sobre áreas que se minimizaron en las que no se pensó que fueran importantes
Se necesita un limitado entrenamiento del entrevistador	
Resulta en entrevistas pequeñas	

Cuadro 3.2.3 Desventajas en el uso de entrevistas

Entrevista estructurada	Entrevista no estructurada
Alto costo de preparación	Puede utilizarse negativamente el tiempo, tanto de quien responde como del entrevistador
Los que responden pueden no aceptar un alto nivel en la estructura y carácter mecánico de las preguntas	Los entrevistadores pueden introducir sus sesgos en las preguntas o al informar de los resultados
Un alto nivel de la estructura puede no ser adecuado para todas las situaciones	Puede recopilarse información extraña
El alto nivel en la estructura reduce responder en forma espontánea, así como la habilidad del entrevistador para continuar con comentarios hacia el entrevistador	El análisis y la interpretación de los resultados puede ser largo
	Toma tiempo extra recabar los hechos esenciales

Cuestionario. Los cuestionarios proporcionan una alternativa muy útil para las entrevistas. Para los analistas los cuestionarios pueden ser la única forma posible de relacionarse con un gran número de personas para conocer varios aspectos del sistema. Cuando se llevan a cabo largos estudios en varios departamentos, se pueden distribuir los cuestionarios a todas las personas apropiadas para recabar los hechos en relación al sistema. En la mayor parte de los casos, el analista no verá a los que responden, no obstante, también es un ventaja porque aplicar muchas entrevistas permite dar anonimato y las respuestas pueden ser más honestas. Por otra parte aunque su aplicación puede realizarse con un mayor número de individuos, es muy rara una respuesta total. Se pueden reunir grupos más grandes sólo para resolver un cuestionario, las tasas de respuestas pueden mejorar, aunque rara vez se tendrá una respuesta total.

Cuestionario abierto. Al igual que las entrevistas, los cuestionarios pueden ser abiertos y se aplican cuando se quieren conocer los sentimientos, opiniones, y experiencias generales, también son útiles al explorar el problema básico

Cuestionario cerrado. El cuestionario cerrado limita las respuestas posibles del interrogado. Por medio de un cuidadoso estilo en la pregunta, el analista puede controlar el marco de referencia. Este formato es el mejor método para obtener información sobre los hechos.

Observación. La observación proporciona información de primera mano en relación con la forma en que se llevan a cabo las actividades del sistema. Las preguntas sobre el uso de documentos, la manera en la que se realizan las tareas y si ocurren los pasos específicos como se preestablecieron, pueden contestarse rápidamente si se observan las operaciones.

La observación es un arte por sí misma. Saber qué buscar y cómo guiar su significado, también requiere de experiencia. Los observadores con experiencia captan quién utiliza los documentos o registros que no se utilizan. También están alertas para detectar documentos y si encuentran dificultades.

Observar las operaciones del negocio, ya sea a nivel de transacciones o en situaciones de toma de decisiones, tiene sus riesgos, debido a que el simple hecho de que alguien observe puede cambiar la forma en que el trabajo se lleva a cabo; esto es normal, pero en caso de que suceda, los analistas desarrollaran una perspectiva distorsionada en relación con el seguimiento de los procedimientos preestablecidos.

Algunas veces la naturaleza del trabajo es tal, que los analistas no pueden comprender totalmente lo que sucede o entender como realizar un trabajo sin conocerlo. En la observación participativa, los analistas se convierten en parte de la situación y hacen el trabajo por sí mismos, lo cual proporcionara información donde el analista se encontrará en la libertad de analizar alguna situación y ser objetivos en cuanto a lo que se ha experimentado.

Documentos. Los documentos se refieren a los manuales escritos sobre políticas, regulaciones y procedimientos de operaciones estándares que la mayoría de las empresas mantiene como guía para gerentes o empleados

Los documentos y formas que se utilizan a través del sistema pueden proporcionar conocimientos al analista sobre el sistema. Puede ser muy revelador muestrear algunos de los documentos o formas utilizadas en el proceso, con el fin de observar que hay y qué falta. Algunos analistas catalogan los documentos a fin de determinar cuales se utilizan y cuándo se utilizan. Este esfuerzo extra puede ser difícil y llevar mucho tiempo si son demasiados documentos, pero establecer un diagrama de procedimientos y flujos puede resultar de gran utilidad, para lograr una visión integral de todas las tareas.

Tampoco deben pasarse por alto los informes de estudios previos, los resúmenes de consultores y los informes de gerentes, ya que pueden representar una historia que explique el mecanismo de los procedimientos.

Prototipos. Requiere de conocimientos del ciclo de vida de sistemas. Busca reacciones iniciales del usuario, innovaciones, sugerencias del usuario, planes de revisión del sistema propuesto, indican sugerencias en cambios para el proyecto final.

Los prototipos son parchados que trabajan de forma no eficiente ni elegante. Un prototipo de características seleccionadas permite que el sistema sea puesto en marcha mientras otras características pueden ser añadidas en fecha posterior.

Los prototipos generan un esfuerzo doble y se corre el riesgo de adoptarlo como el sistema real, sin embargo mediante un prototipo existe el potencial de realizar cambios en el sistema real, se puede detener el desarrollo para encontrar fallas con menos costo y se pueden definir las necesidades del usuario y lo que se espera de él.

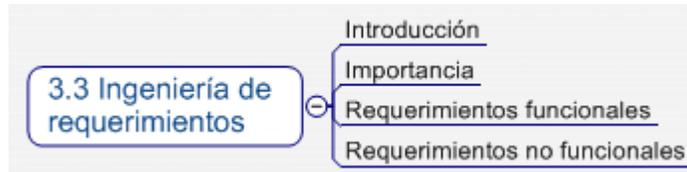
Tema 3.3 Ingeniería de Requerimientos: tipos de requerimientos y notaciones de especificación

“Por lo general las semillas de los desastres más importantes en software se siembran en los primeros tres meses desde el comienzo del proyecto”, Capers Jones

Objetivo específico

Al final del tema, el alumno identificará los requerimientos funcionales de los no funcionales, y la importancia que tienen en el diseño de los sistemas de información, mediante ejemplos diversos expuestos por el propio alumno.

Mapa conceptual



Introducción

La fase del Análisis, de acuerdo con el ciclo de vida del sistema se enfoca totalmente al intento de seccionar un posible problema en las partes que lo conforman para intentar comprenderlo, así como desarrollar soluciones de carácter general que fuesen aplicables.

Un análisis efectivo es una etapa crítica, verdaderamente importante para el desarrollo de futuras etapas en el ciclo del sistema, pero es común que se destine a ello un volumen de recursos reducidos, ya que la distribución de recursos (humanos, materiales, financieros) tiende hacia labores cuyos resultados sean visibles e inmediatamente medibles.

Alrededor del objeto del análisis, se encuentran siempre las personas que requieren su solución o que se ven afectados directa o indirectamente por su existencia llamados usualmente USUARIOS, y el conocimiento de ellos, de sus actividades y sus puntos de vista acerca del problema, es requisito a cubrirse a la brevedad posible, sin perder nunca la posición que debe cubrir el analista de sistemas frente y/o junto al usuario.

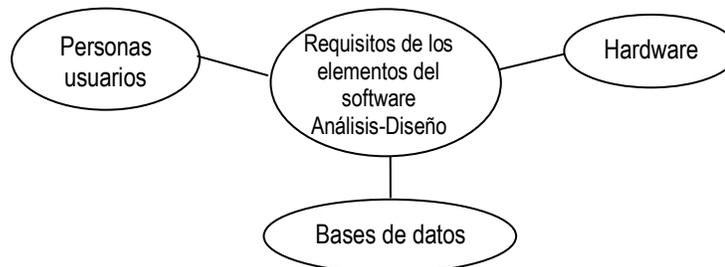


Figura 3.3.1 Relaciones implícitas del análisis y diseño de necesidades

Importancia de la Ingeniería de Requisitos

La Ingeniería de requisitos ayuda al Ingeniero de Software a entender el problema, comprender cual será el impacto del software sobre el negocio. Sería deshonesto decir que la Ingeniería de requisitos es la solución para los retos que se han enunciado. Pero proporciona un enfoque sólido para abordar dichos desafíos.

Muchos desarrolladores argumentan que las cosas se aclaran mientras se construye el software, que los clientes serán capaces de entender mejor las necesidades solo después de examinar las primeras iteraciones del software, que las cosas cambian tan rápido que la ingeniería de requisitos es una pérdida de tiempo, que la línea de base produce un programa que funciona y todo lo demás es secundario.

Lo que hace seductores a estos argumentos es que contienen elementos de verdad, en particular esto es cierto para los proyectos chicos (menos de un mes) que implica un esfuerzo relativamente pequeño. Conforme el software crece en tamaño y complejidad estos argumentos comienzan a derrumbarse.

Es esencial que el equipo de software haga un esfuerzo real por entender los requisitos de un problema antes de intentar resolverlo.

La ingeniería de requisitos es un puente que comienza al pie de los participantes del proyecto (gerentes, clientes, usuarios finales) quienes definen las necesidades del negocio los diversos escenarios, características, funciones y restricciones. La parte alta del proyecto es cuando el equipo de software examina el contexto del trabajo, necesidades específicas del diseño y construcción, prioridades que permitan completar el trabajo, funciones y comportamientos que reflejen el impacto del diseño.

Por tal motivo es necesario que los desarrolladores de software tomen en cuenta los principios del análisis, ya que como se menciona anteriormente, esta fase es una etapa crítica que permitirá en determinado momento diseñar y elaborar lo que el cliente o usuario desea encontrar para la solución de problemas de automatización o incluso de logística en procesos de cualquier índole. A continuación se enumeran los principios del análisis:

1. Debe representarse y entenderse el dominio de la información del problema
2. Deben definirse las funciones que deben realizar el software
3. Debe representarse el comportamiento del software, como consecuencia de acontecimientos externos
4. Deben dividirse los modelos que representan información, función y comportamiento de manera que descubran los detalles por capas o jerárquicamente.
5. El proceso de análisis debería ir desde la información esencial hasta el detalle de la implementación.

Por otra parte el proceso de la Ingeniería de requisitos se lleva a cabo a través de siete distintas funciones:

Inicio. Al inicio del proyecto se hacen una serie preguntas libres de contexto, las cuales deben de tener por objetivo establecer una comprensión básica del problema, las personas que quieren una solución, la naturaleza de la solución y la efectividad de la comunicación preliminar entre el cliente y el desarrollador.

Elaboración. La elaboración es una acción del modelado del análisis y se compone de una serie de tareas de modelado y refinamiento. El resultado final de la elaboración es un modelo de análisis que define el dominio de la información, las funciones y el

comportamiento del problema.

Negociación. Dados los recursos limitados del negocio, no resulta inusual que los clientes y usuarios pidan más de lo que se puede lograr. El Ingeniero de requisitos debe conciliar estos conflictos por medio del proceso de negociación. Se solicita a los clientes, usuarios y otros interesados que ordenen sus requisitos y después discutan los conflictos relacionados con la prioridad.

Mediante un enfoque iterativo, los requisitos se eliminan o modifican de forma que cada parte alcance cierto grado de satisfacción.

Especificación. En el contexto de los sistemas basados en computadoras, éste termino tiene varios significados (documento escrito, modelos gráficos, modelo matemático formal, escenarios de uso, prototipo o una combinación de estos). Sin embargo en esta proceso se describen las funciones y el desempeño de un sistema basado en computadora y las restricciones que tenga el desarrollo.

Validación. La validación de requisitos examina la especificación para asegurar que todos los requisitos de software se han cumplido, y al mismo tiempo dar a conocer inconsistencias, omisiones, errores y la corrección correspondiente. De esta manera el proceso de validación debe asegurar que los productos de trabajo cumplen con los estándares establecidos para el proceso, producto y proyecto.

Gestión. La gestión de requisitos es un conjunto de actividades que ayudan al equipo de proyecto a identificar, controlar y rastrear los requisitos y los cambios a éstos en cualquier momento, mientras se desarrolla el proyecto.

Cuando un sistema es grande y complejo la determinación de las conexiones entre los requisitos puede ser una tarea redituable., por lo que se recomienda el uso de tablas de rastreabilidad para facilitar un poco el trabajo.

Requisito	Aspecto específico del sistema o su ambiente					
	A01	A02	A03	A04	A05	A06
R01	✓	✓		✓	✓	✓
R02		✓	✓		✓	
R03	✓		✓	✓		✓
R04			✓			✓
R05		✓			✓	✓
R06	✓		✓	✓		

Figura 3.2.2 Tabla de rastreabilidad

Entre algunas tablas de rastreabilidad son:

- Tabla de rastreabilidad de las características (características observables para el cliente)
- Tabla de rastreabilidad de la fuente (de los requisitos)
- Tabla de rastreabilidad del subsistema
- Tabla de rastreabilidad de la interfaz (interfaces internas y externas)

Requerimientos funcionales y no funcionales

La ingeniería de requerimientos (especificaciones), involucra procesos para descubrir, analizar, documentar y verificar los servicios y restricciones que cualquier software debe considerar. El **dominio** son los requerimientos que provienen del dominio de aplicación del sistema, y que reflejan las características de ese dominio. Este puede ser funcional o no funcional.

Requerimientos Funcionales.

Son declaraciones de los servicios que proveerá el sistema de manera que éste reaccione a entradas particulares, ¿Cómo se comportará en situaciones particulares? Describen la funcionalidad o los servicios que se espera que el sistema genere. Éstos dependen del tipo de software y del sistema que se desarrolle y de los posibles usuarios del software. Los requerimientos funcionales describen con detalle la función del sistema definidas en entradas, salidas, excepciones, etc.

Por ejemplo: Un sistema bibliotecario para estudiantes y académicos

- **El usuario** debe tener la posibilidad de buscar, en el conjunto inicial de la base de datos o seleccionar un subconjunto de ella por tema, autor, I.D.
- **El sistema** deberá proveer visores adecuados para leer los libros almacenados
- **El software** debe permitir el manejo de un I.D. identificador único, que cada libro deberá tener con el fin de que el usuario pueda copiar para buscarlo físicamente

Muchos de los problemas de la ingeniería de software provienen de la imprecisión de la especificación de requerimientos funcionales de un sistema, la cual debe estar completa y ser consistente. La completitud significa que todos los servicios solicitados por el usuario estén definidos. La consistencia significa que los requerimientos no tienen definiciones contradictorias.

En la práctica para sistemas grandes o complejos, es imposible cumplir con los requerimientos de consistencia y completitud, lo cual se debe parcialmente a la complejidad inherente al sistema y parcialmente a los diferentes puntos de vista con necesidades inconsistentes.

Requerimientos No funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema, por ejemplo: restricciones de tiempo, sobre el proceso de desarrollo de estándares. Los requerimientos no funcionales no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

De forma alternativa, los requerimientos no funcionales definen las restricciones del sistema, capacidad de los dispositivos de entrada/salida, representación de datos que se utilizan en las interfaces.

Así, mientras que el incumplimiento de un requerimiento funcional degradará al sistema, el incumplimiento de un requerimiento no funcional lo inutilizará.

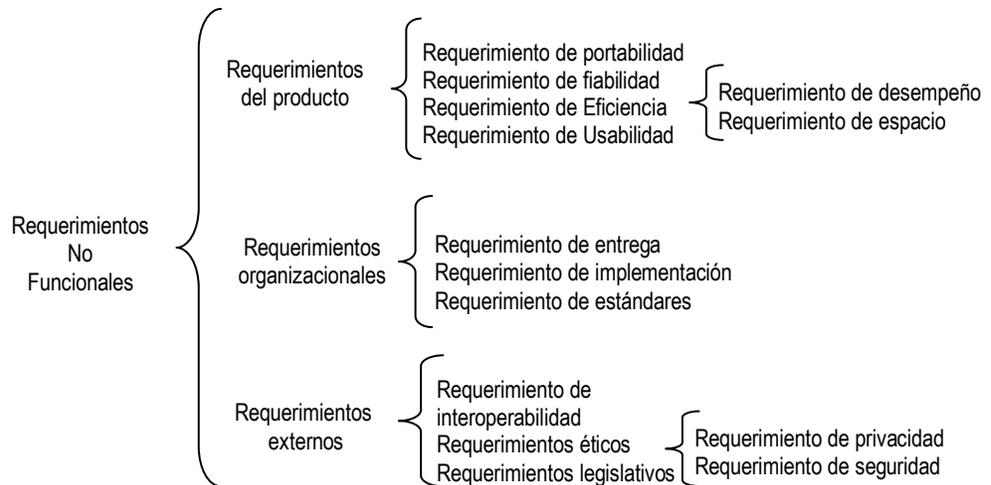
Por ejemplo: Si un sistema de vuelo no cumple los requerimientos de fiabilidad no se certificará como seguro para su operación. Si un sistema de control de tiempo real no cumple los requerimientos de desempeño, las funciones de control no operará correctamente.

Los requerimientos no funcionales surgen de las necesidades del usuario debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware o a factores externos como los reglamentos de seguridad, políticas de privacidad, etc.

Los requerimientos de producto especifican comportamiento del producto. Ejemplo: requerimientos de desempeño en la rapidez de ejecución del sistema, fiabilidad que fijan la tasa de fallas para que el sistema sea aceptable portabilidad y usabilidad

Los requerimientos organizacionales se derivan de las políticas y procedimientos existentes en la organización del cliente y del desarrollador, por ejemplo: estándares en procesos, requerimientos de implementación como lenguajes de programación, método de diseño, requerimientos de entrega que especifican cuando se entregará el producto y su documentación.

Los requerimientos externos cubre todos los requerimientos de interoperabilidad que definen la manera en que el sistema interactúa con los otros sistemas de la organización, tales como requerimientos legales que deben seguirse para asegurar que el sistema opere dentro de la ley y requerimientos éticos.



Algunas métricas para especificar requerimientos no funcionales son

- Rapidez: ya que nos permite conocer el número de transacciones procesadas por segundo, tiempo de respuesta al usuario y a eventos, tiempo de actualización de la pantalla, entre otros.
- Tamaño: es importante reconocer y conocer cuales será la demanda diaria de espacio disponible en el disco duro o en el dispositivo de almacenamiento que se utilizará con mayor frecuencia, con el fin de tomar las medidas correspondientes para no colapsar en momentos críticos(KB, Número de chips en RAM)

- Facilidad de uso: Es indudable que cualquier usuario busca en la solución computacional, la facilidad de administrar la información, es decir, de acceder, de capturar, de consultar de generar los reportes deseados invirtiendo el mínimo esfuerzo y mínimo de tiempo. Y en determinado momento que el sistema pueda responder mediante cuadros de ayuda para el usuario y salir adelante con los procesos de información.

La siguiente frase describe mucho de lo que el usuario y los desarrolladores de software esperan del análisis de sistemas de información, etapa crítica en el desarrollo de software: “A menudo las expectativas fallan y entre más lo hacen más prometen”, William Shakespeare

Por otra parte la QFD(despliegue de la función de calidad, por sus siglas en inglés), es una técnica que traduce las necesidades del cliente en requisitos técnicos para el software, resaltando lo que es mejor para el cliente. La QFD identifica tres tipos de requisitos:

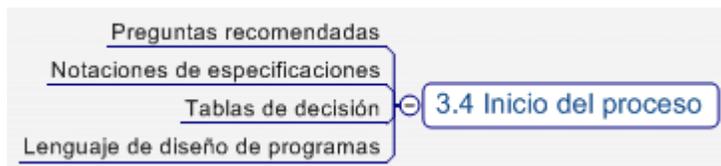
- Requisitos normales. Refleja las metas y objetivos establecidas por el cliente resultado de varias reuniones.
- Requisitos específicos. Son los implícitos por el producto o sistema y pueden parecer obvios. Requerimientos no funcionales
- Requisitos estimulantes. Reflejan características que van más allá de las expectativas del cliente y que prueban ser muy satisfactorias cuando están presentes.

Tema 3.4 Inicio del Proceso

Objetivo Particular

Al término del tema, el alumno elaborará su primeras preguntas y las pondrá en práctica para reconocer la importancia del análisis de requerimientos utilizando las herramientas apropiadas.

Mapa conceptual



Preguntas recomendadas

“El que pregunta es un tonto por cinco minutos, el que no pregunta es un tonto por siempre”, Proverbio chino.

Ahora bien el proceso de inicio se compara con la primera cita entre dos adolescentes, porque

- Nadie sabe que decir o preguntar
- Los dos están preocupados de no haber malentendidos
- Pueden tener expectativas diferentes ¿Qué pasará?
- Los dos quieren que termine rápido, pero que la cita sea un éxito

Entonces el primer bloque de preguntas son libres de contexto, las cuales tienen por objetivo identificar a todos los participantes en el proyecto, identificar el beneficio medible de una implementación exitosa y las alternativas posibles:

- ¿Quién está detrás de esta solicitud?
- ¿Quién usará la solución?
- ¿Cuál será el beneficio económico de una solución exitosa?
- ¿Existe otra fuente para la solución requerida?

La siguiente serie de preguntas permite al equipo de software comprender mejor el problema y deja que el cliente exprese sus propias percepciones.

- ¿Cómo podría caracterizarse un buen resultado generado por una solución exitosa?
- ¿Cuáles problemas deberían atacar esta solución?
- ¿Podría usted describir o mostrar el ambiente de negocios en el que se utilizará la solución?
- ¿Los aspectos especiales del desempeño o las restricciones afectarán la forma en que se busque la solución?

El último bloque de preguntas permite enfocar la efectividad de la actividad de la comunicación en sí misma. Algunos autores (Gause y Weinberg) las llaman metapreguntas:

- ¿Es usted la persona adecuada para el contestar estas preguntas? ¿Sus respuestas son oficiales?
- ¿Mis preguntas son relevantes para su problema?
- ¿Estoy haciendo demasiadas preguntas?
- ¿Alguien más puede proporcionar más información?
- ¿Debería preguntarle otra cosa?

Por otra parte, la **fase del Análisis** se puede trabajar en tres fases:

1. Identificación del problema. Análisis de la situación actual (Investigación preliminar). Técnicas para hallar datos
2. Requerimientos de procesamiento de información. Identificación de las necesidades de información. Definición del problema
3. Análisis preliminar (Propuesta general de solución)

Algunas preguntas claves que nos apoyaran para esta fase, y principalmente en la identificación del problema, pueden ser:

- ¿Qué se está haciendo?
- ¿Cómo se esta haciendo?
- ¿Cuándo ocurre?
- ¿Dónde ocurre?
- ¿Por qué ocurre?
- ¿Quién lo hace?
- ¿Qué tan serio es el problema?
- ¿Cuál es la causa principal?

Una vez identificado el problema, será necesario conocer de forma más específica los **requerimientos de transacciones de los usuarios**, es decir, los sistemas del nivel de transacción capturan, procesan y almacenan los datos. Por ejemplo los analistas asignados para trabajar en un sistema de entrada de pedidos desean saber más acerca de cómo se procesan éstas transacciones, y posiblemente harán preguntas como las siguientes, para identificar el proceso:

- ¿Cuál es el propósito de esta actividad?
- ¿Cuáles son los pasos que se realizan?
- ¿Dónde se realizan?
- ¿Quién los ejecuta?
- ¿Cuánto tiempo consumen?
- ¿Con qué frecuencia se realizan?
- ¿Quién utiliza la información resultante?

De forma particular algunas preguntas para sistemas administrativos, pueden ser:

- ¿Cuáles son los límites impuestos por tiempo y cantidad de trabajo?
- ¿Qué controles de rendimiento se utilizan?
- ¿Qué conforma la transacción que se esta procesando?
- ¿Qué inicia la transacción?
- ¿Quién inicia realmente el pedido?, ¿Para que propósito?
- ¿Con que frecuencia ocurren los pedidos?
- ¿Qué cantidad se asocia con cada uno?
- ¿Existen condiciones diferentes que puedan afectar la forma en la que los pedidos son procesados?
- ¿Qué datos necesitan para procesar la transacción?
- ¿Qué información se genera? ¿qué datos se almacenan?

Sin embargo todo sistema informático utiliza requerimientos de decisión de los usuarios, ya

que las decisiones a diferencia de las actividades de transacción, pueden no seguir un procedimiento específico, debido a que las rutinas no son tan claras y los controles pueden ser muy vagos. Por esta razón el analista que esté investigando los sistemas de apoyo para la toma de decisiones debe realizar las mismas preguntas en los aspectos de tiempo y frecuencia ya estudiados. También debe considerar las siguientes preguntas:

- ¿Qué información se utiliza para tomar la decisión?
- ¿Cuál es la fuente de la información?
- ¿Qué sistemas de transacciones producen los datos utilizados en el proceso de decisión?
- ¿Qué datos del procesamiento de transacciones se requieren, pero no resultan del procesamiento mismo?
- ¿Cómo deben procesarse los datos para producir la información necesaria?
- ¿Cómo debe presentarse la información?

Finalmente y **para reflexionar**, se puede pensar que realizar una serie de preguntas parece simple, sin embargo no es así, algunos motivos que lo hacen difícil son:

Problemas de ámbito. El límite del sistema está mas definido o los clientes/usuarios especifican detalles técnicos innecesarios que pueden confundir en lugar de clarificar, los objetivos generales del sistema.

Problemas de comprensión. Los clientes/usuarios no estan seguros por completo de que lo que se necesita, comprende poco acerca de las capacidades y limitaciones de su ambiente de cómputo, no comprenden del todo el dominio del problema, tienen dificultades al comunicar necesidades al Ingeniero de Software. Omiten información que parece obvia, especifican necesidades que chocan o se contraponen con otros clientes/usuarios, especifican requisitos que son ambiguos o inestables.

Problema de volatilidad. Los problemas cambian conforme transcurre el tiempo.

Así cada uno de estos problemas se minimizará si existe una forma organizada de llevar todas las actividades para la recopilación de requisitos.

Notaciones de especificaciones

Tablas de decisión

Una tabla de decisiones son instrumentos importantes en el análisis de las decisiones estructuradas. Una gran ventaja es que asegura la integridad, es decir, evita excluir alguna condición. También es fácil verificar posibles errores, como situaciones imposibles, contradicciones o redundancias.

1. Hacer una lista de todas las acciones que pueden asociarse con un procedimiento específico (o módulo)
2. Hacer una lista de todas las condiciones (o decisiones tomadas), durante la ejecución del procedimiento
3. Asociar conjuntos específicos de condiciones con acciones específicas eliminando combinaciones imposibles de condiciones, alternativamente, desarrollar cualquier permutación posible de combinaciones
4. Definir reglas indicando que acción o acciones ocurren para un conjunto de condiciones

Cuadro 3.4.1 Estructura de tabla de decisión

Número de reglas	1	2	3	4	5	...	N
Filas de condiciones	Alternativas de condición						
Filas de acciones	Registro de las acciones						

Cuadro 3.4.2 Ejemplo "Pagar una compra"

Condiciones y acciones	1	2	3	4
Menos de \$50	S	S	N	N
Pagos por cheque con dos formas de identificación	S	N	S	N
Uso de tarjeta de crédito	N	S	N	S
Registrar la venta	X			
Verificar la tarjeta de crédito en el boletín		X		
Llamar al supervisor para autorización			X	
Llamar al banco para la autorización de la tarjeta de crédito				X

Lenguaje de diseño de programas (PDL's o LDP)

Un texto explicativo del procesamiento es idealmente una delimitada descripción sin ambigüedades del procesamiento que ocurre dentro de un módulo. La narrativa describe el procesamiento, las tareas, las decisiones y la entrada/salida

También se documentan las restricciones/limitaciones de cada módulo tales como: tipo o formato de datos, limitaciones de memoria o de tiempo, delimitar los valores o cantidades de las estructuras de datos, casos especiales no considerados, características específicas de un módulo. Por tal motivo es importante que todas las especificaciones sigan un formato estándar. Algunas recomendaciones son:

1. Una sintaxis fija de palabras clave para todas las construcciones estructuradas, declaraciones
2. Sintaxis libre en lenguaje natural que describa las características del procesamiento
3. Facilidades para la declaración de datos que deberían incluir tanto estructuras de datos simples (escalar, matriz) y complejas (listas enlazadas o árboles)
4. Definición de subprograma y técnicas de invocación que soporten varios modos de descripción de interfaz.
5. Algunos lenguajes estructurados que siguen estas normas son ADA, C y Cobol

Árboles de decisión

Un árbol de decisión identifica de manera inmediata el orden de verificación de las condiciones y las acciones que se deben llevar a cabo.

Las condiciones y acciones se encuentran en ciertas ramas no en todas. Aquéllas que son decisivas se conectan de forma directa con otras condiciones y acciones, mientras las no importantes no se incluyen. Son más apropiados como método de comunicación que una tabla.

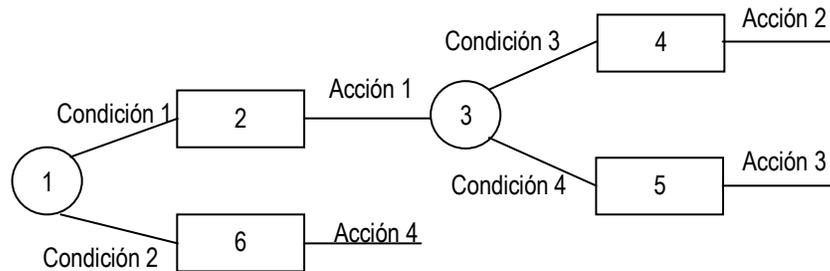


Figura 3.4.1 Estructura de árbol de decisión

Lenguaje Natural Estructurado

Se emplean plantillas para especificar requerimientos en donde se debe incluir:

- Una descripción de la función o entidad a especificar
- Una descripción de sus entradas a especificar
- Una descripción de sus salidas y hacia donde van
- Una indicación de qué otras entidades utiliza la función descrita
- Si se utiliza un enfoque funcional, una precondición que indique lo que debe cumplirse antes de que la función sea invocada y una postcondición que especifique lo que será verdad después de que dicha función se haya invocado.
- Una descripción de los efectos colaterales (si existen) de la operación

Lenguaje de Descripción de Programas

Es un lenguaje derivado de una programación en Java o ADA, el cual contiene construcciones abstractas para incrementar su poder de expresividad. El uso de PDL's conduce a especificaciones detalladas y algunas veces son muy cercanas a las implementaciones para que sean incluidas en un documento de requerimientos.

Sin embargo su utilización se recomienda:

- Cuando una operación se especifica como una secuencia de acciones simples y el orden de ejecución es importante, por ejemplo un cajero automático.
- Cuando se tienen que especificar las interfaces de hardware y software, lo cual permite definir interfaces entre objetos y tipos.

Existen desventajas en este enfoque:

- El lenguaje utilizado puede no ser suficientemente expresivo para describir una funcionalidad del sistema
- La notación solo es comprensible para la gente con conocimientos en lenguajes de programación

Los requerimientos se consideran un diseño de la especificación del diseño, más que un modelo para ayudar a comprender el sistema, una forma efectiva es combinar ambos enfoques

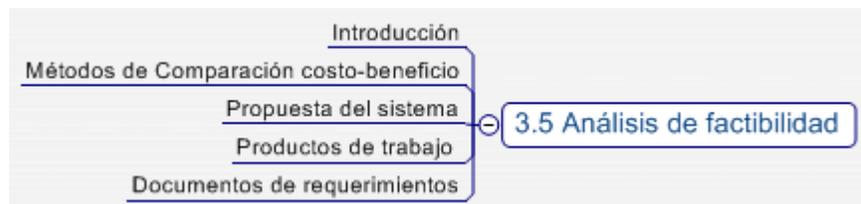
Tema 3.4 Análisis de factibilidad

*“Todos los proyectos son posibles si se tienen infinitos recursos y tiempo”,
Pressman*

Objetivo Particular

Al término del tema, el alumno realizará su propuesta de análisis de factibilidad reflejando el costo-beneficio, alternativas y los requerimientos solicitadas por el usuario, basándose en ejemplos proporcionados por el profesor.

Mapa conceptual



Introducción

A través de un análisis de factibilidad es posible evitar periodos de tiempo invertidos en esfuerzo humano sin resultados, además de invertir miles o millones de pesos o dólares. Así mismo un estudio de factibilidad nos ayuda a evitar bochorno profesional indecible al entregar un sistema mal hecho o con pocas garantías de sobrevivencia.

El análisis de factibilidad permite resolver varias preguntas en primera instancia:

1. ¿El sistema contribuye a los objetivos generales de la organización?
2. ¿El sistema se puede implementar utilizando la tecnología actual y con las restricciones de costo y tiempo?
3. ¿El sistema puede integrarse a otros que existen en la organización?

En una segunda etapa, el análisis de factibilidad permite evaluar la información, mediante las siguientes preguntas:

1. ¿Cómo se las arreglaría la organización si no se lleva a cabo este sistema?
2. ¿Cuáles son los problemas con los procesos actuales y como ayudaría el nuevo sistema a resolverlos?
3. ¿Cuál es la contribución directa que hará el sistema a los objetivos del negocio?
4. ¿La información se puede obtener y transferir a otros sistemas de la organización?
5. ¿El sistema requiere de tecnología que no se ha utilizado previamente en la organización?
6. ¿A que se debe ayudar el sistema y a que no necesita ayuda?

El **objetivo** de este estudio es realizar recomendaciones para el desarrollo del sistema. Se proponen cambios al alcance, el presupuesto y la calendarización del sistema. Se pueden sugerir requerimientos adicionales de alto nivel para los cambios. Las fuentes de información son los administradores, Ingenieros de software, Expertos en tecnología, Usuarios finales.

Esta fase del desarrollo de sistemas se considera subjetiva, por el hecho de contemplar diferentes puntos de vista o criterios de acuerdo a las características propias del proyecto:

- Tamaño del proyecto
- Beneficios esperados
- Calidad del diseño
- Satisfacción del cliente
- Mejores direcciones del negocio (crecimiento empresarial)
- Cantidad de información

Métodos de Comparación Costo - Beneficio

El análisis costo-beneficio también puede referir a Análisis Económico costo-beneficio, cuyo objetivo sea realizar una valoración de la justificación económica para un proyecto de sistema basado en computadora. Por otra parte existen costos tangibles e intangibles que se deben tomar en cuenta en esta fase

Los costos tangibles se pueden proyectar con precisión, incluyen costos de computadoras, terminales, cables, etc. Por otra parte el tiempo ocupado por el analista, programador, y todos los involucrados en el desarrollo de software, también causa costo debido a que a mayor tiempo que pase la implantación de la solución los procesos seguirán generando un costo (Salarios)

Los costos Intangibles son difíciles de estimar, se refiere a cuestiones no físicas, como pérdidas de ventajas competitivas, reputación (innovación), disminución de imagen, por lo tanto la toma de decisión será no efectiva.

Los Beneficios Tangibles, son ventajas medibles, como el incremento de la velocidad de procesamiento, acceso a información anterior y actual. Los beneficios tangibles pueden ser medidos en ponderación de pesos, recursos o tiempo ahorrado

Los beneficios intangibles son difíciles de medir, como el proceso de toma de decisiones, el aumento en la precisión de los resultados, en cuanto a estándares de desempeño en periodo de tiempo relativamente cercano, satisfacción total o parcial del cliente o los usuarios del sistema, son los puntos más importantes a evaluar para buscar la trascendencia del software.

Algunos métodos utilizados para el análisis de costo beneficio nos permiten identificar diversos puntos de vista para analizar el tiempo de recuperación de la inversión, el flujo de efectivo entre otras cosas. A continuación se explicarán algunos de ellos.

Análisis de punto de equilibrio. Se realiza una justificación en costo y no en beneficio, ésta técnica se realiza una gráfica de los costos totales que involucra un sistema actual y el sistema propuesto:

Costos totales= Costos que se repiten durante la operación del sistema + costos de instalación (esta actividad sólo sucede una sola vez)

Ventaja. Este análisis es útil cuando un negocio crece y el volumen de información es una variable importante en el costo.

Desventaja. Los beneficios permanecen iguales, sin importar el sistema que se utilice

Recuperación. Analiza los beneficios tangibles que mejoran el sistema propuesto. Se pone en la disyuntiva de invertir o no en el proyecto, se debe identificar el tiempo de recuperación de la inversión reflejado en las ganancias, los cuales pueden ser hasta años. La desventaja de este método de estimación de costo-beneficio, es que se basa en un enfoque a corto plazo en decisiones de inversión y reemplazo, no toma en cuenta la distribución de pagos y no considera retornos totales del proyecto.

Análisis del flujo de efectivo. Examina dirección tamaño y patrón de flujo de efectivo asociado al sistema de información, con el fin de determinar cuando se tiene ganancias en base al sistema anterior. Solo asocia salidas de dinero, pagos en efectivo y ganancias.

Valor presente. Se refiere al valor del dinero en el tiempo, el flujo de fondos y de inversión.

Métodos comunes. Condicional, correlación, regresión, indicadores principales (parámetros), econometría, modelos de entrada-salida

Estimación de tendencias. Medias móviles, Mínimos cuadrados, juicio gráfico.

El esfuerzo empleado en el análisis de la factibilidad de un proyecto, aunque lleve a la cancelación del mismo, no es un esfuerzo en vano.

Propuesta del Sistema

La planeación de la estructura del software o preparación de la propuesta del sistema considera los siguientes rubros generales, Adquisición de software o hardware y Evaluación del software.

Adquisición de software o hardware

- Es necesario tener contacto con el usuario para determinar el hardware futuro. El analista debe conocer las tareas que debe lograr el sistema de información. A través de entrevistas, cuestionarios e investigación de datos se pueden eliminar supuestos sobre el tipo de hardware y software requerido.
- Es necesario comparar cargas de trabajo entre el sistema existente y el propuesto, con el fin de que el sistema propuesto resista las cargas de trabajo actual y futuras.
- Considere las opciones de adquisición de equipo de cómputo: compra arrendamiento financiero o renta. El arrendamiento financiero de computadoras se recomienda cuando la vida útil del software propuesta en menor a 4 año. Existen esquemas de arrendamiento de computadoras y software desde dos mil pesos mensuales. Otra ventaja, en este caso, es que el gasto también es deducible y puedes renovar tu equipo y programas constantemente.
- ¿Con quién voy a comprar?, Se debe buscar los siguientes beneficios: Soporte técnico en hardware y software, instalación y capacitación, mantenimiento (respuestas inmediatas en situaciones de emergencia, préstamo de equipo mientras se realiza la reparación de la computadora)
- Otras opciones:

- ◆ **Arriendo.** El contrato de arrendamiento tiene como único efecto, el crear derechos de uso por lo general no cedibles a terceros, sobre un bien específico. Por tanto en los contratos de arrendamiento de hardware, no se transfiere la propiedad del bien, pero sí las responsabilidades sobre el mismo, de ahí que sean pocos los contratos que permiten el subarrendamiento, que constituiría la cesión de derechos a terceros. Esta modalidad será conveniente en casos, de no permanencia por mucho tiempo en la actividad para la cual se contrata en arrendamiento el bien informático, o en casos de necesidad de equipo de forma temporal, no necesariamente con todos los adelantos de la tecnología, de tal forma de ahorrar costos para a futuro optar por nueva y probada tecnología.

- ◆ **Leasing.** En el leasing o arrendamiento con opción a compra, "El contratista se obliga a fijar de antemano un valor en venta invariable y a señalar los porcentajes de deducción que está dispuesto a aceptar en función de la cantidad pagada en concepto de canon y del momento del ejercicio de la opción a compra.²¹ Como se puede observar, la transmisión de propiedad del bien, no se realizará, sino hasta que se pague totalmente el precio fijado al equipo, por tanto, generalmente para este tipo de contrato será de esperar que el derecho de propiedad sea cedido a largo plazo. Esta modalidad en los tiempos actuales debe escogerse siempre que los equipos estén acordes a la tecnología de punta imperante, por ejemplo para sistemas abiertos, esta modalidad no presenta riesgos ya que por sus características, ofrece portabilidad, escalabilidad, y los up-grades son fácilmente implementados, de tal manera que al cabo del plazo fijado para el efecto de la compra, los equipos no se encuentren obsoletos. En cualquier caso siempre se deberá añadir en este tipo de contratos, la no-obligatoriedad de la compra del equipo contratado mediante leasing por la Entidad

Evaluación del software

- Efectividad del desempeño. Que realice todas las tareas actuales y futuras, contemplando el crecimiento de la información.
- Eficiencia de desempeño. Tiempo de respuesta, almacenamiento eficiente, entradas, salidas y respaldos eficientes
- Facilidad de uso. Interacción fácil con el usuario, flexible, ayuda en línea, retroalimentación, recuperación de errores.
- Manuales, tutoriales y soporte técnico del fabricante.

Métricas para la eficiencia de la propuesta

- Reducción o eliminación de costos
- Reducción de errores
- Incremento de flexibilidad
- Aumento del ritmo de la actividad
- Mejora en la planificación o control de gestión

Productos de trabajo de la elaboración

Los productos de trabajo producidos como consecuencia de la obtención de requisitos variará de acuerdo con el tamaño del sistema o producto que se vaya a construir.

La mayoría de los trabajos incluyen los siguientes productos:

- Un enunciado de necesidad y factibilidad
- Un enunciado limitado del ámbito del sistema o producto
- Una lista de clientes, usuarios y otros interesados que participaron en la obtención de requisitos
- Una descripción del medio ambiente técnico del sistema
- Una lista de requisitos (de preferencia organizada por función) y las restricciones de dominio aplicables a cada uno
- Un conjunto de escenarios de uso que proporcionen un discernimiento de la utilización del sistema o producto en diferentes condiciones de operación
- Cualesquiera prototipos desarrollados para definir de mejor forma los requisitos.

Documento de requerimientos. Especificaciones

Tiene un conjunto diverso de usuarios que va desde los administradores principales de la organización, quienes pagan por el sistema, hasta los ingenieros responsables del software. Los posibles usuarios de este documento son:

Cuadro 3.4.1 Clasificación de usuarios para el documento de requerimientos

Cientes del sistema	Verifican mediante éste documento que se cumplan las necesidades
Administradores	Planea el proceso de desarrollo
Ingenieros de sistemas	Comprenden porqué se desarrollará el sistema
Ingenieros probadores del sistema	Desarrollaran las pruebas de validación para el sistema
Ingenieros mantenedores del sistema	Le ayudan a comprender el sistema y las relaciones entre las partes.

Este documento busca satisfacer los siguientes puntos:

- Especificar únicamente el comportamiento externo del sistema
- Especificar restricciones de la implementación
- Servir como herramienta de referencia para los mantenedores del sistema (¿será fácil de cambiar?)
- Registrará las previsiones del ciclo de vida del sistema
- Caracterizará las respuestas aceptables para eventos no deseados.

En el caso de que la justificación económica sea obvia (bajo criterios directivos), el riesgo técnico para el desarrollo del sistema sea bajo, no existen consideraciones legales y la alternativa es única, no es necesario un estudio de factibilidad.

Documento de requerimientos

- Introducción
 - Propósito del documento de requerimientos
 - Alcance del producto
 - Definiciones, acrónimos y abreviaturas
 - Referencias
- Descripción general
 - Perspectivas del producto
 - Funciones del producto
 - Características del usuario

- Restricciones generales
- Suposiciones y dependencias
- Requerimientos específicos
 - Requerimientos Funcionales
 - Requerimientos No Funcionales
 - Requerimientos de Interfaz
- Apéndices
- Índices

Requerimientos del usuario

Deben redactarse utilizando el lenguaje natural, representaciones y diagramas intuitivos sencillos, sin embargo pueden surgir problemas, tales como:

- Falta de claridad
- Confusión de requerimientos
- Conjunción de requerimientos

Se recomienda seguir pautas sencillas para redactar requerimientos:

- Inventar un formato estándar que asegure que todos los requerimientos se adhieren al formato, para reducir la probabilidad de omisiones y hacer que los requerimientos se vean fáciles de verificar
- Utilizar el lenguaje de forma consistente, en particular distinguir los requerimientos deseables y los obligatorios.
- Resaltar el texto (con negritas) para ver los puntos clave de los requerimientos
- Evitar hasta donde sea posible, utilizar el lenguaje “técnico“ de computación

Requerimientos del sistema

Incluye diferentes modelos del sistema como el Modelo de objetos y el Modelo de flujo de datos. Los requerimientos del sistema deberán establecer lo que éste hará y no la manera en que se implementará

A menudo se utiliza el lenguaje natural para redactar las especificaciones, sin embargo surgen problemas como:

- La comprensión del lenguaje natural radica en que los lectores y redactores de la especificación utilicen la misma palabra para el mismo concepto.
- Una especificación en requerimientos en lenguaje natural es demasiado flexible (se puede decir lo mismo de diferente forma)
- No existe una forma fácil de modularizar los requerimientos en lenguaje natural. Es difícil exhibir todos los requerimientos relacionados.

Todo esto conlleva a malas interpretaciones, por lo que se han desarrollado otros enfoques: *Lenguaje natural estructurado* y *Lenguaje de descripción de diseño*.

Especificación de interfaces

Existen tres tipos de interfaz

1. Interface de procedimientos. Las salidas de un subsistema se convierten en la entrada de otro sistema.
2. Estructuras de datos que pasan de un subsistema a otro, donde se pueden utilizar PDL's
3. Representaciones de datos (como el orden de los bits)

Esquema del análisis de factibilidad (Propuesta de solución) ⁴

1. Introducción
 - a. Declaración del problema
 - b. Entorno de la implementación
 - c. Restricciones
2. Resumen de gestión y recomendaciones
 - a. Objetivos
 - b. Impacto
 - c. Escenarios
 - d. Criterio empleado en la selección del enfoque final. Tomar la decisión para la mejor alternativa
3. Descripción del sistema
4. Análisis costo-beneficio
 - a. Beneficios del sistema de información
 - i. Cálculo e impresión. Reducción en los costos por unidad de cálculo e impresión, mayor precisión en las tareas de cálculo, capacidad de cambiar rápidamente las variables y valores, mayor velocidad de cálculo e impresión.
 - ii. Mantenimiento de almacenamiento de información. Recoger y almacenar automáticamente datos de los registros, capacidad mejorada de registros en términos de espacio y costos, mantenimiento sistemático de registros, aumento en la cantidad de datos que se puedan almacenar por registro, aumento en la seguridad de almacenar registros.
 - iii. Búsqueda de registros. Recolección rápida de registros, mejor acceso a registros en base de datos grandes, cambio de registros en bases de datos, crear registros de acceso a registros, búsqueda de registros.
 - iv. Capacidad de reestructuración del sistema. Capacidad de mover grandes archivos de datos, crear nuevos archivos incorporando aspectos de otros.
 - v. Capacidad de análisis y simulación. Agregar cantidades de datos útiles para la planificación y toma de decisiones.
 - vi. Control de procesos y recursos. Reducción de la necesidad de fuerza de trabajo en el proceso de control de recursos.
 - b. Costos del sistema de información
 - i. Costos de elaboración. Costo de alquiler o compra de equipo, costo de instalación de los equipos, costo de modificación del emplazamiento de equipo (aire acondicionado, seguridad)
 - ii. Costos de la puesta en marcha. Costo del software de sistema operativo
 - iii. Costo del proceso. Costo de mantenimiento del sistema (hardware, software e instalaciones), costo de suministros (electricidad, renta de internet, renta de inmueble, etc.), costo de depreciación del hardware
5. Evaluación del riesgo técnico. Debe considerarse un estudio en función del rendimiento y restricciones que afecten el desarrollo de un sistema aceptable. ¿Puede desarrollarse el elemento del sistema de tal forma que se consiga la función y rendimiento necesario descubierto en el análisis? ¿Existe disponibilidad de recursos (hardware, software, humano)? ¿Hasta que tiempo el sistema propuesto podrá ir a la vanguardia en conjunto con la tecnología
6. Consideraciones Legales. Es importante considerar toda aquélla situación legal que pueda incurrir el desarrollo o la implantación del sistema. Es decir, responsabilidades, infracciones, sanciones legales que el proceso de desarrollo ocupe o bien en la instalación del software para su uso se requiera de tener presente situaciones legales particulares.
7. Otros temas específicos del proyecto

⁴ Se muestra un ejemplo aplicado a un caso práctico (CD), en donde se puede ver el desarrollo de cada uno de los siete puntos que este esquema utiliza, para la presentación de una propuesta de solución, a un cliente específico, p. 13

Unidad 4. Modelado y especificación formal

Tema 4.1 Visión general

Objetivo Particular

Al término de la unidad el alumno identificará la mejor forma de modelar el sistema de información propuesto, reconociendo las características principales y el objetivo fundamental de cada modelo.

Mapa conceptual



Tema 4.2 Modelos de contexto y comportamiento

Objetivo específico

Al término del tema el alumno identificará los diversos modelos que permiten desarrollar los planos arquitectónicos fundamentales para crear la documentación del sistema de información propuesto.

Mapa conceptual



Modelos de contexto

Los modelos se utilizan en el proceso de análisis para desarrollar una comprensión existente a reemplazar o a mejorar, o para especificar el sistema requerido.

Los métodos de análisis estructurado tienen varias debilidades:

- No proveen apoyo efectivo para comprender o modelar requerimientos del sistema no funcionales
- Son discriminados en el sentido de que normalmente no incluyen lineamientos para ayudar a los usuarios a decidir cuando un método es apropiado para un problema particular. Por lo general, tampoco incluye ayuda que indique como se deben adoptar para utilizarlos en un entorno
- A menudo producen demasiada documentación y la esencia de los requerimientos del sistema se opaca por la cantidad de detalles que se incluyen
- Los modelos que se producen son demasiado detallados y los usuarios a menudo encuentran difíciles de comprender. Por lo tanto, estos usuarios no pueden realmente verificar el realismo de estos modelos

Ejemplos de diferentes tipos de modelos del sistema que se podrían producir durante el proceso de análisis son:

- Modelo de procesamiento de datos. Diagrama de flujo de datos
- Modelo de composición. Diagramas Entidad-Relación muestran la manera en que las entidades del sistema se componen en otras entidades.
- Modelo arquitectónico. Muestra los subsistemas principales que componen a un sistema
- Modelos de clasificación. Diagramas de clases de objetos, herencia, muestran la manera en que las entidades tienen características comunes
- Modelo estímulo-respuesta. Diagramas de transición de estados muestran la forma en que el sistema reacciona a los eventos internos y externos.

Una vez que se han tomado las decisiones sobre los límites del sistema, una parte de la actividad de análisis es la definición de ese contexto y las dependencias que un sistema tiene en

su entorno. Por lo general, el primer paso es la producción de un modelo arquitectónico sencillo⁵ como el que se muestra en la figura 4.2.1

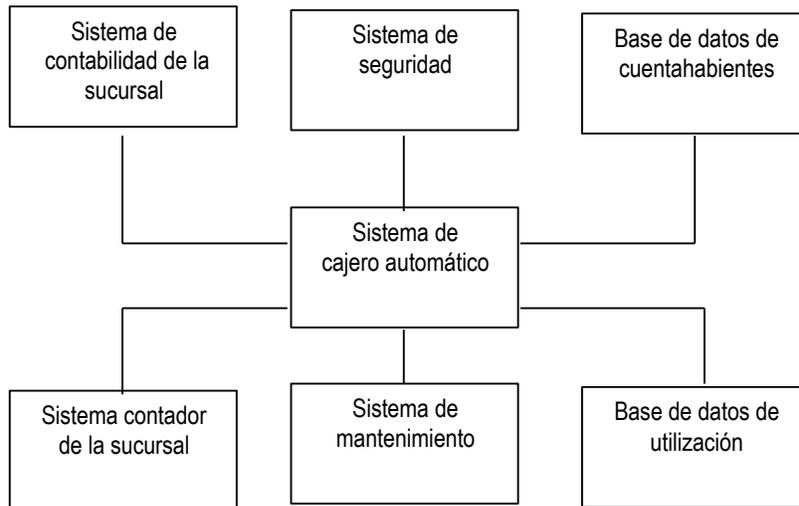


Figura 4.2.1 Modelo arquitectónico sencillo

Modelado de comportamiento

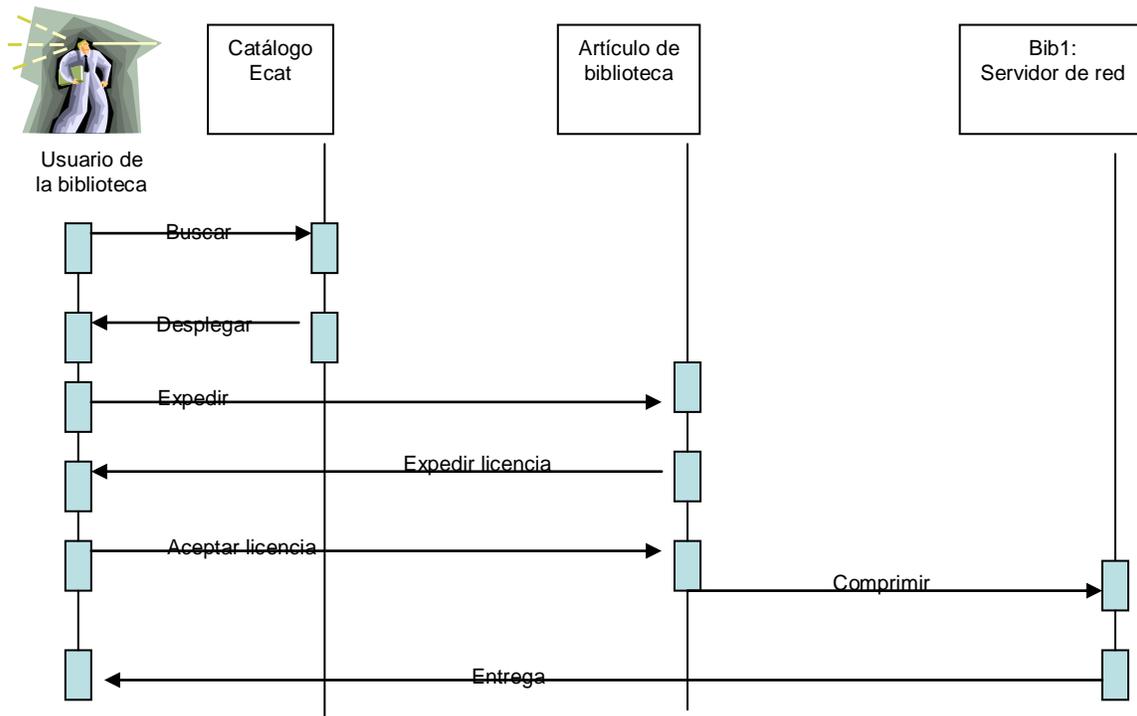


Figura 4.2.2 Modelo comportamiento

Imagine una situación en la que los paquetes de estudio se mantienen electrónicamente y se

⁵ El lector puede consultar el diagrama arquitectónico sencillo en el caso práctico, p.43 (CD)

descarga en la computadora del estudiante

En un diagrama Entidad Relación se especifican los objetos que entran y salen, los atributos que definen las propiedades de los objetos, relaciones de los objetos. Para crear un diagrama Entidad Relación es necesario:

- Listar todos los objetos de entrada y los de salida
- Por cada objeto, definir si este tiene alguna relación con otro o no la tiene
- Siempre existe una conexión objeto-relación
- Para cada objeto-relación se debe definir cardinalidad y modalidad
- Realizar de la actividad a) a la d) evitando omisiones u olvidos de objetos
- Definir los atributos de cada objeto
- Formalizar y revisar el diagrama entidad-relación
- Realizar del inciso a) al g) hasta terminar el modelado de datos

Modelos de máquina de estado (Estimulo-respuesta)

Estos se utilizan para modelar el comportamiento de un sistema en respuesta a eventos internos y externos. Dicho modelo muestra los estados del sistema y los eventos que provocan las transiciones de un estado a otro. No muestra flujo de datos, sino flujo de control, es decir existen procesos o aplicaciones que están conducidas por sucesos en lugar de datos que producen información de control, es decir, la descripción de un cambio de un estado a otro

Estos modelos son útiles para modelar sistemas de tiempo real, debido a estos sistemas a menudo están dirigidos por estímulos provenientes del entorno de sistema.

Algunas actividades a realizar para que dichos modelos cumplan su objetivo son:

- a) Identificar todos los sucesos que involucren al sistema (activa, desactivar el sistema, condiciones de interrupción, condiciones de datos)
- b) Revisar todos los elementos de control, como posibles entradas o salidas, Panel de control, sensores, alarma, línea telefónica
- c) Identificar como se alcanza cada estado
- d) Definir transacciones entre los estados

Por ejemplo, sistema de apertura de puertas, sistema de alarma, para casa o auto.

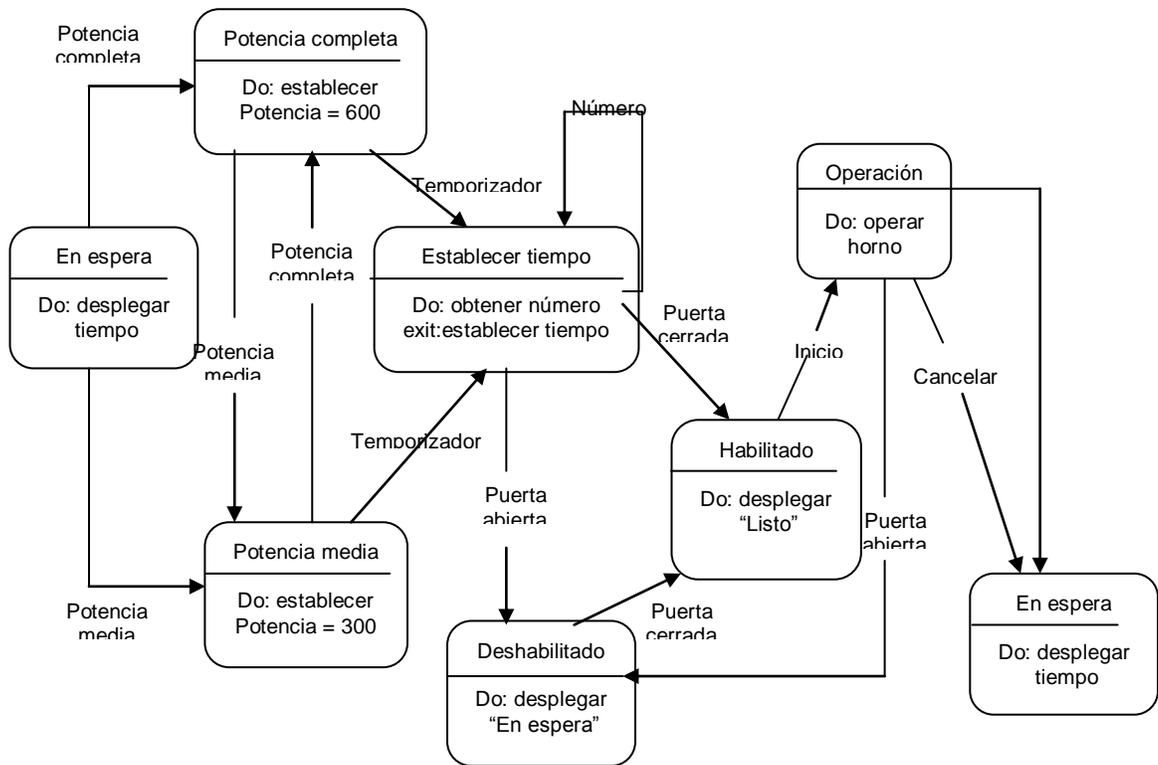


Figura 4.2.3 Modelo estímulo respuesta

En el cuadro 1 se describen los diferentes estados del horno de microondas de la figura 3, y en el cuadro 2, se describen los diversos estímulos-respuestas que el horno de microondas realiza en cada estado.

Cuadro 4.2.1 Descripción de estados del modelo estímulo respuesta. Fase de ESPERA

Estado	Descripción
En espera	El horno está en espera de entrada. La pantalla muestra el tiempo actual
Potencia media	La potencia del horno está establecida a 300 watts. La pantalla muestra "potencia media"
Potencia completa	La potencia del horno está establecido a 600 watts. La pantalla muestra pantalla completa
Establecer tiempo	El tiempo de cocción se establece al valor de entrada del usuario. La pantalla muestra el tiempo de cocción seleccionado y se actualiza en cuanto se establece el tiempo
Deshabilitado	La operación del horno se deshabilita por seguridad. La luz interior del horno se enciende. La pantalla muestra "No listo"
Habilitado	La operación del horno se habilita. La luz interior de éste se apaga. La pantalla muestra "Listo para cocinar"
Operación	El horno entra en operación. La luz interior de éste se enciende en pantalla muestra la cuenta regresiva en el temporizador. Cuando completa la cocción, la alarma suena por cinco segundos. La luz del horno se enciende. La pantalla muestra "cocción completa" mientras que la alarma suena.

Cuadro. 4.2.2 Descripción de estados del modelo estímulo respuesta. Fase de RESPUESTA

Estímulo	Descripción
Potencia media	El usuario ha presionado el botón de potencia media
Potencia completa	El usuario ha presionado el botón de potencia completa
Temporizador	El usuario ha presionado uno de los botones del temporizador
Número	El usuario ha presionado una tecla numérica
Puerta abierta	El interruptor de la puerta del hormo no está cerrado
Puerta cerrada	El interruptor de la puerta del hormo está cerrado
Iniciar	El usuario ha presionado el botón de inicio
Cancelar	El usuario ha presionado el botón de cancelar

Diagramas de flujos de datos⁶

Algunas de las características de este modelo es que se puede ver que los datos fluyen de un lugar a otro y se transforman durante este movimiento. Las transformaciones se ejecutan secuencialmente o en paralelo. Los datos se pueden procesar para cada transformación, elemento por elemento o en lote. Algunas veces se le llama modelo de canal o filtro.

Ventajas

- Permite la reutilización de transformaciones.
- Es intuitiva puesto que muchas personas visualizan su trabajo en términos de procesos de entrada y salida.
- Permite que el sistema evolucione al agregar nuevas transformaciones de forma directa
- Es sencilla de implementar sea como un sistema concurrente o uno secuencial

Desventaja

- Cada transformación debe estar acorde con las transformaciones con las que se comunica, en el formato de datos a ser procesados, o se debe imponer un formato estándar para todas los datos comunicados
- Los sistemas interactivos son difíciles de describir utilizando un modelo de flujo de datos, debido a las necesidades de procesar gran cantidad de datos.

Notación.

Flujo de datos. Los datos cambian en una dirección específica, desde su origen hasta su destino, en forma de un documento, carta, llamada telefónica o en cualquier otro medio. El flujo de datos es un “paquete” de datos

⁶ Además de los ejemplos aquí mostrados, el lector podrá darse una idea más completa de cómo elaborar los diagramas de flujo pertinentes a su proyecto de software, al revisar el Caso Practico (CD), p. 35

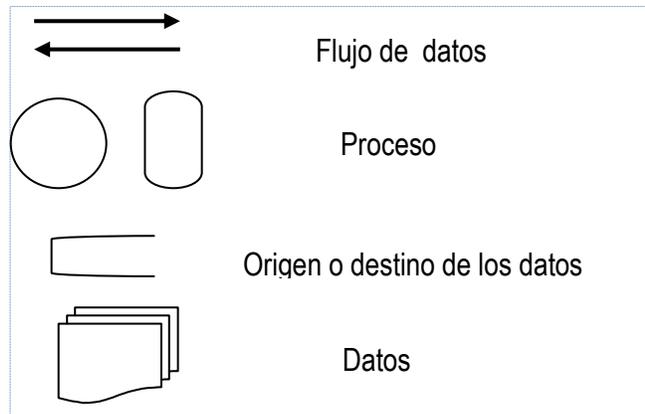


Figura 4.2.4 Notación Diagrama de Flujo de Datos

Procesos. El personal, procedimientos o dispositivos utilizan o producen o transforman datos. No se identifica el componente físico

Origen o destino de los datos. El origen externo o destino de los datos, que pueden ser individuos, programas, empresas u otras entidades, interactúan con el sistema pero están fuera de su límite

Datos almacenados. Aquí los datos se almacén o se hace referencia a ellos a través de un proceso dentro del sistema. Puede o no representar dispositivos de computadora.

Niveles de un Diagrama de Flujo de Datos (DFD).

Los modelos de flujo de datos se utilizan para mostrar como fluyen los datos a través de una secuencia de pasos de procesamiento. Los datos se transforman en cada paso antes de moverse a la siguiente etapa. Si los diagramas de flujo de datos se utilizan para documentar un diseño de software, estos pasos de procesamiento o transformaciones se representan por funciones en los programas.

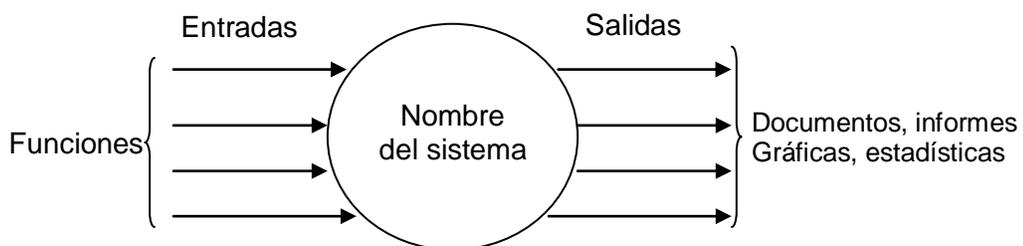


Fig. 4.2.5 Estructura del nivel 1, del DFD

Los modelos de flujo de datos muestran una perspectiva funcional en la que cada transformación representa una única función

Este modelo puede mostrar la manera en que diversos sistemas y subsistemas intercambian información. Por ejemplo, un subsistema podría ser un servidor, con una interfaz relativamente

compleja.

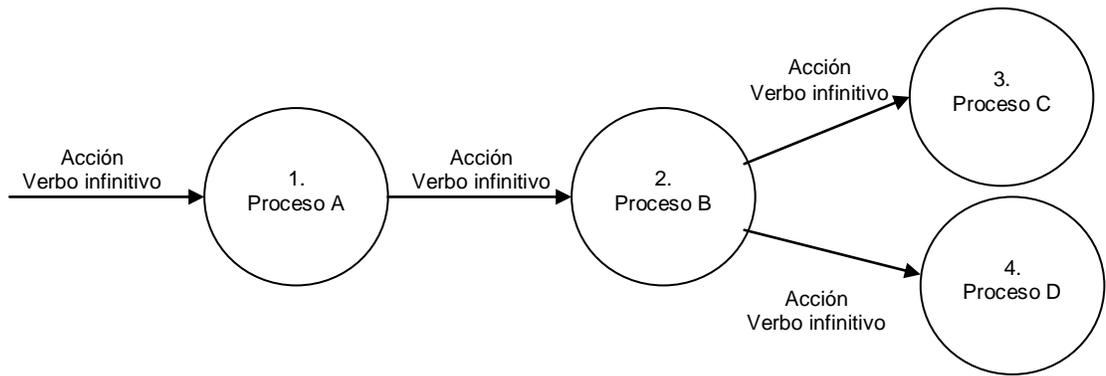


Fig. 4.2.6 Estructura del nivel 2, del DFD. Descripción de funciones

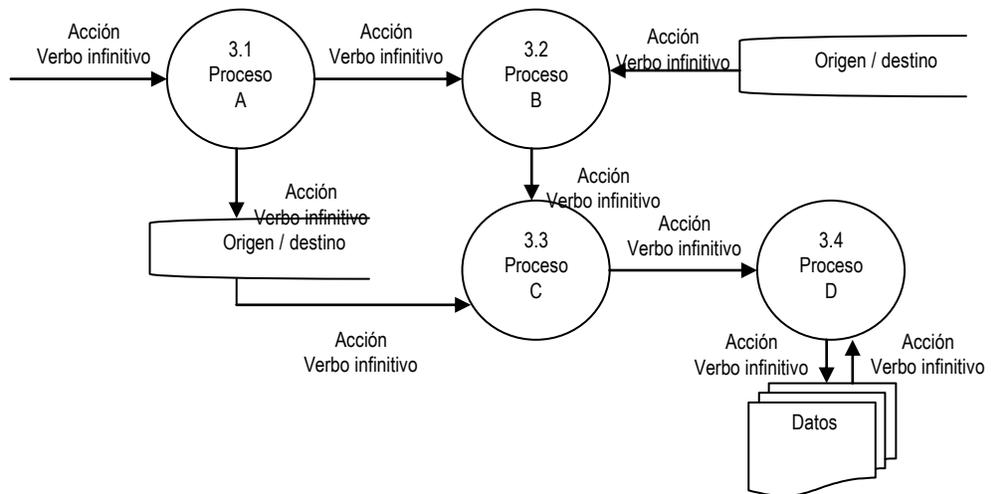
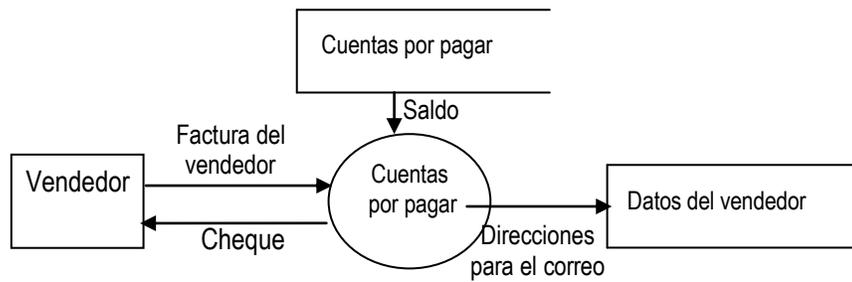


Fig. 4.2.7 Estructura del nivel 3, del DFD. Detalle de subsistema o interfaz

El análisis de flujo de datos estudia el uso de estos en cada actividad: documenta los hallazgos en los diagramas de flujo de datos que muestran gráficamente la relación entre los procesos y datos y en los diccionarios de datos que describen formalmente los datos de sistemas y donde se utilizan.

Un diagrama de flujo de datos es una representación gráfica de los procesos de datos a lo largo de una organización, como se muestra en la figura 4.2.8.

Nivel 1.



Nivel 2.

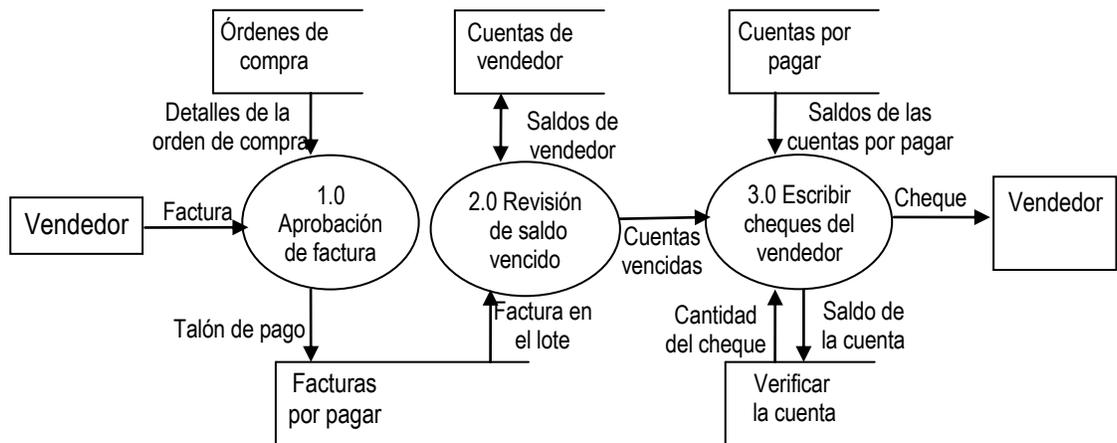


Fig. 4.2.8 DFD de una organización

Ventajas.

1. Libertad conceptual. Libertad temprana para la implementación de sistemas, colocar símbolos.
2. Mayor comprensión de las interrelaciones de los sistemas y subsistemas, en donde se permite diferenciar el sistema de su ambiente o entorno.
3. Comunicación del conocimiento de las diferentes funciones y objetivos del sistema actual a los usuarios. Es decir, permite a los usuarios que realicen comentarios acerca de las funciones y objetivos del sistema actual, de tal forma que se logre conceptualizar lo que se desea hacer y el analista también lo pueda comprender. Es importante educar al usuario, en el buen sentido, con el fin de que comprenda para que sus aportaciones sean significativas y no confundirlos.
4. Análisis de un sistema propuesto para determinar si han sido definidos los datos y procesos necesarios. Permite al analista describir cada componente del sistema propuesto y asegurar que las entradas y salidas son las suficientes y que la lógica de procedimientos es la correcta. Finalmente ayuda a corregir errores y detectarlos con más facilidad.
5. Los diagramas deben ser trazados en forma sistemática: de lo general a lo particular

Tema 4.3 Modelado de datos

Objetivo específico

Al término del tema el alumno diseñará el diccionario de datos, la estructura de las bases de datos y el diagrama entidad relación que guiará al desarrollo del sistema de información.

Mapa conceptual



Diccionario de Datos⁷

El diccionario de datos es un listado organizado de todos los elementos de datos que son pertinentes para el sistema, con definiciones precisas y rigurosas que permiten que el usuario y el analista del sistema tengan una misma comprensión de las entradas, salidas, de los componentes de los almacenes de datos (Bases de datos o tablas) y de los cálculos intermedios.

El Diccionario de Datos es una aplicación especializada, la cual se ocupa como referencia de datos. En dicha aplicación se realiza una recolecta de datos, los cuales se encuentran coordinados en base a ciertas características propias de cada dato específico, en un documento. Los diagramas de flujo tienen un papel muy importante ya que ayudan a definir las entradas al diccionario de datos.

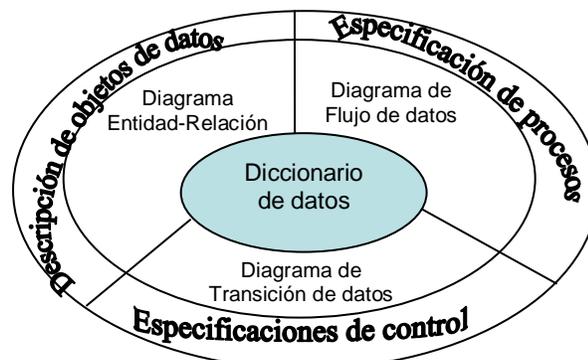


Figura 4.3.1. Relaciones implícitas del diccionario de datos

El Diccionario de datos automatizado, permite realizar referencias cruzadas, es decir comparte información, y se pueden realizar cambios. La utilidad del diccionario de datos es validar el

⁷ El lector podrá consultar el Diccionario de datos del caso práctico, en el CD, p. 49

diagrama de flujo de forma completa y precisa, así mismo el Diccionario de datos automatizado se convierte en un punto inicial para el diseño de pantallas y reportes.

El diccionario de datos tiene por objetivo la descripción lógica de los procedimientos, indicando las relaciones de datos, por lo que algunos de los rubros definidos en él son:

- Número de identificación (ID de elemento)
- Nombre descriptivo
- Alías
- Campo llave
- Descripción general del dato
- Longitud
- Tipo de dato (cantidad, fecha, alfanumérico, numérico, porcentaje, etc)
- Origen del flujo del dato (de dónde viene y a donde va)
- Destino del flujo de información
- Detallar el tipo de dato (Es un dato de Entrada o salida, es un registro o un archivo, lo contiene alguna pantalla, forma o reporte)
- Control de validación de datos, valores por omisión
- Área para comentarios generales

El uso del diccionario de datos no es un fin, sino una actividad en paralelo con el análisis y diseño de sistemas, ya que se relaciona con la descripción de objetos de datos, con la especificación de procesos y control, como se muestra en la figura 1..

Diseño de la base de datos

Algunos de los objetivos que se buscan cumplir en el diseño de la base de datos son:

- Los datos deben ser capturados una y sola una vez por el sistema, sin importar sus múltiples procesos y consultas
- La claridad y exactitud de los documentos fuente (precodificados de preferencia), determinará el número y la complejidad de los procesos necesarios para hacerlo parte del sistema.
- Las cifras de control serán actualizadas y validadas antes y después de cada actualización y/o resguardo de manera automática (lógica de programas)
- Los datos serán almacenados en un solo lugar en la base de datos (o archivos tradicionales) exceptuando aquéllos que funjan como “apuntadores” y “llaves”
- Todo archivo deberá ser susceptible de impresión formateada y editada para fines de auditoría
- Los archivos temporales (de transacciones, por ejemplo) deberán ser retenidos cuando menos en un ciclo, sí actualizan a los maestros o a la base de datos.
- Todo archivo contará con procedimientos de resguardo y de ser “no esencial”, contará con procedimientos de reorganización.
- En lo posible, las características y formatos de registros “parámetros” deberán ser semejantes.

El contenido de la Base de datos debe involucrar los siguientes datos principalmente:

- ID de la base de datos
- Nombre del archivo (descriptivo y único)
- Aliás
- Descripción
- Tipo de archivo (base de datos, archivo indexado, archivo secuencial, archivo directo, archivo manual)
- Cantidad máxima y promedio de registros
- Crecimiento promedio
- Programas que los utilizará
- Área para comentarios generales

El dominio de la información involucra a los datos y al control de estos. Hay tres visiones:

1. Contenido de la información. Involucra objetos individuales de datos, objetos de control, atributos y/o características de cada objeto, interrelación entre objetos.
2. Flujo de la información. Muestra como cambia la información y el control de datos.
3. Estructura de la información. Permite organizar y controlar de forma interna los datos, mediante tablas de decisión, árboles de decisión o estructuras jerárquicas.

Algunas directrices para la representación de especificaciones son:

- El formato de la representación y el contenido deberían estar relacionados con el problema
- La información contenida dentro de la especificación debería estar escalonada (capas de información)
- La numeración de párrafos y diagramas debería indicar el nivel de detalle que se muestra
- Los diagramas y otras formas de notación deberían restringirse en número y ser consistentes en su empleo
- Las representaciones deben permitir revisiones.

Diagrama Entidad-Relación⁸

Denominado por sus siglas como: E-R; este modelo representa a la realidad a través de un esquema gráfico empleando la terminología de **entidades**, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas **atributos**, el enlace que rige la unión de las entidades está representada por la **relación** del modelo.

De esta manera, daremos algunas definiciones de los elementos principales de un diagrama Entidad Relación:

⁸ En el caso práctico se presentan los diagramas entidad relación sobre el proyecto de software, (CD), p.48.

4. Modelado y especificación formal

- Objetos. Representación de cualquier composición de información (cualquier cosa que produce o consume información). Un objeto de datos no incluye referencia de acción solo es dato.
- Atributos. Son propiedades de un objeto, se usan para nombrar, describir o hacer referencias.
- Relaciones. Conexión de objetos de datos en formas diferentes.
- Cardinalidad. Implica comprender la cantidad de ocurrencias del objeto X que están relacionadas con el objeto Y. Uno a uno, uno a muchos, Muchos a muchos
- Modalidad. Obligatorio u opcional.

Algunos ejemplos de modelos E-R, considerando las cardinalidades que existen entre ellos. Diseñar el modelo E-R, para la relación Registro de automóvil que consiste en obtener la tarjeta de circulación de un automóvil con los siguientes datos:- Automóvil- Modelo, Placas, Color - Tarjeta de circulación -Propietario, Número de serie, Tipo, se muestra en la figura 2.



Figura 4.3.2 Ejemplo1. Registro de automóvil

En este ejemplo existe una relación de pertenencia de uno a uno, ya que existe una tarjeta de circulación registrada por cada automóvil.

En el siguiente ejemplo, representamos que existe un solo presidente para cada país.

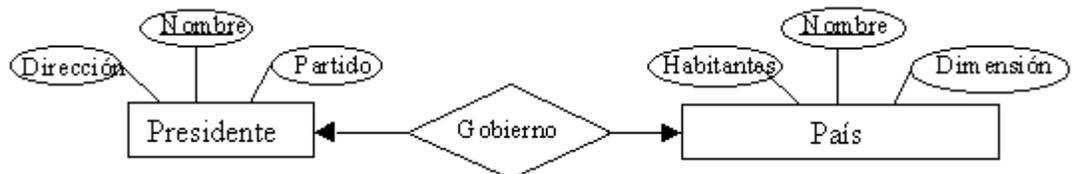


Figura 4.3.3 Ejemplo 2. Presidentes y gobiernos

Otro ejemplo es la apertura de cuentas bancarias. Un cliente puede tener muchas cuentas, pero una cuenta puede llegar a pertenecer a un solo cliente, hoy día existen cuentas registradas a favor de más de una persona.

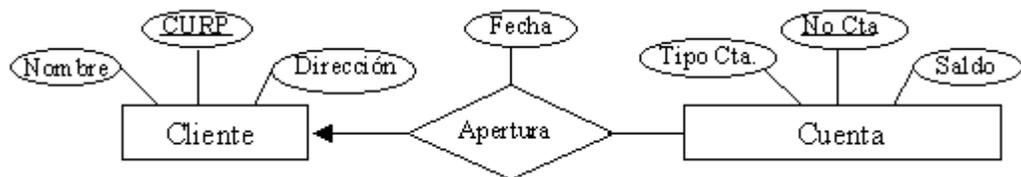


Figura 4.3.4 Ejemplo 3. Cuentas bancarias

Tema 4.4 La especificación formal en el proceso de software

Objetivo específico

Al término del tema el alumno reconocerá la importancia de las matemáticas y ciencias exactas en los métodos formales, así como su rol en el desarrollo de software especializado.

Especificación formal

Los métodos formales de desarrollo de software no son muy utilizados, debido a que muchas empresas de desarrollo de software no consideran costeable aplicarlos en sus procesos de software. Sin embargo la necesidad de una especificación formal significa que los lenguajes de especificación se deben basar en conceptos matemáticos.

En los 80's se predijo que para el siglo XXI una gran cantidad de software se desarrollará utilizando métodos formales, pero no se ha cumplido por las siguientes causas:

- Ingeniería de software exitosa. Se utilizan métodos estructurados, administración de la configuración, ocultamiento de información lo cual condujeron a mejorar la calidad del software.
- Cambios en el mercado. Hoy día se prefiere tener un software lo más rápido posible aceptando algunas fallas. Localidad en un factor importante para debe llevarse a cabo en el contexto de entregas rápidas
- Alcance limitado de los métodos formales. No son apropiados para especificar las interfaces e interacciones con el usuario.
- Escalabilidad limitada de los métodos formales

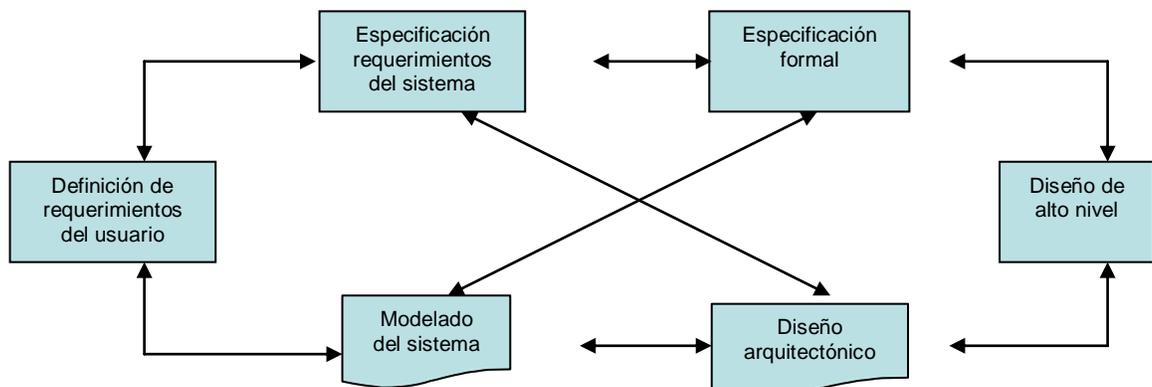


Figura 4.4.1 Productos de la especificación formal

La rama de las matemáticas que se utiliza son: matemáticas discretas, teoría de conjuntos, lógica, álgebra, etc. Existen dos enfoques para la especificación formal que se han utilizado para redactar especificaciones detalladas de sistemas no triviales. El primero llamado enfoque algebraico, se describen en términos de operaciones y relaciones. El segundo es un enfoque basado en modelos, donde se construye un modelo del sistema usando elementos matemáticos

como conjuntos y sucesiones.

La especificación formal es una excelente forma de descubrir los errores de especificación evitando ambigüedades. El uso de los métodos formales crece en el desarrollo de sistemas críticos o de tiempo real, debido a que reducen costo de fallas y de validación. Por ejemplo, control de tráfico aéreo, control de señalización de vías, control médico, etc.

Las matemáticas en el desarrollo de software pueden describir en forma exacta una situación física, el comportamiento de un objeto o el resultado de una acción. Las matemáticas proporcionan una transición suave entre las actividades de ingeniería del software. Las matemáticas permiten la abstracción y es un excelente medio para el modelado dado que es un medio exacto hay pocas probabilidades de ambigüedad o vaguedad.

Algunos conceptos utilizados en el desarrollo de los métodos formales son:

- Una tabla de símbolos. es una colección de elementos sin duplicar, es utilizada por un sistema operativo para almacenar nombres de usuarios de los sistemas.
- Invariante de dato. La tabla no contendrá más de un número máximo de elementos, no hay datos repetidos.
- Se maneja también como concepto el estado, la operación (donde se añade, se eliminan o se comprueban los procesos), la precondition (antes de procesar), postcondición (valor lógico).
- Conjunto= operadores de conjuntos operadores lógicos sucesiones.
- Los lenguajes formales de especificación formal: CSP, HOR, LARCH, VDM, Z

Finalmente, algunos autores señalan de forma recurrente que “Los diez mandamientos de los métodos formales”, son:

1. Seleccionar la notación adecuada
2. Formalizarás pero no de más
3. Estimarás los costes
4. Poseerás un experto de métodos formales a tu disposición
5. No abandonarás tus métodos formales de desarrollo
6. Documentarás suficientemente
7. No comprometerás los estándares de calidad
8. No serás dogmático
9. Comprobarás, comprobarás y volverás a comprobar
10. Reutilizarás cuanto puedas

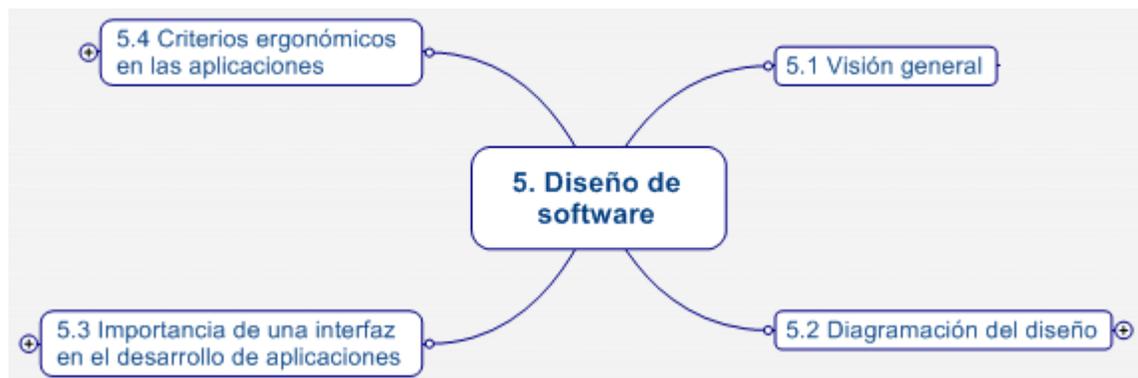
Unidad 5. Diseño de software

Tema 5.1 Visión general

Objetivo Particular

Al término de la unidad, el alumno reconocerá la importancia de la diagramación del diseño y las interfaces en el desarrollo de aplicaciones para que éstas sean funcionales en el sistema de información propuesto.

Mapa conceptual



Tema 5.2 Diagramación del diseño

Objetivo Particular

Al término del tema, el alumno identificará los diferentes tipos de diseño que involucra esta fase del ciclo de vida de un sistema de información, resaltando la funcionalidad de cada uno de ellos.

Mapa conceptual



Definición

La identificación del diseño y la documentación de una arquitectura de software tienen tres ventajas primordiales: la comunicación entre los programadores (stakeholders), debido a que la arquitectura es una presentación de alto nivel del sistema y es utilizada como punto de discusión; el análisis del sistema en requerimientos críticos como el desempeño, la fiabilidad y mantenibilidad, y finalmente la reutilización a gran escala, es decir la arquitectura se puede transferir a lo largo de los sistemas con requerimientos similares y así poder reutilizar software a gran escala.

Como una segunda etapa del Ciclo de vida de un sistema de información, el diseño recoge los productos finales del análisis, y fundado en ellos “TRADUCE” los lineamientos que de ahí emanan en postulados de carácter técnico que sirvan de guía a las etapas posteriores, sin perder el contexto determinado por el sistema aprobado en la propuesta del análisis.

Definición 1. Diseño es conocer las diferentes actividades para conducir (ampliando y adaptando los requisitos de la propuesta del análisis), la creación del sistema hasta el momento en que se dé inicio al desarrollo del código.

Definición 2. Es el proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema con suficientes detalles para permitir su realización física.

La fase del diseño comprende cuatro etapas, como se muestra en la figura 1.

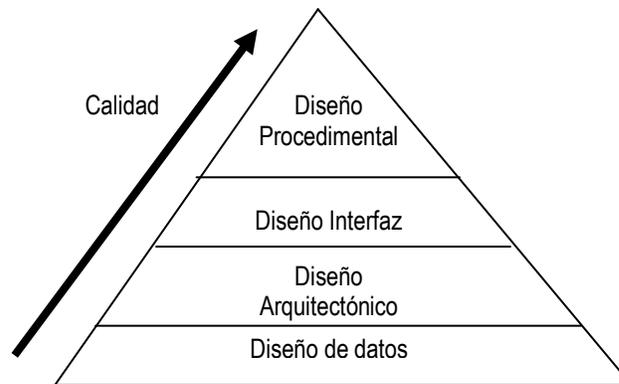


Figura 5.2.1 Fases del diseño

- Diseño de datos. Transforma el modelo de dominio de la información, en las estructuras de datos necesarias para implementar el software (diagrama entidad-relación, contenido detallado del diccionario de datos, objetos de datos)
- Diseño de interfaz. Como se comunica el software entre sus propios módulos, con otros sistemas y los usuarios. (Diseño de entradas, salidas , interfaz y procesos)
- Diseño procedimental. Transforma elementos estructurales de la arquitectura del programa

Existen puntos guía que nos permite no desviarnos de nuestro objetivo en la fase del diseño..

- a. Todas las estructuras de datos y las operaciones a llevar a cabo en cada una de ellas deberían estar claramente identificadas
- b. Se debería establecer un diccionario de datos y usarlo para definir el diseño de datos y del programa
- c. Las decisiones de diseño de datos de bajo nivel deberían dejarse para el final del proceso de diseño (refinamiento paso a paso)
- d. La representación de las estructuras de datos deberían conocerla sólo aquellos módulos que deban hacer uso directo de los datos contenidos dentro de la estructura (ocultación de información)
- e. Debería desarrollarse una biblioteca de estructuras de datos útiles y de las operaciones que se les pueden aplicar
- f. Un diseño del software y un lenguaje de programación debería soportar la especificación y realización de los tipos abstractos de datos. (Seleccionar adecuadamente el software de desarrollo del sistema así como del manejador de base de datos)

Diseño preliminar

En la primera etapa el objetivo es desarrollar un modelo conceptual mediante “acercamientos” sucesivos, es decir, partir de lo general a lo particular. Esto es, realizar un primer intento del modelo a través de un diagrama que identifique los datos de entrada, describa vaga y brevemente los procesos internos y se ennumeran las salidas.

A partir de tan pequeño principio y por desgloses cada vez más amplios, debe llegarse a un diagrama general del sistema cuyos nexos mostrarán los subsistemas que lo componen, y en ellos, las cadenas de programas necesarias, así como los archivos y/o base de datos con los que trabaja, así como la distribución de información y reportes producidos.

Aún cuando el mayor detalle corresponde a la segunda etapa, algunas cosas deberán quedar claras desde la primera etapa (Arquitectura del sistema), tales como, la identificación del producto principal del sistema, el cual debe concordar con los objetivos de la propuesta, enumerando la procedencia y tratamiento (proceso) de los datos.

La aprobación del diseño preliminar, que culmina la primera etapa, adquiere tan relevante importancia que algunos analistas la consideran como una fase intermedia de la etapa de diseño.

Resumidamente, procede a tal aprobación, la comparación punto por punto de la “Propuesta” del sistema contra el diseño preliminar y dicho procedimiento se hace reiterativo en la medida en que la comparación arroje discrepancias, lo que implica una revisión autocrítica tanto de la propuesta del analista como del diseño preliminar; la participación del usuario en la aprobación resulta indispensable.

Diseño detallado

La fase del sistema detallado del sistema corresponde virtualmente a la construcción de las especificaciones de cada programa, lo que viene a ser el puente entre la idea general o preliminar del diseñador y el equipo de programadores, sin embargo, ésta etapa también observa otros puntos de interés como diseño de formas y reportes (al detalle y estableciendo los controles de producción, distribución, etc.). Aún así, las especificaciones de programas aportan mayor peso específico en el detallado del sistema, aportando algunos beneficios importantes como el ofrecer al diseñador una última oportunidad de revisar la lógica del sistema como un todo, proporcionar una comunicación amplia y documentada con los programadores, etc.

Diseño arquitectónico

Define la relación entre los principales elementos estructurales del programa, es decir, describe la transición del flujo de información a una estructura donde se identifiquen aquellas variables exógenas que intervienen en el sistema a desarrollar.

Los sistemas grandes siempre se descomponen en subsistemas que suministran algún conjunto relacionado de servicios. El proceso de diseño inicial que tiene por objeto identificar estos subsistemas y establecer un marco de trabajo para el control y comunicación de los subsistemas se llama Diseño arquitectónico

Algunos puntos guía son:

- a. Establecer el tipo de flujo de información

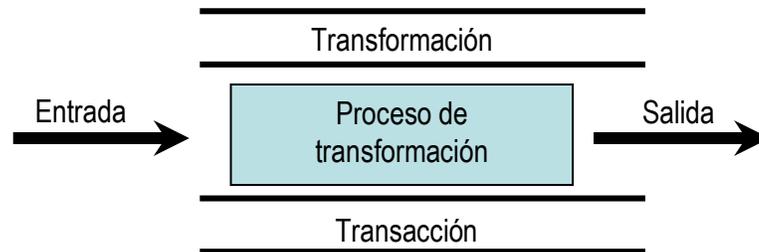


Figura 5.2.2 Proceso de transformación de información

- b. Indicar los límites del flujo
- c. Convertir el DFD en la estructura del programa (PDL's, Diagrama de bloques, etc.)
- d. Definir jerarquías de control, descomponiéndola mediante particionamiento
- e. Refinar la estructura resultante usando medidas y heurística de diseño.

Objetivos

- Implementar todos los requisitos explícitos contenidos en el modelo de análisis y acomodar todos los requisitos implícitos que desea el cliente
- Debe ser una guía que puedan leer y entender los que construyan el código, los que prueban y mantienen el software
- Proporcionar una idea completa de que es el software enfocando los dominios de datos, funcional y de comportamiento desde la perspectiva de la implementación.

Los objetivos del diseño pueden dividirse en tres grupos:

- a. Inmediato
- b. A corto plazo
- c. A largo plazo

Principios directrices de diseño

- El proceso de diseño no debería ponerse "orejeras".
- Se deben seguir los pasos del diseño hasta el modelo del análisis
- El diseño no debe inventar nada que ya este inventado (reutilización)
- El diseño debe minimizar la distancia intelectual
- El diseño debe estructurarse para admitir cambios
- El diseño debe estructurarse para degradarse poco a poco incluso cuando se enfrenta a datos, sucesos o condiciones operativas aberrantes (aceptar circunstancias inusuales)
- El diseño no es escribir código y escribir código no es diseñar
- Se debe valorar la calidad del diseño mientras se crea, no después de terminarlo
- Se debe revisar el diseño para minimizar los errores conceptuales (semánticos)(omisiones, ambigüedades, inconsistencias)
- Desarrollar prototipos que permitan al usuario entender como sería la interacción hombre-máquina

La etapa de diseño de un sistema de información está técnicamente orientada a responder la pregunta ¿Cómo hacerlo?, una vez que la etapa de análisis nos ha dicho ¿Qué hay que hacer?,

recurrentemente el diseñador utilizará metodológicamente las preguntas:

- ¿Qué pasaría en caso de...?
- ¿Por qué no intentarlo...?

Las etapas en que podemos subdividir ésta fase, serían las siguientes:

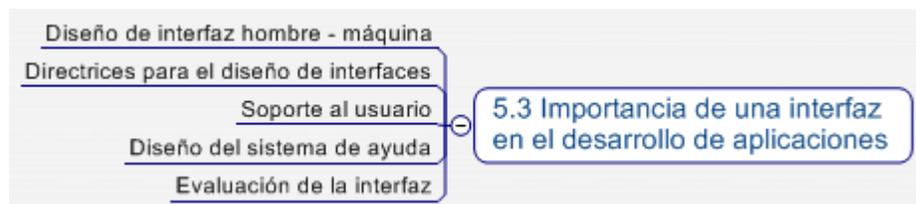
- Diseño preliminar
- Arquitectura del sistema
- Diseño de las salidas
- Diseño de las entradas
- Diagrama de Flujo de datos
- Diseño de la base de datos
- Diseño de Esquema de control
- Identificación y utilización de productos secundarios
- Identificación de los recursos de almacenamiento (base de datos, archivos, etc.)
- Diseño detallado (detallado o particular del sistema)

Tema 5.3 Importancia de una interfaz en el desarrollo de aplicaciones

Objetivo Particular

El alumno reconocerá las principales principios y directrices que permiten el buen diseño de interfaces adecuados a los diferentes usuarios para el sistema de información propuesto, mediante la presentación de diversos ejemplos.

Mapa Conceptual



Diseño de interfaz hombre - máquina

La interfaz se puede definir como interna y externa, para ambas el diagrama de flujo será nuestro apoyo ya que nos permite ver como fluye la información internamente y los datos de entrada, formato y tipo determinan la interfaz externa.

El modelo de usuario se categoriza como:

- Novatos: Sin conocimiento sintáctico del sistema y escaso conocimiento semántico (comprensión de funciones, datos de entrada, salida, metas y objetivos del sistema) de la aplicación o del uso de la computadora en general.
- Usuarios esporádicos con conocimientos: conocimiento semántico razonable de la aplicación, pero que recuerda vagamente la información sintáctica necesaria para usar la interfaz.
- Usuario frecuente con conocimientos: buenos conocimientos semánticos y sintácticos que a menudo conduce al “síndrome del usuario potente”, es decir, individuos que buscan acceso directos y modos abreviados de interacción

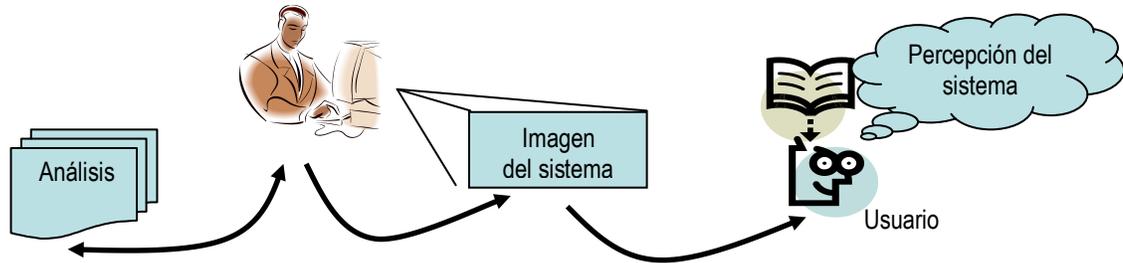


Figura 5.3.1 Modelo de la percepción del analista

La percepción del sistema (modelo del usuario), es la imagen del sistema que lleva en la cabeza el usuario final.

La imagen del sistema combina la manifestación exterior del sistema basado en computadora (el aspecto y sensación de la interfaz) con toda la información de soporte (libros, manuales, cintas de video) que describen sintaxis y semántica del sistema.

Cuando coincide la imagen del sistema y la percepción del sistema, los usuarios se sienten a gusto con el software y lo utilizan eficazmente.

Directrices para el diseño de interfaces

Interacción general

- Ser consistente
- Ofrecer respuestas significativas
- Pedir verificación de cualquier acción destructiva, ¿Está usted seguro?
- Permitir deshacer la mayoría de las acciones
- Reducir la cantidad de información que se debe memorizar entre las acciones
- Buscar la eficiencia en el diálogo, el movimiento y el pensamiento (ubicación del ratón o ventanas desplegadas para tratar de que el usuario no tenga en mente ¿Qué significa esto?)
- Perdonar los errores (autoprotección de errores)
- Categorizar las actividades por función y organizar la pantalla de acuerdo a esto
- Proporcionar ayudas de acuerdo al contexto (ayuda en línea)
- Usar verbos de acción sencillos o frases verbales cortas para nombrar las órdenes (a través de menús o íconos)

Visualización de la información⁹

- Mostrar sólo la información que sea relevante en el contexto actual
- No abrumar al usuario con datos, utilizar un formato de visualización que le permita una rápida asimilación de la información
- Usar etiquetas consistentes, abreviaciones estándar y colores predecibles
- Permitir al usuario mantener el contexto visual (maximizar o minimizar ventanas)

⁹ En el caso práctico, se muestran algunos diseños de vistas del sistema de información sobre la entrada de datos, p. 44

- Producir mensajes de errores significativos
- Usar mayúsculas y minúsculas, tabulaciones y agrupamientos del texto para ayudar a entenderlo
- Usar presentaciones analógicas para representar información que es más fácilmente y asimilable de esta manera (Ilustrar con objetos, gráficos, etc. algo relevante)
- Estudiar la geografía disponible en la pantalla y usarla efectivamente

Entrada de datos

- Minimizar el número de acciones de entrada de datos que necesita realizar el usuario
- Mantener la consistencia entre la visualización y la información de datos
- Permitir al usuario personalizar la entrada (Puede darse el caso de deshacerse de algunos mensajes de entrada destinados a usuarios novatos)
- La interacción debería ser flexible pero también sintonizarse el modo preferido del usuario de entrada (teclado, ratón, etc.)
- Desactivar las órdenes que son inapropiadas en el contexto de las acciones actuales (significa protección al usuario de errores)
- Dejar al usuario controlar el flujo interactivo (volver a la actividad después de un error sin necesidad de reinicializar el programa o la máquina)
- Proporcionar ayuda en todas las acciones de entrada
- Eliminar entradas innecesarias (por ejemplo evitar escribir .00 o información que el sistema ya tiene como la fecha , hora, proporcionar valores predeterminados)

El proceso de diseño de la interfaz lleva el siguiente enfoque:

1. Establecer objetivos e intenciones de la tarea (entiéndase como tarea lo que comúnmente realizan los usuarios = actividades del usuario)
2. Analizar cada objetivo o intención en una secuencia de acciones específicas
3. Especificar la secuencia de la acción tal como se ejecutará a nivel de la interfaz
4. Indicar el estado del sistema por ejemplo ¿Cómo es la interfaz cuando se realiza una acción de la secuencia?
5. Definir los mecanismos de control al estado del sistema
6. Mostrar como afectan los mecanismos de control al estado del sistema
7. Indicar como interpreta el usuario el estado del sistema por la información que le proporciona a través de la interfaz

A medida que evoluciona el diseño de la interfaz del usuario, emergen casi siempre cuatro aspectos comunes del diseño:

- **El tiempo de respuesta del sistema.** Tiene dos características importantes: duración y variabilidad. Si la duración del tiempo de respuesta es demasiado largo, el resultado será la frustración y estrés del usuario. En cuanto a la variabilidad, se refiere a establecer un ritmo de respuesta. Sin embargo, el software moderno proporciona una ayuda en línea que permite al usuario responder una pregunta o resolver un problema sin abandonar la interfaz. Si un sistema provee al usuario con ayuda en línea, se incrementará la amigabilidad de la interfaz.
- **La manipulación de la información de errores.** Los mensajes de error o advertencias ofrecen información inútil o engañosa para aumentar la frustración

del usuario. En general todos los mensajes de error o de advertencia producidos por un sistema interactivo deberían tener las siguientes características:

- i. El mensaje debería describir el problema en un argot que pueda entender el usuario
 - ii. Proporcione información constructiva para recuperarse del error
 - iii. Indique cualquier consecuencia negativa del error (archivos dañados), de forma que el usuario pueda comprobarlos
 - iv. Se debe acompañar por una señal audible o visible
 - v. El mensaje no debe hacer juicios, no culpar al usuario
- **Etiquetado de órdenes.** En muchas situaciones, se proporciona al usuario una opción: las funciones de software se pueden seleccionar de un menú estático o desplegable o invocado a través de alguna secuencia de órdenes con el teclado.
 - i. ¿Tendrán todas las opciones del menú, su correspondiente orden?
 - ii. ¿Qué formato deben tener las órdenes?
 - iii. ¿Qué puede hacer si se le olvida una orden?
 - iv. ¿Puede el usuario personalizarlo o abreviar las órdenes?
 - **Facilidades de ayuda al usuario.** Existen kits de herramientas de interfaz de usuario o sistemas de desarrollo de interfaz de usuario (SDIU). La computadoras personales proveen una GUI (Interfase Gráfica del Usuario), que permite el despliegue en color de alta resolución e iteración utilizando tanto el ratón como el teclado. Algunas características de las interfaces gráficas del usuario son: ventanas, íconos, menús, apuntador (para seleccionar opciones de menú), gráficos (que pueden mezclar texto en el mismo despliegue). Los GUI son fáciles de aprender, ya que interactúa mediante pantallas múltiples (ventanas) sin perder de vista la información generada durante la primera tarea, así también existe rapidez para un acceso inmediato a cualquier punto de la pantalla.

¿Cómo introducir la información al sistema?

De acuerdo con Shneiderman en 1998, se clasifican en cinco formas

Manipulación directa	Son videojuegos, Aplicaciones CAD, Bote de basura para eliminar objetos.
Ventaja	Interacción rápida e intuitiva
Desventaja	Puede ser difícil, sólo es adecuado cuando existan metáforas visuales para tareas y objetos

Selección de menús	Sistemas de propósito general
Ventaja	Evita errores de teclado
Desventaja	Lenta para el usuario experimentado, puede ser compleja si existen varios menús

Llenado de formularios	Control de almacén, proceso de préstamo a personal.
Ventaja	Es fácil de aprender por su sencillez, se ocupan botones o selección de posibles respuestas.
Desventaja	Ocupan mucho espacio en pantalla

Lenguaje de comandos	Son comandos especiales dirigidos a una tarea específica. Sistemas operativos, Sistemas bibliotecarios de recuperación de información
Ventaja	Son poderosos y flexibles
Desventaja	Son difíciles de aprender, en caso de conocer los suficiente se pueden cometer errores irremediables, son manipulados por expertos.

Lenguaje natural	Sistemas de búsqueda en la Internet, procesos de búsqueda en general
Ventaja	Accesible a usuarios casuales, son fáciles de ampliar
Desventaja	Se requiere teclear más, y en ocasiones los sistemas de lenguaje natural no son fiables.

Cada uno de estos estilos pueden mezclarse en una misma aplicación. Sin embargo el analista debe saber diseñar formas de captura, las cuales deben ser:

- a. Fáciles de llenar, es decir, el llenado se debe realizar de izquierda a derecha y de arriba a abajo, indicando títulos de acuerdo a los datos que se desea que el usuario ingrese, éstos títulos pueden estar en línea , debajo de la línea, en cuadros de texto, selección vertical u horizontal.

Título de línea Título debajo de línea	Nombre		
	Paterno	Materno	Nombre(s)
Los títulos deben ser pequeños e incluso se pueden considera como notas aclaratorias	DD / MM / AA		
	Nombre A. Paterno A. Materno		

Figura 5.3.2 Esquema sencillo de etiquetado de datos

- b. Que satisfagan el objetivo de su diseño, ya que de acuerdo a éste la información que se capture puede ser útil a más de un departamento
- c. El diseñador de entradas deben asegurar un llenado preciso, ya que con esto se puede lograr la disminución de errores que integren un control interno.
- d. Finalmente se debe considerar que la forma sea atractiva para su uso, a través del color, los tipos de letra, las etiquetas, títulos y categorías de cada dato, pero sobre todo mantener un orden lógico

Principios de Diseño de la interfaz de usuario

Los diseñadores de interfaz deben tomar en cuenta las capacidades físicas y mentales de la gente que utilizará el software, ya que las personas olvidan rápidamente y cometen varios errores especialmente cuando tienen que manejar demasiada información o están bajo presión.

- **Familiaridad con el usuario.** Debe utilizar conceptos que se toman de la experiencia de las personas que más utilizan los sistemas. Por ejemplo: Un sistema controlador de tráfico aéreo, los objetos son aviones, trayectorias de vuelo, guías, etc. Las operaciones asociadas podrían ser: incrementar o reducir la velocidad del avión, ajustar posición del avión y cambiar la altura.

- **Consistencia.** Los comandos o menús del sistema deben tener el mismo formato, los parámetros deben pasarse a todos los comandos de la misma forma, y la puntuación de los comandos debe ser similar. Una interfaz consistente reduce tiempo de aprendizaje: CTRL-V, CTRL-C, CTRL-X. Suena sensato implementar el borrado de entidades arrastrándola a un bote de basura, pero no es natural borrar texto en un procesador de palabras de ésta forma.
- **Mínima sorpresa.** Los usuarios se enojan demasiado cuando el sistema se comporta de forma inesperada. Los diseñadores de interfaces deben asegurar que las acciones comparables tengan efectos comparables.
- **Recuperabilidad.** Se deben incluir mecanismos que permitan al usuario recuperarse de los errores. El diseño de interfaz puede minimizar estos errores aunque nunca se eliminen completamente. La recuperación del error se puede hacer de dos formas:
 - a. Confirmación de acciones destructivas. Si los usuarios llevan a cabo una acción que es potencialmente destructiva, se les preguntará que confirmen que esto es realmente lo que desean antes de que la información se destruya.
 - b. Proveer un recurso para deshacer. El recurso deshacer regresa el sistema al estado previo antes de que ocurriera la acción. Es conveniente tener varios niveles de este recurso puesto que los usuarios no siempre reconocen de forma inmediata que han cometido un error.
- **Guía al usuario.** Asistencia al usuario, las cuales se integran al sistema en diferentes niveles, las cuales van desde una descripción básica hasta una descripción completa de las características del sistema, sin llegar a saturar la información proporcionada.
- **Diversidad de usuarios.** El diseño de interfaces debe considerar que existen diversos tipos de usuarios y tocando los dos tipos extremos de usuario podemos pensar en que algunos usuarios son temporales y ocasionalmente tienen interacción con el sistema, por lo que su necesidad de interfaz deberá ser satisfecha mediante una Guía que los lleve de la mano. Sin embargo existen usuarios que son potenciales, y que ocupan en su mayor parte el sistema, por lo que requerirán de teclas de acceso rápido para ejecutar ciertas funciones.

Color en el diseño de la interfaz.

El color puede ayudar al usuario a comprender y manejar la complejidad del sistema, por lo que los diseñadores deben ser conservadores, lo que implica:

- Limitarse en el número de colores utilizados. No más de cuatro o cinco colores en una ventana y no más de siete en una interfaz.
- Utilizar un cambio de color para mostrar un cambio de estado del sistema. Resaltar el color es muy importante en los despliegues complejos
- Utilizar el código de colores en una forma consciente y consistente, si los mensajes de error son en rojo, no se ocupará este color para ningún otro caso.
- Ser cuidadoso al utilizar pares de colores, se recomienda: Negro sobre amarillo, verde sobre blanco, azul sobre blanco, blanco sobre azul y amarillo sobre negro.
- Los dos errores más comunes son: asociar significados a un color particular, y utilizar demasiados colores en los despliegues. Dichas acciones pueden tener consecuencias tales como el malinterpretar el significado de un color, además se

ha comprobado que cerca del 10% de los hombres no perciben el color, así mismo las percepciones del significado del color pueden ser diferentes de acuerdo a la persona, por ejemplo rojo en general puede significar peligro, mientras que para un químico significa caliente. Por otra parte una cantidad de colores perturba y cansa a la vista ocasionando confusión al usuario.

Interacción del usuario

El usuario ante un sistema nuevo o desconocido para él, siempre le surgen dos preguntas ¿Cómo introducir la información al sistema? ¿Cómo el sistema mostrará esa información? La calidad de entrada determina la calidad de salida, por lo que el diseño de entradas debe satisfacer:

- Efectividad. Cada entrada del sistema debe tener definido un propósito específico
- Precisión. El diseño de la entrada debe considerar la información necesaria
- Facilidad de uso. Las formas y pantallas que el sistema genere deben ser directas y de fácil comprensión
- Consistencia. Los datos solicitados en aliguas de las formas o pantallas se deben agrupar de tal forma que la información sea coherente
- Simplicidad. La vista de las entradas debe ser fácil de comprender, sin amontonamientos con enfoque hacia el usuario.
- Atractivo. La vista de la entrada debe agradar al usuario a través de un diseño interesante

¿Cómo mostrará el sistema esa información?¹⁰

La presentación de la información puede ser una presentación directa de la información de entrada. Por ejemplo texto en un procesador de palabras o presentar la información gráficamente.

Es recomendable separar o manejar por separado el software requerido para la presentación de la información misma, lo que contradice la filosofía orientada a objetos. Sin embargo esto presupone que el diseñador de los objetos siempre conoce la mejor forma para presentar la información, aunque esto último sea difícil de llevar a cabo dado que una estructura de objetos no será suficiente para conocer la forma apropiada de su presentación ante un usuario.

¹⁰ En el caso práctico, se muestran algunos diseños de vistas de salida del sistema de información, p.46

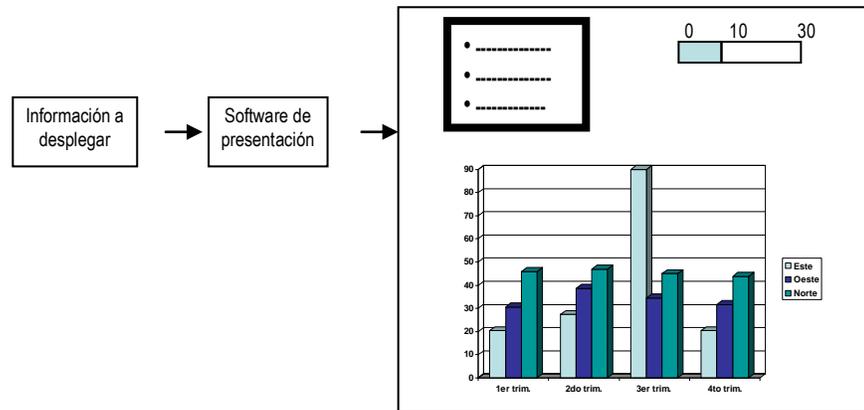


Figura 5.3.3 Vistas de salidas de datos

Se puede cambiar la presentación sobre la pantalla del usuario sin necesidad de cambiar el sistema de cómputo subyacente.

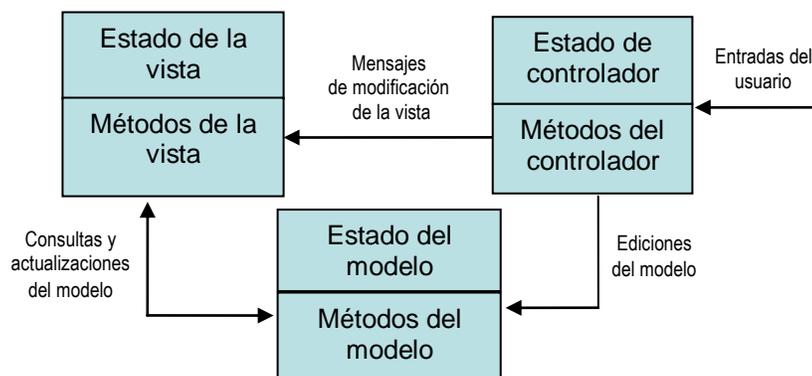


Figura 5.3.4 Esquema de salida de vistas

Cada vista tiene un objeto controlador asociado que maneja las entradas del usuario y la interacción de los dispositivos. Por lo tanto, un modelo que representa datos numéricos puede tener una vista que represente los datos como un histograma y una vista que represente los datos como una tabla. El modelo se edita acortando las barras en el histograma.

Para encontrar la mejor presentación de la información es necesario que el diseñador tome en cuenta los siguientes factores:

- ¿El usuario está interesado en información precisa o en las relaciones entre los diferentes valores de los datos?
- ¿Qué tan rápido cambian los valores de la información?, ¿Se indicarán de forma inmediata al usuario los cambios en un valor?

- ¿El usuario debe llevar a cabo una acción en respuesta a los cambios de la información?
- ¿El usuario necesita interactuar con la información desplegada vía una interfaz de manipulación directa?
- La información que se va a desplegar es ¿textual o numérica o ambas?, ¿son importantes los valores relativos de los elementos de la información?

Cuando se tienen que presentar grandes cantidades de información se pueden utilizar las visualizaciones abstractas para vincular elementos de los datos relacionados. Por ejemplo:

- Información climática. Lo cual se puede mostrar mediante el uso de mapas con isóbaras de frentes climáticos
- Estado de una red telefónica, se despliega gráficamente como un conjunto de nodos en un centro de administración de red.
- Un modelo de una molécula, la cual se despliega y se manipula en 3D utilizando realidad virtual.
- Un conjunto de páginas Web, se despliega como un árbol de decisiones

Tecnología de salida.

- Impresoras. Debe ser confiable, compatible con el software y hardware, además de contar con soporte técnico del fabricante
- Impresora especializadas, para la impresión de etiquetas, códigos de barras, logotipos o íconos significativos.
- Salidas a pantalla, como una vista preliminar lo cual me permite revisar antes de imprimir, o bien manejar este tipo de salida como una consulta debido a que la información no se requiere impresa.
- Audio, Archivos .WAV, menús bancarios, presentaciones con video y sonido.
- CD ROM, puertos USB, sin embargo se requiere de hardware especializado.
- Salida electrónica. correo electrónico, intercambio de mensajes, fax.

Es importante que antes de pensar en comprar algún otro dispositivo para complementar la salida de información del sistema se realicen las siguientes preguntas:

- ¿Cuántas personas usará la salida?
- ¿Con que frecuencia se necesita dicha salida?
- ¿Cuáles son los costos iniciales, cuales los de mantenimiento y cuales los de consumo del equipo que se solicita en la salida de información?
- ¿Cuáles son los requerimientos ambientales para su mejor conservación (absorción de ruido, temperatura, espacio para el equipo y cableado)?

El soporte al usuario debe cubrir tres áreas: Mensajes de error, Sistema de ayuda en línea y Documentación del usuario. Es conveniente involucrar a escritores profesionales, artistas gráficos con el fin de elevar la calidad del proceso de interfaz

Mensajes de error. Factores de diseño en la redacción de mensajes

- Contexto. De acuerdo a la acción será el mensaje de salida
- Experiencia. Conforme el usuario se vaya familiarizando con el sistema se incrementa la molestia por leer mensajes largos y los principiantes tienen dificultades de entender mensajes concisos
- Nivel de habilidad. Los mensajes se deben ajustar a las habilidades del usuario, así como a su experiencia
- Estilo. Los mensajes deben ser positivos nunca negativos, deben estar escritos en estilo activo no en pasivo. No deben ser insultantes o tratar de ser chistosos.

Por ejemplo, suponga que el usuario del sistema es una enfermera en una sala de cuidados intensivos de un hospital. La supervisión del paciente se lleva a cabo por un sistema de cómputo. Para ver el estado actual del paciente, el usuario selecciona desplegar e introduce el nombre del paciente

Figura 5.3.5 Captura de información

Figura 5.3.6 Mensaje de error orientado al sistema

Figura 5.3.7 Mensaje de error orientado al usuario

Dado que el nombre es incorrecto se genera un mensaje de error, el primer mensaje está más diseñado porque es negativo, no se ajusta a las habilidades y al nivel del usuario y no toma en cuenta la información del contexto. No sugiere como identificar la situación

El segundo es positivo, indica que el problema fue el sistema y no el usuario, identifica el problema en términos entendibles para la enfermera y como darle solución.

Diseño del sistema de ayuda

Los problemas con los sistemas de ayuda surgen cuando los usuarios entran a la red (una estructura compleja donde cada marco de información se vincula con otros marcos de información) después de cometer el error, entonces navegan hasta que en poco tiempo se encuentran perdidos incitados a abandonar la sesión e iniciar nuevamente

El texto de un sistema de ayuda se prepara en con el apoyo de especialistas en la aplicación. El marco de ayuda no es sólo reproducir el manual del usuario. El texto mismo, su exposición y estilo deben diseñarse para asegurar que sea legible en una ventana relativamente pequeña.

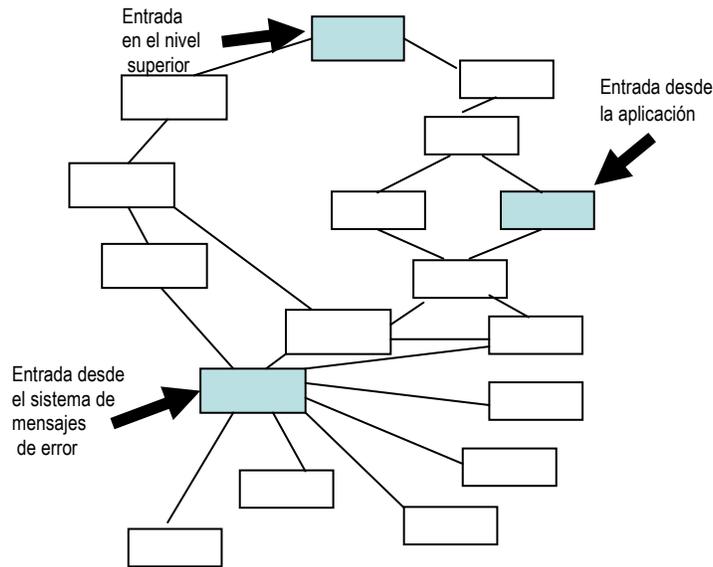


Figura 5.3.8. Red de marco de ayuda

Un marco de ayuda es relativamente corto, si éste se encuentra sin saturación de información con opciones a solicitar más en caso de requerirlo, o bien mostrar temas a fines. Aquí se pueden ocupar colores para resaltar la posición donde se encuentra el usuario o bien para seleccionar la opción deseada.

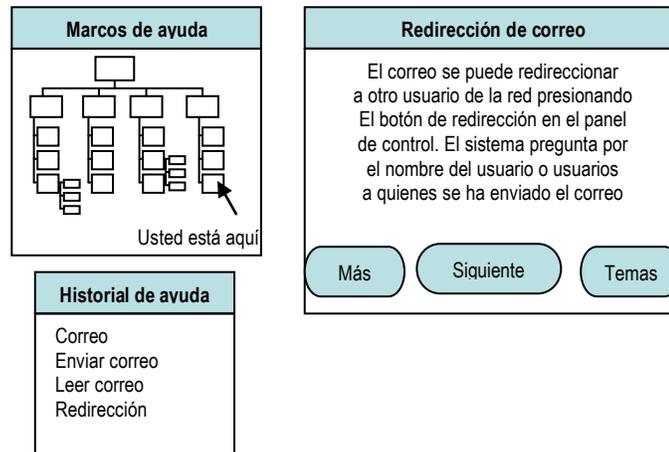


Figura 5.3.9 Otra opción de Red de marco de ayuda

Evaluación de la interfaz.

La evaluación de la interfaz se compara con la especificación de la usabilidad. Los diseñadores de interfaces tienen que utilizar su juicio y experiencia en la evaluación de las interfaces.

- Aprendizaje, cuanto tiempo tarda el usuario nuevo en ser productivo con el sistema.
- Velocidad de operaciones, que tan bien responde el sistema a las operaciones de trabajo del usuario
- Robustez, que tan tolerante es el sistema a los errores del usuario
- Recuperación, que tan bien se recupera el sistema a los errores del usuario
- Adaptación, que tan atado esta el sistema a un solo modelo de trabajo.

Los resultados de dicha evaluación se realizan a través de la aplicación de cuestionarios, entrevistas y observación, así mismo se pueden realizar grabaciones para después estudiarlas en sesiones privadas.

Tema 5.4 Criterios ergonómicos en las aplicaciones

Objetivo Particular

El alumno conocerá los diversos criterios ergonómicos para las aplicaciones en sistemas de información instalados en computadoras personales o en sitios de internet, mediante diversos ejemplos

Mapa Conceptual



Definición

La ergonomía es básicamente una tecnología de aplicación práctica e interdisciplinaria, fundamentada en investigaciones científicas, que tiene como objetivo la optimización integral de Sistemas Hombres-Máquinas, los que estarán siempre compuestos por uno o más seres humanos cumpliendo una tarea cualquiera con ayuda de una o más "máquinas" (definimos con ese término genérico a todo tipo de herramientas, máquinas industriales propiamente dichas, vehículos, computadoras, electrodomésticos, etc.). Al decir optimización integral queremos significar la obtención de una estructura sistémica (y su correspondiente comportamiento dinámico), para cada conjunto interactuante de hombres y máquinas, que satisfaga simultánea y convenientemente a los siguientes tres criterios fundamentales:

- Participación: de los seres humanos en cuanto a creatividad tecnológica, gestión, remuneración, confort y roles psicosociales.
- Producción: en todo lo que hace a la eficacia y eficiencia productivas del Sistema Hombres-Máquinas (en síntesis: productividad y calidad).
- Protección: de los Subsistemas Hombre (seguridad industrial e higiene laboral), de los Subsistemas Máquina (siniestros, fallas, averías, etc.) y del entorno (seguridad colectiva, ecología, etc.).

Este paradigma de las "3 P" se puede interpretar muy gráfica y sencillamente con la imagen de un trípode que sostiene a un Sistema Hombre-Máquina optimizado ergonómicamente; si a ese trípode le faltase aunque más no fuese una de sus tres patas (o sea que estuviese diseñado considerando únicamente a dos cualesquiera de las 3 P enunciadas arriba), todo se vendría al suelo (no se cumpliría la optimización ergonómica pretendida en el diseño). La amplitud con que se han fijado estos tres criterios requiere, para su puesta en práctica, de la integración de diversos campos de acción que en el pasado se desarrollaban en forma separada y hasta contrapuesta. Esos campos de acción eran principalmente:

- Mejoramiento del ambiente físico de trabajo (confort e higiene laboral).
- Diseño de herramientas, maquinarias e instalaciones desde el punto de vista del usuario de las mismas.
- Estructuración de métodos de trabajo y de procedimientos en general (por rendimiento y por seguridad).
- Selección profesional.
- Capacitación y entrenamiento laborales.
- Evaluación de tareas y puestos.
- Psicología industrial (y, con más generalidad, empresarial).

Naturalmente, una intervención ergonómica considera a todos esos factores en forma conjunta e interrelacionada. Además, se ha desarrollado desde hace ya un tiempo una ampliación del concepto ergonómico, dando lugar a la "macroergonomía", la que es conceptualizada como la optimización ergonómica de los Sistemas Hombres-Máquinas desde el punto de vista organizacional y últimamente se encuentra en pleno desarrollo la "ecoergonomía", ampliando aún más el campo de la optimización ergonómica. Para practicar la ergonomía se necesita, por lo tanto, poseer una buena capacidad de relación interdisciplinaria, un agudo espíritu analítico, un alto grado de síntesis creativa, los imprescindibles conocimientos científicos y, sobre todo, una firme voluntad de ayudar a los trabajadores para lograr que su labor sea lo menos penosa posible y que produzca una mayor satisfacción tanto a ellos mismos como a la sociedad en su conjunto.

La palabra ERGONOMÍA se deriva de las palabras griegas "ergos", que significa trabajo, y "nomos", leyes; por lo que literalmente significa "leyes del trabajo", y podemos decir que es la actividad de carácter multidisciplinar que se encarga del estudio de la conducta y las actividades de las personas, con la finalidad de adecuar los productos, sistemas, puestos de trabajo y entornos a las características, limitaciones y necesidades de sus usuarios, buscando optimizar su eficacia, seguridad y confort.

Criterios de ergonomía

A continuación se presenta una lista no exhaustiva de los criterios fundamentales de ergonomía que debe tener en cuenta un sitio web:

Criterios	Subcriterios	Descripción
Sobriedad 	Simplicidad	Un sitio web bien organizado reforzará la credibilidad de una organización.
	Sin sobrecargar	No se recomiendan las imágenes animadas. Estas deben reservarse para mensajes importantes ya que llaman la atención del usuario.

Criterios	Subcriterios	Descripción
Legibilidad 	Claridad	Se debe tener en cuenta que la información escrita es más difícil de leer en la pantalla que en el papel (lleva un 25 por ciento más de tiempo). Por lo tanto, el texto debe ser lo suficientemente espaciado.
	Estructuración	El texto debe estructurarse en párrafos y títulos en distintos niveles para facilitar su lectura.
	Organización	Las distintas partes de la información se deben organizar de acuerdo al nivel de importancia. Las partes más importantes de la información deben ubicarse en la parte superior de la página.
Funcionabilidad 	El sitio debe ser fácil de navegar	La "regla de los tres clics," ampliamente seguida, estipula que se debe poder acceder a toda la información en menos de tres clics.
	Ubicación	El visitante debe poder ubicar en todo momento el lugar del sitio en el que se encuentra. Además, el logotipo debe estar ubicado en el mismo lugar en todas las páginas y se debe aplicar un mismo estatuto gráfico que rija todo el conjunto para que el usuario sepa que todavía se encuentra en el mismo sitio web. Se recomienda ofrecer un mapa del sitio para ayudar al visitante a ubicarse.
	Libertad al navegar	El sitio web debe darle al usuario la posibilidad de regresar a la página de inicio y a los principales encabezados con un simple clic, sin importar en qué página se encuentre (por ejemplo, al usar la barra de navegación). En lo posible, el usuario debe poder navegar todo el contenido sin tener que regresar a la página de inicio. Se deben evitar las páginas de introducción y las pantallas de presentación porque molestan a la mayoría de los visitantes.
	Visibilidad de la dirección	La URL de la página debe estar siempre visible y clara para que los usuarios sepan dónde se encuentran y puedan volver a dicha página.
	Materialización de la información	La información en un sitio web debe ser calificada, es decir, que algunos datos como la fecha de la última actualización y el nombre del autor, deben estar presentes. Además, no es recomendable que un sitio web posea una página de inicio llamativa que reenvíe a páginas que se encuentren en construcción.
	Homogeneidad estructural	Dentro de lo posible, los elementos que se utilicen para la navegación deben estar ubicados en el mismo lugar en cada página y deben tener la misma presentación de una página a la otra.

Crterios	Subcriterios	Descripción
Velocidad 	Tiempo de carga	El tiempo necesario para que una página se cargue debe ser lo más corto posible porque la mayoría de los usuarios de Internet no esperarán durante más de 15 segundos que se complete este proceso. Este factor depende en particular de la conexión con que cuente el usuario, el tamaño de la página y de sus imágenes y las capacidades del servidor web.
	Imágenes optimizadas	Es recomendable optimizar al máximo posible el tamaño de las imágenes eligiendo un formato de buena adaptación y la menor cantidad de colores posible. Se recomienda que cada imagen no exceda los 30 o 40 kb, a menos que el usuario solicite otra cosa por su propia voluntad (por ejemplo, descargar una imagen de alta resolución).
	Imágenes sin cambio de tamaño	Por la misma razón, el tamaño de las imágenes no debe cambiarse.
Interactividad 	Hipervínculos	La interactividad describe las interacciones posibles entre los usuarios y el sitio web. Los hipervínculos ofrecen a los usuarios grandes posibilidades en este campo y proporcionan a los visitantes rutas variadas que pueden elegir a su gusto. Se les aconseja a los diseñadores web que coloquen vínculos suficientes entre las diferentes páginas. Sin embargo, demasiados vínculos pueden dificultar la lectura.
	Segmentación de la información	Es una buena idea segmentar la información para que los visitantes la incorporen con mayor facilidad y, en algunos casos, para despertar su curiosidad. Puntualmente, la segmentación se puede realizar a través de la identificación bibliográfica y de copyright de la contribución y un hipervínculo que dirija al resto del artículo.
	Facilitación de los intercambios	Se recomienda configurar todo para que la comunicación con los visitantes sea más sencilla, en especial para reunir sus impresiones y solicitudes con el fin de adaptar el sitio web a sus necesidades. Como mínimo, debe ser fácil para los visitantes poder contactar a alguien, ya sea a través del correo electrónico o completando un formulario de contacto.
Adaptabilidad 		La adaptabilidad se define como la posibilidad de personalizar un sitio web por medio de la intervención de los usuarios.
	Cambio de tamaño de fuente	Los textos de los sitios web deben tener fuentes cuyo tamaño no esté expresado en valores absolutos. De esta manera los usuarios podrán cambiar el tamaño de fuente si lo desean.

Criterios	Subcriterios	Descripción
Accesibilidad 	Acceso para todos	La accesibilidad significa poder acceder a un sitio web universalmente, es decir, que el sitio sea accesible para todo tipo de usuarios, incluso para las personas con ceguera parcial y discapacitados visuales. Existe una cierta cantidad de reglas de accesibilidad simples que deben seguirse para procurar el acceso a la mayor cantidad posible de personas, independientemente de la configuración de software o hardware que tengan o de sus discapacidades.
	Interoperabilidad	Seguir los estándares, en particular las recomendaciones de accesibilidad del W3C , ayuda a garantizar un nivel óptimo de interoperabilidad, es decir, la capacidad que permite a los clientes que usan software diferentes acceder a un sitio web.
	Transparencia de formato	Es preferible que los formatos utilizados sean transparentes, es decir, que se puedan leer en modo de texto. Por eso se prefiere el uso de HTML en lugar de Flash. Además, las imágenes y animaciones no deben quitar mérito a la información escrita o constituir un obstáculo para la gente con discapacidad visual. Por lo tanto, las ilustraciones gráficas sólo deben usarse como complemento de los textos.
	Leyenda	La función de una leyenda o texto de reemplazo es reemplazar imágenes (por medio del atributo <i>alt</i>) para posibilitar a los minusválidos visuales entender el significado de una imagen.
	Elección de los colores	Los colores deben elegirse de modo que las personas daltónicas puedan distinguirlos correctamente.
	Uso adecuado de las hojas de estilo	La información debe ser accesible aún sin una hoja de estilo.
	Contraste adecuado	Debe existir bastante contraste entre las imágenes de fondo y el texto para que personas con ceguera parcial puedan leerlo.
	Tamaño de fuente modificable	El tamaño de fuente debe poder adaptarse para que los usuarios puedan agrandarlos de ser necesario. Además, no debe emplearse una fuente pequeña, para evitar la fatiga visual de los lectores y para que el texto resulte legible.

Unidad 6. Verificación, validación y pruebas del software

Tema 6.1 Visión general

Objetivo Particular

Al término de la unidad, el alumno identificará la mejor forma de poner a prueba un sistema de información, y reconocerá la importancia de los datos estadísticos que una prueba de software puede dar con el fin de incrementar la confiabilidad en el uso del software desarrollado.

Mapa conceptual

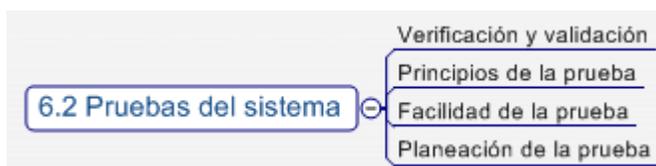


Tema 6.2 Pruebas del sistema

Objetivo Particular

Al término del tema, el alumno reconocerá la importancia de la planeación de pruebas y las características de una prueba efectiva, mediante diferentes formas de llevarla a cabo.

Mapa conceptual



Verificación y validación

La verificación y validación son los procesos de comprobación y análisis que asegura que el software este acorde con su especificación y necesidades del cliente. Dichos procesos forman parte del ciclo de vida de un sistema de información.

Este proceso inicia con las revisiones de los requerimientos y continúa con las revisiones del diseño y las inspecciones de código hasta la prueba del producto.

Boehm (1979) expresó la diferencia entre verificación y validación, y de acuerdo a él, dijo que la validación responde a la pregunta ¿estamos construyendo el producto correcto?, en tanto que la verificación responde a ¿estamos construyendo el producto correctamente?. La validación es un proceso general, lo cual nos permite asegurar que el sistema cumpla con las expectativas del cliente, es decir va más allá de comprobar requerimientos. En tanto que la verificación representa que el sistema cumpla los requerimientos funcionales y no funcionales especificados.

Existen dos técnicas de comprobación y análisis de sistemas, las inspecciones del software y las pruebas de software. Las inspecciones del software analizan y comprueban las representaciones del sistema como el documento de requerimientos, diagramas de diseño y código fuente del programa. Las inspecciones del software y los análisis automatizados son técnicas de V&V estáticas dado que no requieren que el sistema se ejecute.

6. Verificación, validación y pruebas del software

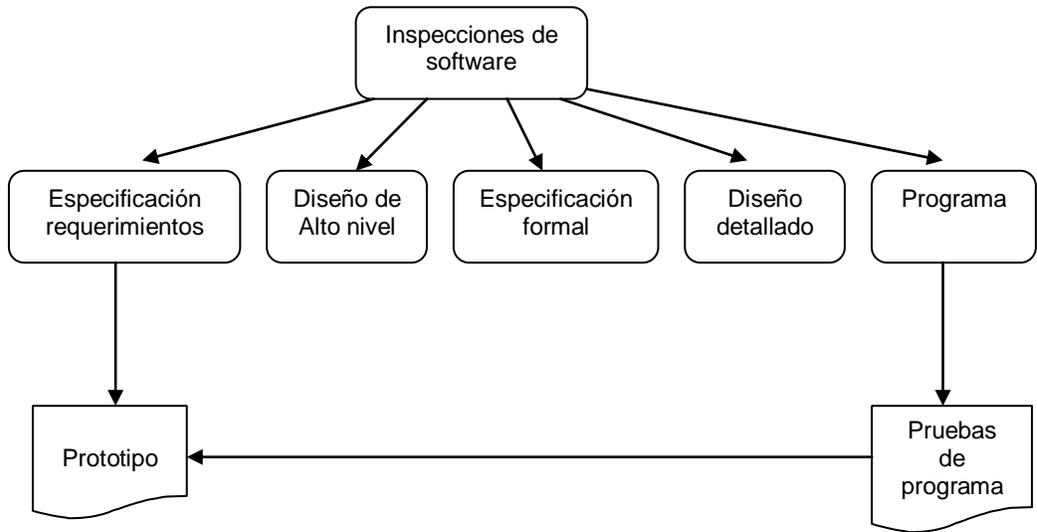


Figura 6.2.1 Inspecciones de software

Las pruebas de software consisten en llevar a cabo una implementación del software con los datos de prueba y examinar las salidas del software y su comportamiento operacional. Es una técnica V&V dinámica dado que se llevan a cabo en una representación ejecutable del sistema.

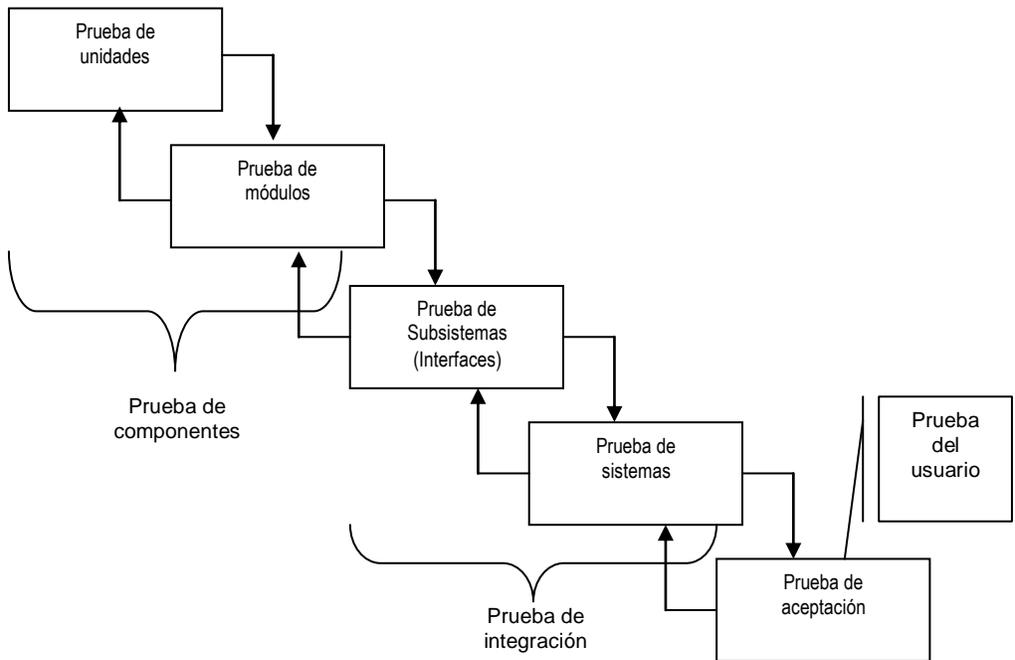


Figura 6.2.2 Pruebas de software

Así mismo también existe diferencia entre las técnicas de V&V y la depuración (Figura 1), ya que mientras la V&V es un proceso que establece la existencia de defectos en un sistema de software, la depuración es un proceso que localiza y corrige estos defectos.

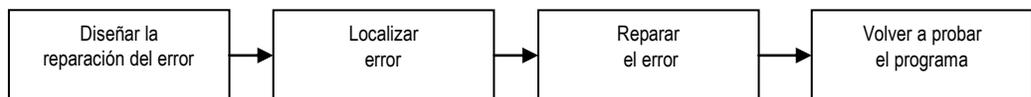


Figura 6.2.3 Depuración

Localizar una falla no es un proceso sencillo, dado que en ocasiones no necesariamente se localiza cerca del punto donde falló el programa. Sin embargo el responsable de la depuración tiene que diseñar pruebas adicionales que repitan la falla original y ayuden a determinar la falla en el programa.

Existen herramientas interactivas de depuración que generalmente el compilador ya incluye y hace más fácil la ejecución de la depuración.

Después de descubrir el defecto, se corrige y se evalúa, lo que implica volver a inspeccionar el programa o repetir las pruebas anteriores lo que se llama “pruebas de regresión”. Sin embargo, este procedimiento llega a ser costoso para proyectos grandes, por lo que la opción es planear las pruebas.

La inspección de software solo comprueba la correspondencia entre un programa y su especificación (verificación) no pueden demostrar que el software sea de utilidad ni comprobar las características no funcionales de software, como su desempeño y fiabilidad.

La meta fundamental de V&V es generar confianza sobre el software, de tal forma que éste se ajusta a los propósitos para los que fue hecho. Sin embargo, no significa estar libre de errores, más bien que el sistema es suficientemente bueno para la utilización que se pretende.

Este nivel de confianza depende:

- El propósito o función del software
- Las expectativas del cliente o usuario
- El entorno del mercado actual (tener prevista la competencia, el precio, tiempos de instalación)

Principios de la prueba

- A todas la pruebas se les debería hacer un seguimiento hasta los requisitos del cliente
- Las pruebas deberían planificarse mucho antes de que empiecen (tan pronto como esten los modelos del diseño)
- La prueba debería empezar por lo pequeño y progresar hacia lo grande, es decir, las primeras pruebas se centran en módulos individuales y conforme avanza la prueba se sigue en interfaces y luego en el sistema entero
- No son posibles las pruebas exhaustivas, debido a que el numero de permutaciones de caminos, incluso para un programa moderado, es excepcionalmente grande
- Para ser más efectiva, las pruebas deben realizarlas o conducir las un equipo independiente.

Cada uno de los puntos anteriores quedan ilustrados en la figura 4, sobre el proceso de ejecución de la planeación de pruebas.

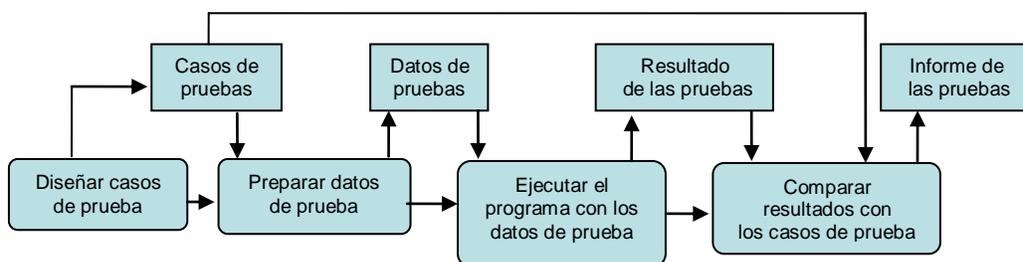


Figura 6.2.4 Proceso de ejecución de la planeación de pruebas

Facilidad de la prueba

La prueba representa una interesante contradicción para el ingeniero de software, debido a que éste intenta construir algo abstracto, y en la prueba se intenta demoler lo ya construido

Las pruebas exhaustivas, donde se prueba cada posible secuencia de ejecuciones del programa, no son prácticas. Las organizaciones deben desarrollar políticas para los posibles casos de prueba, en lugar de dejar esta tarea al equipo de desarrollo. Por ejemplo, probar todas las funciones del sistema que se accesen mediante menús, probar combinaciones de funciones (por ejemplo dar formato al texto), accesadas por menús, las funciones deben probarse con datos de entrada correctos e incorrectos.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces, no debe ser redundante, no debe ser ni demasiado sencilla, ni demasiado compleja. Por lo tanto una prueba de defectos exitosa es una prueba que provoca que el sistema se desempeñe incorrectamente y así exponer un defecto.

Algunas características que permiten a un software fácil de probar son:

- Operatividad. Cuanto mejor funcione, más eficientemente se puede probar. El sistema tiene pocos errores, ningún error bloquea la ejecución de las pruebas, el producto evoluciona en fases funcionales.
- Observabilidad. Lo que ves es lo que pruebas. Se genera una salida distinta para cada prueba, los estados y variables del sistema están visibles, un resultado incorrecto se identifica fácilmente, se informan de los errores internos, el código fuente es accesible
- Controlabilidad. Cuanto mejor podamos controlar el software, más se puede automatizar y optimizar. Todos los resultados posibles se pueden generar a través de alguna combinación de entrada. El ingeniero de prueba puede controlar directamente los estados y las variables de hardware y software. Los formatos de entrada y resultados son consistentes y estructurados. Las pruebas pueden especificarse, automatizarse y reproducirse convenientemente
- Capacidad de descomposición. Controlando el ámbito de las pruebas, podemos aislar más rápidamente los problemas y llevar a cabo mejores pruebas de regresión. El sistema de software está construido con módulos independientes. Los módulos de software se pueden probar independientemente.

- Simplicidad. Cuanto menos haya que probar, más rápidamente podemos probarlo. Simplicidad funcional, es un conjunto de características mínimas para cumplir los requisitos. Simplicidad modular, arquitectura modular. Simplicidad de código, es un estándar para facilitar la inspección y mantenimiento.
- Estabilidad. Cuantos menos cambios, menos interrupciones a las pruebas. Los cambios de software son infrecuentes, están controlados, no invalidan las pruebas existentes. El software se recupera bien de los fallos.
- Facilidad de comprensión. Cuanta más información tengamos, más inteligentes serán las pruebas. El diseño se ha entendido perfectamente, las dependencias entre los componentes internos, externos y compartidos se han entendido perfectamente. Se han comunicado los cambios de diseño. La documentación técnica es instantáneamente accesible, bien organizada, específica, detallada y exacta.

Planeación de pruebas¹¹

La planeación de la V&V inicia en las primeras etapas del proceso de desarrollo. Así el diagrama anterior muestra como los planes de prueba se derivan de la especificación y diseño del sistema. Así también muestra como se dividen las actividades de verificación y validación en varias etapas, donde cada fase es conducida por pruebas definidas.

Los componentes principales de un plan de pruebas son:

- El proceso de prueba (como el diagrama anterior) es necesario visualizar las fases principales de una prueba
- Requerimientos de rastreo o seguimiento, probar de forma individual los requerimientos del usuario
- Elementos probados, donde se deben especificar los productos del proceso de software a probar.
- Calendarización de las pruebas, asignación de tiempo, personal, recursos (materiales, monetarios, etc.)
- Procedimiento de registro de las pruebas. No solo es ejecutar una prueba sino registrar los resultados, con opción a auditar las pruebas para corroborar que se han llevado a cabo correctamente.
- Requerimiento de hardware y software, donde se señalan las herramientas de software
- Restricciones. Se anticipan restricciones para ejecutar la prueba como el de escasez de personal.

¹¹ En el caso práctico (CD) se muestran algunas estrategias y planes de pruebas, p. 56

6. Verificación, validación y pruebas del software

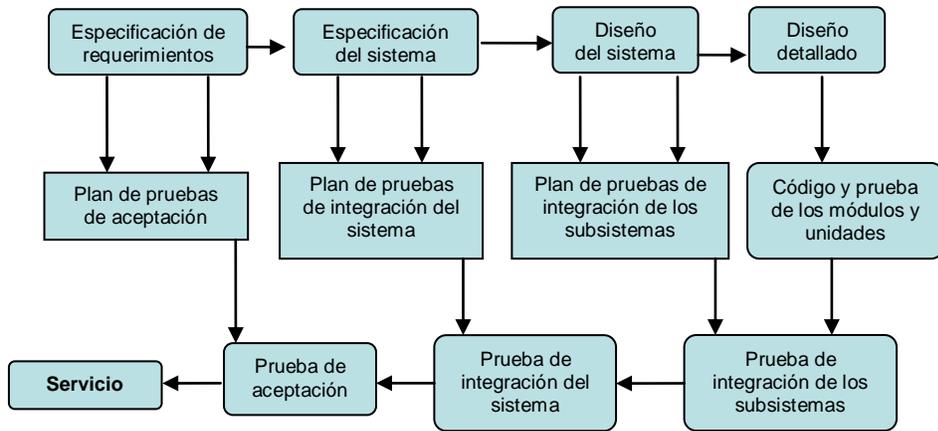


Figura 6.2.5 Plan de pruebas

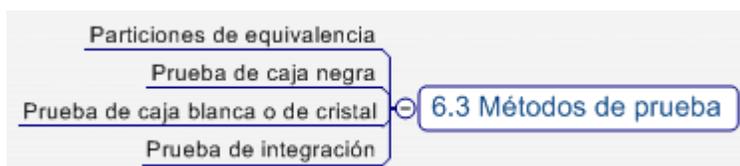
El plan de pruebas no es un documento estático. Se revisa regularmente ya que llevar a cabo las pruebas es una actividad que depende de la implementación.

Tema 6.3 Métodos de prueba

Objetivo Particular

Al término del tema, el alumno elaborará diversos tipos de prueba que permitan garantizar la calidad del software y representar la revisión final del diseño y la codificación.

Mapa conceptual



Particiones de equivalencia

Las particiones de equivalencia se basa en clasificar los diferentes datos de entrada a un programa, como números positivos, negativos, cadenas sin espacios en blanco, etc. Como particiones de equivalencia o dominio donde normalmente el programa se comporta de la misma forma.

Los casos de prueba se diseñan de tal forma que las entradas o salidas pertenezcan a estas particiones. Las particiones de equivalencia son conjuntos de datos donde todos los miembros del conjunto se procesan en forma equivalente, particiones de entrada válidas y no válidas, así como particiones de salida.

Una vez identificado el conjunto de particiones se definen los casos de prueba considerando los límites de las particiones y casos cercanos al punto medio de la partición. Dado que los programadores en su momento sólo probaron los procesos con datos típicos cercanos al punto medio de la partición, cuando los valores en los límites se consideran atípicos y son éstos los que logran el objetivo de la prueba, entonces se encuentra un defecto o error.

Las particiones de equivalencia se identifican utilizando la especificación del programa o documentación del usuario, y el probador usará su experiencia para predecir el tipo de valores de entrada que pueden causar errores.

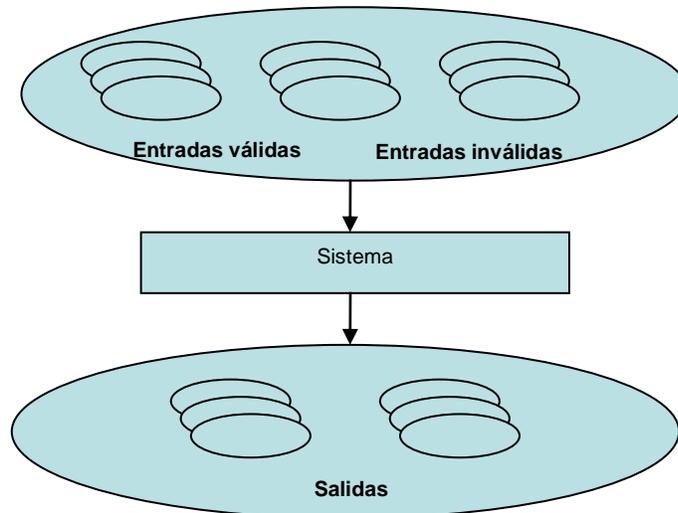


Figura 6.3.1 Particiones de equivalencia

Por ejemplo, algunos lineamientos de prueba en un proceso de búsqueda son:

- Probar el software con secuencias que solo tienen un único valor
- Utilizar varias secuencias de diferentes tamaños en diversas pruebas
- Generar pruebas de tal forma que se puedan acceder al primer elemento, el que está en medio y el último de la secuencia

Prueba de caja negra

Algunas pruebas de defectos son las pruebas de caja negra y particiones de equivalencia. Una prueba de defectos tiene como objetivo exponer los defectos latentes en un software antes de entregar el sistema.

La prueba de caja negra también es conocida como “pruebas funcionales”. Estas pruebas consisten en ver al sistema como una caja negra, cuyo comportamiento solo se puede determinar estudiando las entradas y salidas relacionadas, no se mete con el código. Si las salidas no son las previstas, entonces la prueba ha detectado exitosamente un problema con el software.

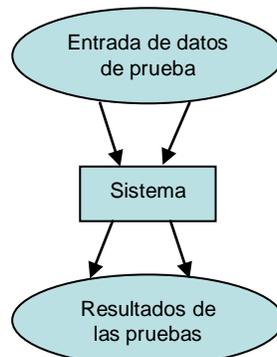


Figura 6.3.2 Caja negra

Prueba de caja blanca o caja de cristal.

Es conveniente aclarar que la prueba de caja negra busca conocer la función específica para la que fue diseñado el producto, en cambio la prueba de caja de cristal busca conocer el funcionamiento del producto y desarrollar la prueba que asegure que todas las piezas encajan.

La prueba de caja blanca tiene por objetivo, que por lo menos una vez se ejerciten las siguientes actividades:

- Todos los caminos independientes
- Las decisiones lógicas en sus vertientes falso o verdadero
- Los bucles en sus límites y con sus límites operacionales
- Asegurar la validez de las estructuras internas de datos

Éstas pruebas también se conocen como pruebas estructurales, y consiste en analizar el código, el cual puede dar a conocer el número de casos de prueba necesarios para garantizar que todas las instrucciones del programa o componente se ejecuten al menos una vez durante el proceso de prueba.

Este tipo de prueba es necesaria debido a la naturaleza misma de los defectos de software:

- Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de ejecutar un camino de programa
 - A menudo se cree que un camino lógico tiene pocas probabilidades de ejecutarse cuando de hecho se puede ejecutar de forma normal
 - Los errores tipográficos son aleatorios, errores de dedazo, no lo sabes hasta que corres el programa

En este tipo de prueba, toman un papel importante las particiones de equivalencias, dado que de acuerdo al trato de éstas se recomendará un caso de prueba con algún orden de entrada de datos o restricciones avisadas, etc.

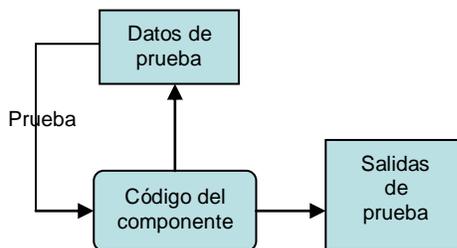


Figura 6.3.3 Caja de cristal

Pruebas de trayectorias

Las pruebas de trayectoria son una estrategia de prueba estructural, cuyo objetivo es probar cada trayectoria de ejecución independiente en un componente de programa. Así mismo permite visualizar la ejecución de procesos al menos una vez. El punto inicial de las pruebas de trayectoria es un diagrama de flujo de programa.

La elaboración de dichos diagramas se basan en diversas notaciones como se muestra en la figura 9, que permiten ver la ejecución de un procedimiento.

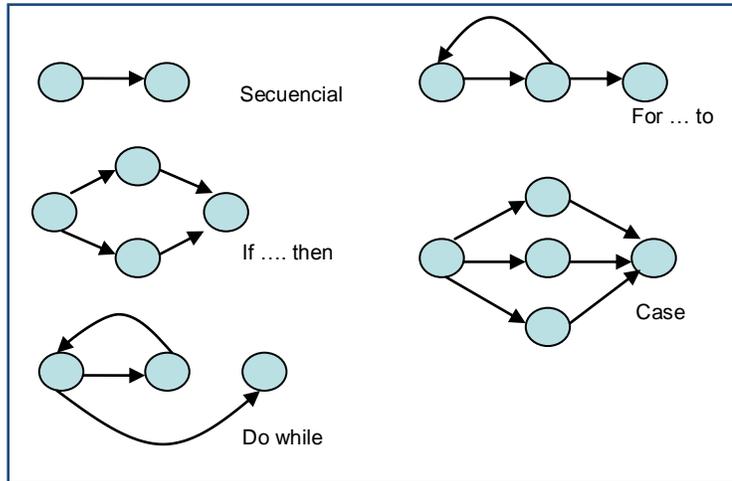


Figura 6.3.4 Notación de grafos para pruebas

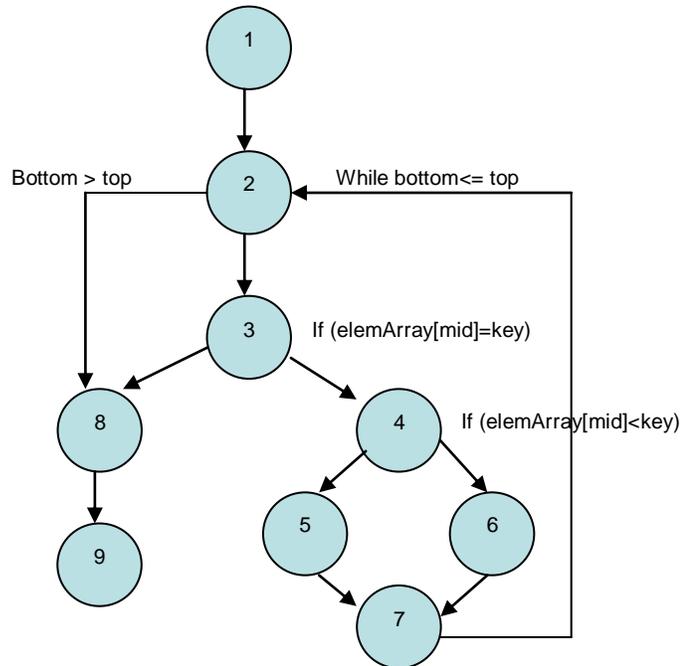


Figura 6.3.5 Camino básico de prueba

Conjunto de caminos básicos

C1 = 1,2,3,8,9

C2 = 1,2,8,9

C3 = 1,2,3,4,5,7,2,8,9

C4 = 1,2,3,4,6,7,2,8,9

Por otra parte para obtener casos de prueba, podemos seguir las siguientes recomendaciones:

- Usando el diseño o el código como base, dibujamos el correspondiente grafo de flujo
- Determinamos un conjunto básico de caminos linealmente independientes
- Preparamos los datos de prueba que forzarán la ejecución de cada camino básico
- Determinamos la complejidad ciclomática del grafo de flujo resultante (Opcional)

Los datos se escogen de tal forma que las condiciones de los nodos estén adecuadamente establecidas con el fin de comprobar cada camino.

Matriz de grafo

Es una matriz cuadrada, donde sus filas y columnas son el número de nodos y las entradas corresponden a las conexiones (aristas)

A esta matriz se le puede agregar un peso de enlace, es decir, una ponderación que nos indique el valor de la conexión o enlace con el fin de aprovechar esta información para interpretaciones del mismo o de otros niveles (operativo, decisión, enlace, etc.). Así entonces, se le puede asignar propiedades como:

- Probabilidad de que un enlace (arista) sea ejecutado
- El tiempo de procesamiento asociado al recorrido de un enlace
- Memoria requerida durante el recorrido
- Recursos requeridos durante el recorrido

Por ejemplo la interpretación mas sencilla es asignar un 0 que signifique “no existe conexión” y un 1 donde “sí existe conexión”.

Cuadro 6.3.1 Matriz de conexiones

Nodo	1	2	3	4	5
1	—	—	1, a	—	—
2	—	—	—	—	—
3	—	1, d	—	1, b	—
4	—	1, c	—	—	1, f
5	—	1, g	1, e	—	—

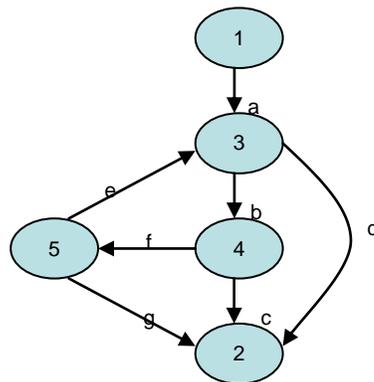


Figura 6.3.6 Camino básico de prueba

Algunos métodos de prueba de comportamiento que pueden hacer uso de los grafos son:

- Modelación del flujo de transacción (interfaces), validación de consultas, disponibilidad de la información, procesamiento para la generación de reportes, etc.
- Modelación de estado finito, representa los estados observables por el usuario, por ejemplo: entrada-información-factura-cliente
- Modelado de flujo de datos, son transformaciones para convertir un objeto de dato en otro, por ejemplo: El impuesto retenido se calcula de sueldo bruto usando la relación de $TIR = 0.062 * SB$
- Modelación de planificación, donde los nodos representan los programas a elaborar, las aristas las conexiones secuenciales que existe entre dichos programas y el peso de enlace son los tiempos de ejecución requerido.

Cada relación es estudiada separadamente de manera que se puedan obtener casos de prueba. Así mismo es importante estudiar la transitividad para estudiar como se propaga el impacto de las relaciones entre los objetos:

X es necesario para calcular Y
Y es necesario para calcular Z
Entonces X es necesario para calcular Z

Una relación simétrica de prueba, busca demostrar que es bidireccional. Una relación reflexiva permite asegurar que hay un bucle nulo presente.

Pruebas de integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que al mismo tiempo, se lleva a cabo pruebas para detectar errores asociados con la interacción. El objetivo es escoger los módulos probados en unidad y construir una estructura del programa que esté de acuerdo con lo que dicta el diseño.

A menudo hay una tendencia a intentar una integración no incremental; es decir, a construir el programa mediante un enfoque de “big bag”. Se combinan todos los módulos por anticipado. Se prueba todo el programa en conjunto. ¡Normalmente se llega al caos! Se encuentra un gran conjunto de errores. La corrección se hace difícil, puesto que es complicado aislar las causas al tener delante el programa entero en toda su extensión. Una vez que se corrigen esos errores aparecen otros nuevos y el proceso continúa en lo que parece un ciclo sin fin.

La integración incremental es la antítesis del enfoque del “big bag”. El programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y de corregir, es más probable que se puedan probar completamente las interfaces y se puede aplicar un enfoque de prueba sistemática.

Prueba de integración descendente

La integración descendente es un planteamiento incremental a la construcción de la estructura de programas. Se integran los módulos moviéndose hacia abajo por la jerarquía de control, comenzando por el módulo de control principal. Los módulos subordinados al módulo de control principal se van en la estructura, bien de forma Primero-en-profundidad, o bien de forma primero-en-anchura.

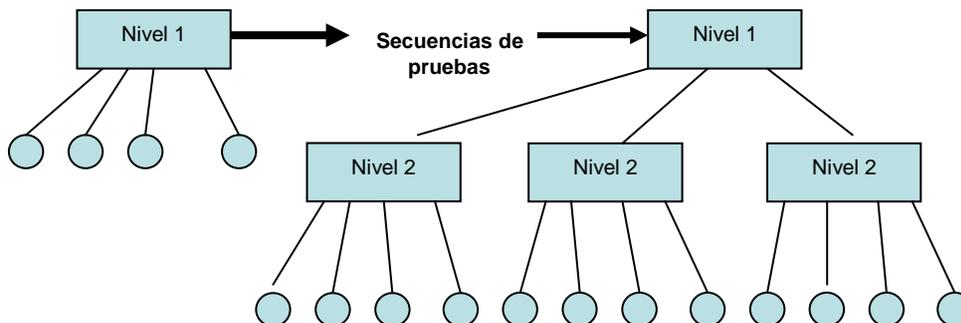


Figura 6.3.7 Prueba descendente

El proceso de integración se realiza en una serie de 5 pasos:

1. Se usa el módulo de control principal como controlador de la prueba disponible de resguardos par todos los módulos directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración elegido se van construyendo los resguardos subordinados uno a uno por los módulos reales.
3. Se llevan a cabo pruebas cada vez que se integran un nuevo módulo.
4. Tras terminar cada conjunto de pruebas, se reemplaza otro resguardo con el módulo real
5. Se hace la prueba de regresión para asegurarse de que no se han introducido errores nuevos.

Prueba de integración ascendente

La integración ascendente, como su nombre lo indica, empieza la construcción y la prueba con los módulos atómicos. Dado que los módulos se integran de abajo hacia arriba, el proceso requerido de los módulos subordinados a un nivel dados siempre están disponibles y se eliminan la necesidad de resguardos.

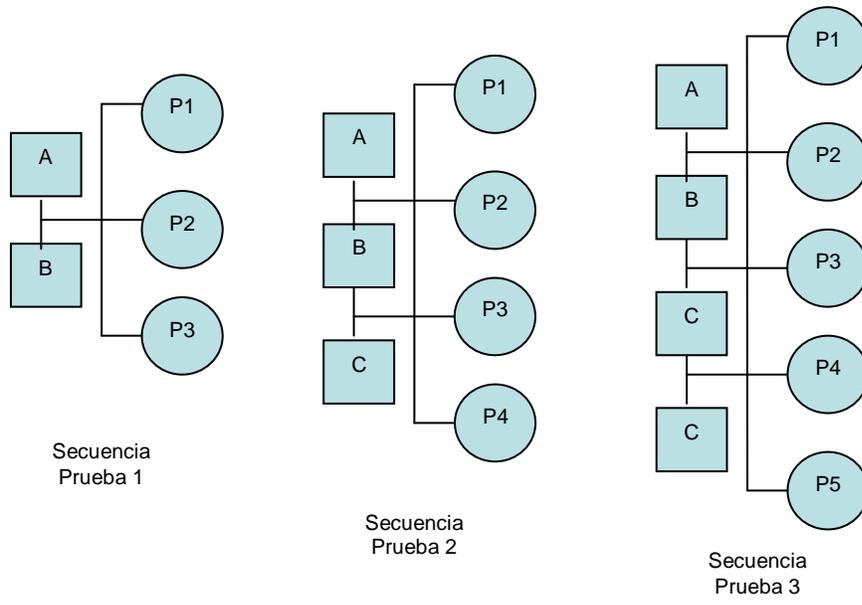


Figura 6.3.8 Prueba ascendente

Tema 6.4 Verificación formal

Objetivo Particular

Al término del tema, el alumno reconocerá las características de los sistemas de información críticos y la importancia de una verificación formal para elevar la confianza en su uso, en este tipo de desarrollos.

Mapa conceptual



Métodos formales

Existe un gran debate entre los sistemas críticos y los métodos formales, dado que varios desarrolladores mencionan que los métodos formales no son costeables y no ayudan a incrementar la confiabilidad del sistema.

Los métodos formales se ocupan en dos niveles de desarrollo de sistemas críticos:

- Se puede desarrollar y analizar matemáticamente una especificación formal del sistema para descubrir inconsistencias
- Se puede desarrollar una verificación formal de que el código de un sistema de software es consistente con la especificación

Con el primer enfoque se fuerza a un análisis detallado de la especificación, el cual puede tener inconsistencias que no se reflejan hasta que el sistema esté en operación, sin embargo no se pone en duda la confiabilidad. Este escenario va en contra de la especificación de la Ingeniería en Software dado que requiere de notaciones especializadas y los expertos en el dominio (de la empresa) no las comprenden. Así los problemas de requerimientos quedan ocultos por la formalidad.

Verificar un software no trivial requiere de tiempo, herramientas especializadas, tales como demostradores de teoremas y expertos en matemáticas. En caso de un incremento al tamaño del sistema, el costo de la verificación se incrementa de forma desproporcionada.

No existe garantía de que una especificación formal tenga menos anomalías, y por lo mismo tampoco existe garantía de que el software sea fiable para su utilización práctica:

- Las especificaciones pueden no reflejar los requerimientos reales de los usuarios del sistema
- Las pruebas pueden contener errores, por ser largos y complejos.

- La prueba puede asumir un patrón de utilización incorrecta, si el sistema no se utiliza como se estipuló las pruebas no son válidas

Validación de sistemas críticos

La naturaleza de los sistemas críticos es tal que por lo general es necesario incrementar el análisis normal y las pruebas con procesos adicionales que permitan producir evidencias para clasificar al sistema digno de confianza, por las siguientes razones:

- Los costos de las caídas. Es más costeable gastar más en verificación y validación antes de sufrir una caída con costos potencialmente grandes
- Validación de los atributos de la confiabilidad. Los clientes de un sistema crítico deben estar convencidos de que el sistema garantiza confianza: disponibilidad, fiabilidad, seguridad y protección.

Las pruebas de programas es aún la técnica de verificación y validación predominante. Existen dos tipos diferentes de pruebas que se utilizan en varias etapas del proceso del software, pruebas de defectos y pruebas estadísticas.

Las pruebas de defectos pretenden encontrar inconsistencias entre un programa y su especificación, los cuales se deben a fallas o defectos del programa. Este tipo de prueba se usa para revelar la presencia de un error más que para simular su utilización operacional

Las pruebas estadísticas se usan para probar el desempeño y fiabilidad del programa y comprobar como trabaja bajo una estimación de la fiabilidad operacional contando el número de caídas observadas en el sistema. El desempeño del sistema se realiza midiendo el tiempo de ejecución y el tiempo de respuesta, al procesar datos estadísticos.

Confiabilidad del software

La Ingeniería de fiabilidad se encarga de valorar completamente la fiabilidad, tomando en cuenta la probabilidad de caídas de los componentes de un sistema y la combinación de estos para afectar la fiabilidad total del sistema:

$$PS = P_A + P_B \quad \text{caídas dependientes}$$

$$PS = P_A^n \quad \text{caídas independientes}$$

Sin embargo y siguiendo los pasos mostrados en la figura 15, podemos calcular estadísticamente la fiabilidad del sistema, medido en el numero de caídas del mismo.

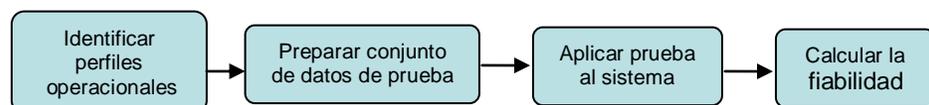


Figura 6.4.1 Fiabilidad del sistema

- Perfil operacional. Identifica las diferentes clases de entradas al sistema y la probabilidad de ocurrencia de éstas entradas en la utilización normal
- Conjunto de datos de prueba, se construye un conjunto de datos de prueba que reflejan el perfil operacional
- Aplicar pruebas. Se aplican las pruebas y se observa el número de caídas, los tiempos de caídas
- Calcular la fiabilidad. Después de observar el número de caídas importantes estadísticamente, se calcula la fiabilidad del software, en la cual se debe considerar tres dimensiones:
 - a) Fiabilidad del hardware, probabilidad de que un componente falle y cuanto tiempo toma reparar ese componente
 - b) Fiabilidad del software, probabilidad de que un componente de software produzca una salida incorrecta
 - c) Fiabilidad del operador, que tan probable es que el operador de un sistema cometa un error.

Verificación estadística

Debido a los altos costos de las caídas de los sistemas críticos, las técnicas de verificación y validación, tales como las de especificación y prueba formales que normalmente son más caras, son las adecuadas y costeables para sistemas críticos.

Las pruebas estadísticas se utilizan para estimar la fiabilidad del software. Se basa en probar el sistema con un conjunto de datos de prueba que reflejan el perfil operacional del software. Los datos de prueba se pueden generar de forma automática.

Sin embargo no debemos olvidar que implícitamente se busca cumplir con objetivos claros sobre la calidad del software como:

1. Los requisitos del software son la base de las medidas de la calidad, la falta de concordancia con los requisitos es una falta de calidad
2. Algunos estándares específicos definen un conjunto de criterios de desarrollo que guían la manera en que se hace la ingeniería del software, sino se siguen los criterios habrá poca calidad
3. Existe un conjunto de requisitos implícitos que ha menudo no se nombran (por ejemplo facilidad de mantenimiento). Si el software cumple con sus requisitos explícitos pero falla en los implícitos, la calidad del software no será fiable

Un modelo de crecimiento de la fiabilidad es un modelo que muestra cómo cambia la fiabilidad del sistema durante el proceso de prueba

Existen varios modelos derivados de los experimentos de fiabilidad en varios dominios de aplicación diferente

6. Verificación, validación y pruebas del software

El modelo más simple de crecimiento de la fiabilidad es el modelo escalonado, el cual supone que las reparaciones del software siempre se implementan de forma correcta por lo que el número de caídas asociadas decrecen con el tiempo, como se muestra en la figura 15.

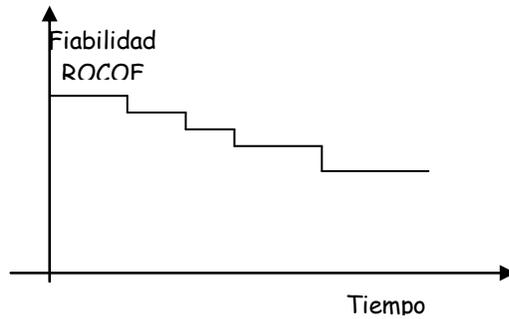


Figura 6.4.2 Tasa ideal de ocurrencias de caídas

En la práctica, la depuración no permite reparar las fallas del software (figura 16) y el cambio en el código introduce nuevas fallas en el sistema. La probabilidad de ocurrencia de estas fallas es más grande que la probabilidad de ocurrencias de las fallas reparadas. Por lo tanto la fiabilidad empeora en lugar de mejorar

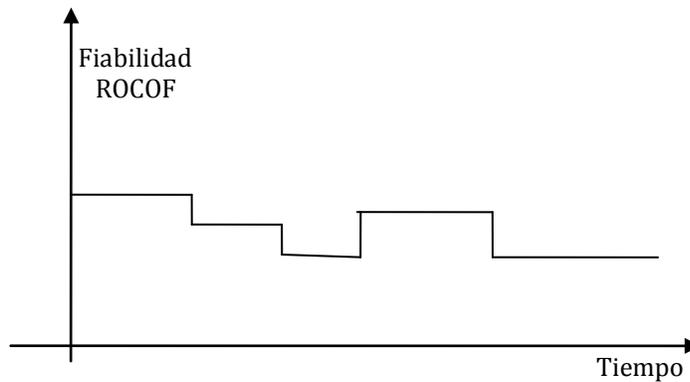


Figura 6.4.3 Tasa real de ocurrencias de caídas

Cuando una reparación en el software introduce errores adicionales la fiabilidad negativa crece, por lo tanto reparar dichas fallas contribuye al crecimiento de la fiabilidad.

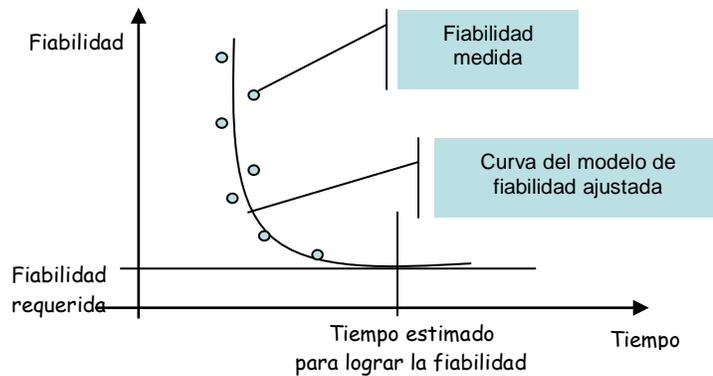


Figura 6.4.4 Predicción de la fiabilidad

6. Verificación, validación y pruebas del software

Predecir la fiabilidad a partir de un modelo tiene dos ventajas

- Planeación de las pruebas
- Negociación con los clientes. A veces el modelo muestra un crecimiento lento de la fiabilidad y requiere de una cantidad de esfuerzo desproporcionada para obtener pocos beneficios o incluso que nunca se alcanzará la fiabilidad pactada, por lo que la renegociación de los requerimientos es fundamental.

A modo de resumen se presenta el siguiente cuadro de las medidas de fiabilidad:

Cuadro 6.4.1 Pruebas estadísticas

Nombre	Descripción	Explicación	Ejemplo
POFOD	Probabilidad de caída en la demanda	La probabilidad de que un sistema falle cuando se haga una petición del servicio	Un POFOD de 0.001 significa que una en mil peticiones de servicio conduce a una caída
ROCOF	Tasa de ocurrencias de las caídas	La frecuencia de ocurrencia con la cual un comportamiento inesperado es probable que ocurra. Esta métrica se conoce algunas veces como la intensidad de la caída	Un ROCOF de 2/100 significa que dos caídas son probables que ocurran en cada 100 unidades de tiempo de operación
MTTF	Tiempo medio entre caídas	El promedio de tiempo entre caídas observadas en el sistema	Un MTTF de 500 significa que una caída se puede esperar cada 500 unidades de tiempo
AVAIL	Disponibilidad	La probabilidad de que un sistema esté disponible para su utilización en un momento dado	La disponibilidad de 0.998 significa que de cada 1000 unidades de tiempo es probable que el sistema esté disponible 998 de éstas.

Sin embargo el costo de generar perfiles operacionales son muy altos (no solo económico, sino en tiempo y esfuerzo) pero la contribución a la empresa se refleja en la inversión devuelta que a veces ha llegado a ser de 10 a 1. Por lo tanto, es importante detener lo antes posible una prueba y no “sobre probar” el sistema, dicho momento será cuando se haya alcanzado el nivel de confianza requerido.

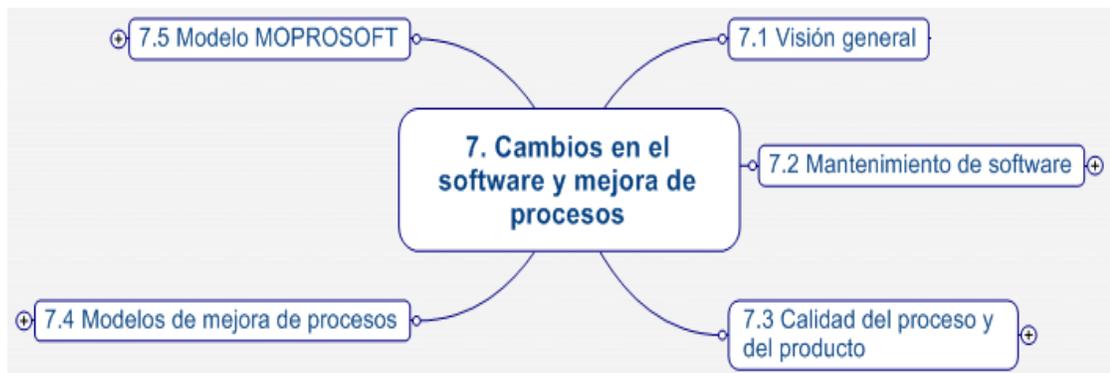
Unidad 7. Cambios en el software y mejora de procesos

Tema 7.1 Visión general

Objetivo Particular

Al término de la unidad, el alumno identificará la importancia de los estándares de calidad para crear proyectos y promover la mejora continua en los ya existentes.

Mapa conceptual



Tema 7.2 Mantenimiento de software

Objetivo Particular

Al término del tema, el alumno reconocerá la importancia de mejorar en forma sistemática los procesos de un software, tomando en cuenta la evolución continua de la administración de la información.

Mapa conceptual



El cambio en el software es muy importante puesto que las organizaciones dependen completamente de sus sistemas de software y han invertido millones de dólares en estos sistemas.

Existen varias estrategias diferentes para cambiar el software:¹²

- Mantenimiento de software. Los cambios en el software se hacen en respuesta a los cambios de requerimientos pero la estructura fundamental de dicho software permanece estable. Este es el enfoque más común que se utiliza para cambiar los sistemas.
- Transformación arquitectónica. Este es el enfoque más radical para cambiar el software y continuar dándoles mantenimiento conforme se implementan cambios más importantes en la arquitectura de datos centralizada a un cliente- servidor.
- Reingeniería de software. Esta es diferente de las otras estrategias en el sentido de que no se agrega nueva funcionalidad al sistema. Más bien, el sistema se modifica para hacerlo más fácil de comprender y cambiar.

Sin embargo para algunos sistemas, ninguna de éstas estrategias de cambios es apropiada y es necesario reemplazar el sistema.

Los cambios en el software inevitablemente generan muchas versiones diferentes de un sistema de software y sus componentes. Es muy importante registrar estas versiones diferentes y asegurar que las versiones apropiadas de los componentes se utilizan en cada versión del sistema.

¹² Se recomienda consultar el caso práctico, en donde se establecen algunas estrategias de mantenimiento al software, p 57

El proceso

Se inicia por un conjunto de peticiones de cambios provenientes de los usuarios del sistema, los administradores o los clientes. El costo e impacto de éstos cambios se evalúan para ver cuanto del sistema se afecta por los cambios y cuanto podría costar incrementarlos. Si los cambios propuestos se aceptan, se planea una nueva versión del sistema.

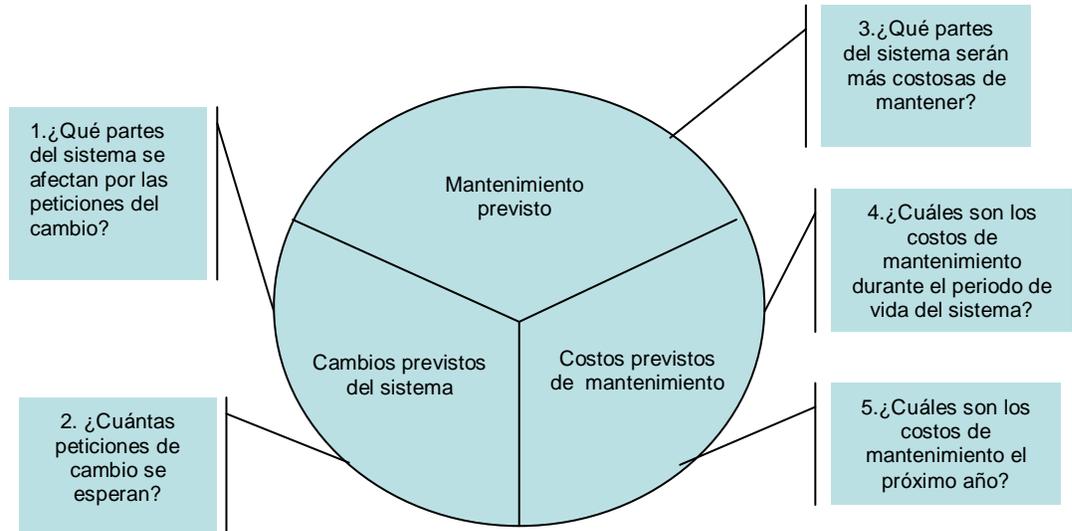


Figura 7.2.1 Relación mantenimiento y costos previstos

Como apoyo a la pregunta 2 se requiere entender la relación del sistema y su entorno para evaluar:

- El número y la complejidad de las interfaces del sistema, entre más grande sea el número de interfaces y más compleja sea ésta, es más probable que se hagan más peticiones de cambio.
- Número de requerimientos inherentemente volátiles del sistema, se refiere a las políticas y procedimientos organizacionales
- Procesos de negocio en los que se utiliza el sistema. Dado que los procesos de negocio evolucionan, provocan peticiones de cambio.

El modelo COCOMO 2 de estimación costos definido por Boehm (1995), señala que una estimación para el esfuerzo de mantenimiento del software se basa en el esfuerzo de comprender el código existente y en el esfuerzo de desarrollo del nuevo código.

Dinámica de la evolución de los programas

Lehman y Belady (1985), realizaron un estudio en donde proponen un conjunto de leyes que se refieren al cambio en los sistemas. Así mismo señalan que éstas leyes son invariantes y de aplicación amplia.

- Cambio continuo. Un programa utilizado en un entorno real necesariamente debe cambiar o llegar a ser progresivamente menos útil en ese entorno. Puesto que el entorno del sistema cambia, surgen nuevos requerimientos y el sistema debe modificarse, así cuando el sistema modificado se introduce en el entorno, éste promueve más cambios ambientales por lo que el proceso de evolución se recicla.

2. Incremento en la complejidad. Puesto que un programa evolutivo cambia, su estructura tiende a ser más compleja. Dado que un sistema cambia, su estructura se degrada, ésto se observa en sistemas heredados. La única forma de evitar que esto pase, es invertir en mantenimiento preventivo en donde se mejora la estructura sin agregar funcionalidad.
3. Evolución prolongada del programa. Es un proceso autoregulatorio donde los atributos del sistema: tamaño, tiempo de entrega, número de errores reportados; son aproximadamente invariantes para cada entrega del sistema. Esta ley sugiere que los sistemas grandes tengan una dinámica por sí mismos establecida en un etapa inicial del proceso de desarrollo.
4. Estabilidad organizacional. Durante el tiempo de vida de un programa, su tasa de desarrollo es aproximadamente constante e independiente de los recursos dedicados al desarrollo del sistema.
5. Conservación de la familiaridad. Durante el tiempo de vida de un sistema, el cambio incremental en cada entrega es aproximadamente constante

Distribución del esfuerzo total disponible para mantenimiento

El siguiente enfoque fue avalado por Lientz y Swanson en 1980 y posteriormente por Nosek y Palvia en 1990.

- Mantenimiento para preparar las fallas del software 17%
- Mantenimiento para adaptar el software a diferentes entornos operativos 18%
- Mantenimiento para agregar o modificar su funcionalidad del sistema 65%

Los costos de mantenimiento del sistema representan una gran proporción del presupuesto de muchas organizaciones. Mckee (1984) señala que la cantidad de esfuerzo invertida está entre 65% y 75% del esfuerzo total disponible. Guimares (1983), demostró que los costos de mantenimiento eran comparables con los costos de desarrollo del sistema. Por lo general, es costoso invertir esfuerzo para reducir costos de mantenimiento, dado que es más costoso agregar funcionalidad después de la entrega.

Las buenas técnicas de ingeniería de software, como la especificación precisa, la utilización del desarrollo orientado a objetos y la administración de configuración, contribuyen a una reducción en los costos de mantenimiento

Factores que influyen en el desarrollo y mantenimiento y conducen a costos de mantenimiento más altos son:

- Estabilidad en el equipo. Aunque se entiende la necesidad de asignar nuevos proyectos al equipo de desarrollo, es importante seguir considerando al equipo inicial en la fase de mantenimiento.
- Responsabilidad contractual. Las empresas suelen contratar el desarrollo y mantenimiento por separado y esto trae como consecuencia que el equipo de desarrollo no tomé en cuenta la facilidad del diseño involucrando al mantenimiento
- Habilidades del personal. Debido a que el proceso de mantenimiento tiene una imagen pobre entre los ingenieros de software por lo general el equipo dedicado a esta tarea son principiantes y gente inexperta en el área
- Edad y estructura del programa. Conforme para el tiempo, lo programas tienden a degradarse por los cambios y esto hace más difícil comprenderlos, más aún aquéllos sistemas heredados en donde las técnicas de software no eran tan modernas muy probablemente los programas nunca estuvieron bien estructurados, y la documentación está extraviada o es inconsistente.

7. Cambios en el software y mejora de procesos

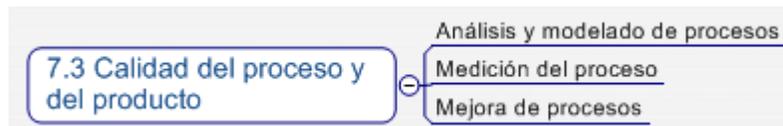
Más que gastar esfuerzo y costo en mantenimiento y desarrollo de sistemas hasta que sea imposible seguir con el mantenimiento y sugerir el reemplazo es preferible considerar SISTEMAS EVOLUTIVOS, los cuales van cambiando en respuesta a las nuevas demandas, lo cual se logra a través de la Reingeniería, en donde se procura mejorar la estructura y comprensión del sistema.

Tema 7.3 Calidad del proceso y del producto

Objetivo Particular

Al término del tema, el alumno reconocerá la importancia de medir y mejorar los procesos de elaboración de software para obtener productos de calidad.

Mapa conceptual



Análisis y modelado de procesos

W.E. Deming trabajó con la industria japonesa después de la segunda guerra mundial para mejorar la calidad introduciendo la idea de control estadístico de la calidad basándose en medir el número de defectos y relacionarlos con el proceso, de tal forma que el proceso se mejora cuando se logre que sea repetible, es decir, hasta que los resultados del proceso sean predecibles y el número de defectos se reduzca

Para productos de software, existen cuatro factores que afectan la calidad del producto:

- Tecnología de desarrollo
- Calidad del proceso
- Calidad del personal
- Costos, tiempo, y duración

Así la influencia de estos factores dependen del tamaño y tipo de empresa. Por ejemplo, si se trata de una empresa grande compuesto por subsistemas independientes, el factor determinante es la calidad del proceso. Sin embargo, la causa real de los problemas de calidad del software es que las organizaciones deben competir para sobrevivir.

Como se indicó, el análisis y modelado de procesos comprende estudiar los procesos existentes y desarrollar un modelo abstracto de estos procesos que capture sus características principales, los cuales son toda la gama de diagramas vistos anteriormente.

Sin embargo, también es indispensable realizar estudios etnográficos para descubrir los procesos que realmente se utilizan. Este tipo de análisis se basa en la observación externa del proceso y cubre desde las primeras etapas del proyecto hasta la entrega y mantenimiento del producto.

Los modelos de procesos incluyen actividades, subprocesos, roles, excepciones, comunicaciones de los productos a entregar y otros procesos. Así mismo las excepciones se tienen que manejar dinámicamente de tal forma que cualquier problema que se tenga que

resolver por el administrador de proyecto afecte lo menos posible la calendarización de actividades o presupuestos definidos o recursos del proyecto.

Es importante considerar las excepciones más comunes con el fin de establecer un estándar de acuerdo al proyecto.

Medición del proceso

Las métricas del proceso se utilizan para evaluar si la eficiencia de un proceso ha mejorado. Se puede recolectar tres clases de métricas del proceso:

1. Tiempo requerido para completar un proceso en particular
2. Recursos requeridos para un proceso en particular
3. El número de ocurrencias de un evento en particular

Los dos primero ayudan a descubrir si los cambios en el proceso mejoran la eficiencia de un proceso. El tercero tiene una influencia directa en la calidad del software.

La dificultad del proceso de medición es conocer ¿qué medir?, Basili y Rombach proponen el paradigma GQM (Goal Question Metric) el cual ayuda a decidir que mediciones tomar y cómo utilizarlas. Este enfoque se basa en la identificación de:

- Metas. Lo que la organización esta tratando de lograr. Mejorar la productividad de los programadores, reducir tiempos de desarrollo del producto, incrementar fiabilidad del producto.
- Preguntas. Son refinamientos de las metas donde se identifican áreas específicas de incertidumbre relacionadas con las metas. ¿cómo se puede reducir el tiempo requerido para finalizar los requerimientos del producto? ¿cómo se pueden llevar a cabo de forma más efectiva las evaluaciones de la fiabilidad?
- Métricas. Son mediciones que se requieren recolectar para ayudar a responder las preguntas y confirmar si las mejoras del proceso ayudaron a cumplir la meta deseada.

Mejora de procesos

El objetivo de la mejora de procesos es explicar como se pueden mejorar los procesos del software para producir mejor software:

- Calidad del proceso
- Análisis y modelado de procesos
- Medición del proceso
- El modelo de madurez de la capacidad del proceso del SEI (Software Engineering Institute)
- Clasificación del proceso

La mejora de procesos significa comprender los procesos existentes y cambiarlos para mejorar la calidad del producto y o reducir costos y tiempos de desarrollo

7. Cambios en el software y mejora de procesos

Los procesos del software son inherentemente complejos y comprenden un gran número de actividades, por tanto así los procesos como los productos también tienen atributos o características como:

- Comprensión: ¿Hasta que punto se definen completamente el proceso y que tan fácil es comprender su definición?
- Visibilidad: ¿Las actividades del proceso culminan en resultados claros de tal forma que el progreso del proceso sea visible externamente?
- Apoyo: ¿Hasta que punto las actividades del proceso pueden apoyarse en herramientas CASE?
- Fiabilidad: ¿El proceso diseñado es de tal forma que los errores del proceso se evitan o identifican antes de que se conviertan en errores del proceso?
- Aceptación: ¿El proceso definido es aceptable y utilizable por los ingenieros responsables de producir el producto de software?
- Robustez: ¿El proceso puede continuar a pesar de los problemas inesperados?
- Mantenibilidad: ¿El proceso puede evolucionar para reflejar los requerimientos organizacionales cambiantes o las mejoras identificadas del proceso?
- Rapidez: ¿Qué tan rápido se puede completar el proceso de construcción de un sistema a partir de una especificación dada?

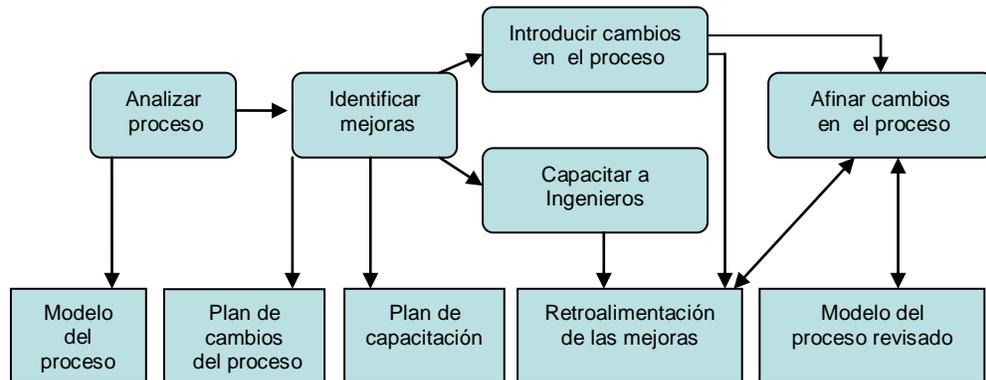


Figura 7.3.1 Mejora de procesos

Tema 7.4 Modelos de mejora de procesos

Objetivo Particular

Al término del tema, el alumno identificará dos tipos de modelos para la mejora de procesos, su utilidad y las diferentes formas de llevarla a cabo.

Mapa conceptual



Modelo de madurez de la capacidad del proceso del SEI (CMM)

El SEI (Software Engineering Institute) en EUA, lleva a cabo un programa de largo plazo de mejora del proceso del software, parte de este programa es el Modelo de Madurez de la Capacidad (CMM), el cual se refiere a las mejores prácticas en Ingeniería de Software.

Este modelo fue desarrollado por el Instituto de Ingeniería de Software de la Universidad de Carnegie – Mellon, fundado por el departamento de defensa de los Estados Unidos de América. Este modelo tuvo mejoras adaptadas a proyectos enfocados a la defensa, por lo que no debe tomarse como un modelo de capacidad definitiva para todos los procesos del software dado que el desarrollo de software para la defensas son sistemas de larga vida y de largo plazo que tienen interfaces complejas con el hardware y con otros sistemas de software. Son desarrollados por grandes equipos de ingenieros y deben seguir estándares y procedimientos de desarrollo señalados por el Departamento de Defensa de los Estados Unidos de América

Los principales problemas con el modelo de madurez de la capacidad son:

1. El modelo se enfoca exclusivamente en la administración del proyecto más que en el desarrollo del producto, no toma en cuenta la tecnología utilizada por la organización, como construcción de prototipos, métodos formales, etc.
2. Excluye el análisis y resolución de riesgos como una tecnología de procesos clave. Dicho análisis permite descubrir problemas antes de que afecten seriamente el proceso de desarrollo
3. El dominio de aplicación del modelo no está definido, los autores del modelo Humphrey y Curtis 1991, aceptan que el modelo no es apropiado para todas las organizaciones pero no describen tampoco el tipo de organización donde el modelo puede ser utilizado.

Sin embargo Paulk en 1995, compara este modelo con el estándar ISO 9000 y concluye que CMM es más detallado y preescribe e incluye un marco de trabajo para la mejora de procesos, lo cual no cubre ISO 9000.

Así, aquellas empresas que se encuentran en el nivel 1 o 2 son compatibles con ISO 9000. Sin embargo se han elaborado más modelos CMM enfocados a los siguientes puntos:

- Ingeniería de sistemas
- Adquisición de sistemas
- Administración de personal
- Clasificación del proceso

El CMM del SEI es apropiada para proyectos de software de larga vida y largo plazo realizado por organizaciones grandes, así en lugar de clasificar los procesos en niveles, se cree que un enfoque más general para la clasificación de los procesos se puede aplicar a lo largo de un espectro de organizaciones y proyectos.

El modelo SEI es apropiado para sistemas grandes desarrollados por equipos grandes de ingenieros, no se aplica sin la adaptación de circunstancias locales.

El modelo está formado por cinco niveles que describen las metas prácticas y capacidades específicas con que debe contar un proceso maduro de software. Estas clasificaciones obviamente se solapan y un proceso puede caer en varias clases, sin embargo esta clasificación sirve de base para la mejora de procesos multidimensional. En cada nivel, además de establecer una escala de medida de la capacidad de los procesos, se fijan unos objetivos que ayudan a la organización a priorizar los esfuerzos dedicados a la mejora de estos procesos.

En el Nivel 1 o Inicial, no existen áreas de proceso, los procesos no están definidos aunque algunos de ellos se realicen de forma controlada. El trabajo se realiza de manera impredecible. El éxito depende de las capacidades y esfuerzos individuales realizados por las personas.

En el Nivel 2 o Repetible, se encuentran establecidas las actividades básicas de gestión de proyectos, de forma que pueden observarse y reproducirse las prácticas que han tenido éxito en proyectos anteriores.

En el Nivel 3 o Definido, se documentan y estandarizan tanto los procesos de desarrollo y mantenimiento de software, como los de gestión. Todos los proyectos usan una versión de los procesos integrada en la organización.

En el Nivel 4 o Gestionado, se establece un programa amplio y detallado de medidas, tanto para el proceso como para el producto software, Se recogen y analizan los datos de todos los proyectos, formando una base de datos cuantitativa que será de gran ayuda en la evaluación y estimación de proyectos.

En el Nivel 5 o en Optimización, la organización está inmersa en un proceso de mejora continua de todos sus procesos, recogiendo datos de todos sus proyectos y utilizándolos para aportar nuevas ideas y para proporcionar innovaciones que utilizan nuevos métodos y tecnologías.

Estos niveles se ilustran en la figura xxx



Figura 7.4.1 Niveles del Modelo CMM

Por otra parte, cada vez que se desea lograr el siguiente nivel se deben seguir ciertos planes o estrategias, de tal forma que los procesos evolucionen como se ha planeado, en la figura xxx, se muestra la arquitectura o procedimientos que realiza el Modelo CMM en cada uno de sus niveles.

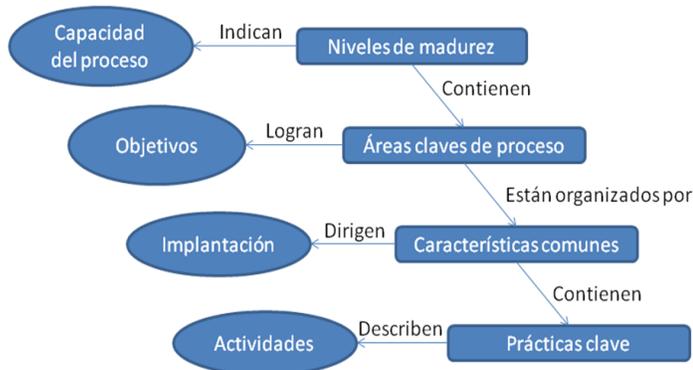


Figura 7.4.2 Arquitectura del modelo CMM

Modelo de madurez de la capacidad del personal P-CMM

El modelo CMM, se refiere a las mejores prácticas en Ingeniería de software, por otra parte el modelo P-CMM, se utiliza para mejorar la administración de sus activos humanos. El modelo CMM tiene definido cinco niveles:

P-CMM, se puede utilizar como un marco de trabajo para mejorar la forma en la que una organización administra sus activos humanos. Es un modelo de cinco niveles:

- Inicial. Adhoc, se utilizan prácticas informales de administración de personal.

- Repetible. Establece las políticas para el desarrollo de la capacidad del personal, compensación capacitación, administración del desempeño, comunicación, ambiente de trabajo.
- Definido. Estandarización de las mejores practicas de administración de personal a lo largo de la organización. Cultura participativa, prácticas basadas en la competencia, desarrollo profesional, planeación de la fuerza de trabajo, análisis del conocimiento y las habilidades.
- Administrado. Se establecen e introducen las metas cuantitativas para la administración de personal. Alineación del desempeño organizacional, administración de la competencia organizacional, prácticas en el equipo, formación de equipos , asesoramiento.
- Optimizado. Existe un enfoque continuo en la mejora de la competencia individual y la motivación de la fuerza de trabajo. Innovación continua de la fuerza de trabajo, asesoramiento, desarrollo de competencia personal.

El modelo P-CMM, es el modelo de madurez de gestión de personal, el cual consiste de 7 niveles:

- Reclutamiento
- Selección
- Gestión del desempeño
- Entrenamiento
- Retribución
- Desarrollo de carrera
- Diseño de la organización y del trabajo.

Los administradores de software tienen por función asegurar que la organización obtenga los mejores beneficios posibles al invertir en las personas, así también deben resolver problemas técnicos y no técnicos, motivar a la gente, planear y organizar su trabajo, y asegurar que el trabajo se realice de la manera adecuada. No realizar lo anterior o hacerlo pobremente lleva al fracaso de los proyectos.

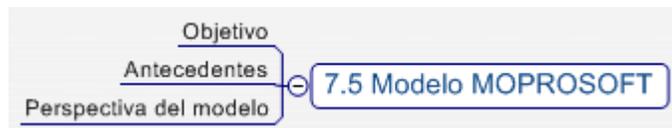
La discusión de la administración se centra en dos factores cognoscitiva y social, la ingeniería de software es precisamente una actividad cognoscitiva y social, por tanto si los administradores comprenden estos factores, podrán hacer mejor su trabajo.

Tema 7.5 Modelo de procesos para la industria de software (MoProSoft)

Objetivo Particular

Al término del tema, el alumno reconocerá la importancia de implementar normas de calidad para la elaboración de proyectos de software con el fin de alcanzar la etapa de maduración en éste ámbito.

Mapa conceptual



Objetivo del modelo

En la actualidad, es indudable que el software es la herramienta que establece las dinámicas laborales, de producción y hasta de convivencia en todo el mundo. Los múltiples desarrollos que en este ámbito se dan –casi cotidianamente – generan como consecuencia la necesidad de establecer cánones de calidad para cada producto, para así garantizar que su desempeño y sus funciones cubran las expectativas de sus consumidores y que, en la praxis, cumplan con su cometido satisfactoriamente. Consciente de ello, la Asociación Mexicana para la Calidad en Ingeniería de Software (AMCIS) ha trabajado en el desarrollo de un modelo que cubra los requisitos que la norma ISO 9000 demanda de los productos de esta naturaleza. Al mismo tiempo, la AMCIS innova en este campo, pues se espera que sea la contribución mexicana a la industria del software.

El Modelo de Procesos para la Industria de Software, Moprosoft, tiene por objetivo proporcionar a la industria mexicana, y a las áreas internas dedicadas al desarrollo y mantenimiento de software, un conjunto integrado de las mejores prácticas basadas en los modelos y estándares reconocidos internacionalmente, tales como ISO 9000:2000, CMM-SW, ISO/ IEC 15504, PMBOK, SWEBOK entre otros. Moprosoft contiene tres categorías de procesos que corresponden a las capas de Alta Dirección, Gestión y Operación. La categoría de Alta Dirección contiene el proceso de Gestión de Negocio; la categoría de Gestión se compone de Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos, a su vez, este último se divide en tres subprocesos: el de Recursos Humanos, el de Bienes, Servicios e Infraestructura y el de Conocimiento de la Organización. Finalmente, la categoría de Operación contiene los procesos de Administración de Proyectos Específicos y de Desarrollo y Mantenimiento de Software. El propósito de contar con un modelo de estas características es apoyar a la industria de software en su tránsito del estado actual, en el cual la calidad de los productos depende principalmente de las habilidades de los individuos, al estado deseado: en donde la calidad de los productos de software será la consecuencia de la madurez en los procesos de las organizaciones.

Antecedentes del modelo

Moprosoft fue desarrollado durante el 2002, como consecuencia de los acuerdos de la mesa de la Estrategia 6 del Programa para el Desarrollo de la Industria de Software dirigido por la Secretaría de Economía, bajo un convenio con la Facultad de Ciencias de la UNAM.

Para el desarrollo de este proyecto se convocó, a través de la Asociación Mexicana para la Calidad en Ingeniería de Software (AMCIS), a personas con experiencia y conocimiento en los modelos de procesos y calidad de software, quienes serían los que conformarían al grupo editor bajo la dirección de la Dra. Hanna Oktaba.

En atención a una solicitud de la Secretaría de Economía, Moprosoft fue desarrollado como el documento base para la definición de una norma mexicana dirigida a la industria de software. Actualmente, está por incluirse dentro del Programa de Normalización Nacional de la Dirección General de Normas una propuesta de norma basada en modelo Moprosoft.

Al estandarizar los procesos y las prácticas usando Moprosoft, la industria de software podría tener las siguientes ventajas, que ayudarían en su crecimiento:

- Al tener prácticas integradas, que abarcan desde la gestión de negocio hasta el desarrollo y mantenimiento de software, las empresas tendrían mayor control sobre su desempeño en el mercado.
- El costo de la incorporación del nuevo personal podría disminuir si se enfocan la educación y la capacitación a un modelo único.
- Las empresas pequeñas, al seguir procesos similares, podrían asociarse con mayor facilidad para afrontar proyectos de mayor envergadura.
- La exportación de servicios de software de las empresas mexicanas sería más factible. Incluso se podría disminuir la necesidad de la intermediación de las empresas transnacionales, gracias a que Moprosoft considera las prácticas reconocidas internacionalmente. Moprosoft contiene tres categorías de procesos que corresponden a las capas de Alta Dirección, Gestión y Operación

Perspectiva del modelo

La norma mexicana basada en Moprosoft será armonizada con la ISO/IEC 15504, cuya parte 2 se encuentra actualmente en publicación como norma internacional. Esto significa que la norma mexicana contendrá, además del modelo de procesos basado en Moprosoft, un modelo de escala de evaluación basado en ISO/IEC 15504 y un método de evaluación, que todavía está en proceso de definición, que cumplirá con los requisitos de ISO/IEC 15504. Al lograrlo, tendremos en México una norma equiparable con el modelo CMMI y el método de evaluación SCAMPI desarrollados en los Estados Unidos. Entonces para la implantación y el buen funcionamiento de la norma se requiere constituir un organismo rector que se ocupe de ofrecer la capacitación y la certificación de instructores, consultores y evaluadores, el control y registro de las evaluaciones, al igual que de la evaluación y la mejora de la norma. La implantación de Moprosoft no demandará la incorporación de personal especializado en las empresas, únicamente requerirá de una adecuada capacitación del personal existente.

Moprosoft, en su versión original como modelo de procesos, ya está disponible de manera gratuita a través del portal www.software.net.mx. Las empresas y las áreas internas interesadas ya pueden estudiarlo y compararlo con sus prácticas. Se trata de un documento con poca extensión, unas 121 páginas, y al compararlo con otros modelos y estándares, resulta bastante práctico. Si las empresas lo encuentran útil y comprensible, y empiezan a adoptarlo antes de que se convierta en la norma, la industria comenzará de inmediato su proceso de maduración para mejorar la calidad de los procesos y los productos, adelantando así el proceso de implantación, incluso antes de que éste adquiera un carácter de normatividad.

7. Cambios en el software y mejora de procesos

Moprosoft, a diferencia de CMM-SW y CMMI, está dirigido a la micro y pequeña industria. Sintetiza las mejores prácticas en un conjunto pequeño de procesos que abarcan las responsabilidades de la alta dirección, gestión y operación. Se trata de un modelo integrado, en el cual: las salidas de un proceso están claramente dirigidas como entradas a otros procesos; las prácticas de planeación, seguimiento y evaluación se incluyeron en todos los procesos de gestión y administración; los objetivos, los indicadores, las mediciones y las metas cuantitativas fueron incorporadas de manera congruente y práctica en todos los procesos; las verificaciones, validaciones y pruebas están incluidas de manera explícita dentro de las actividades de los procesos; y existe una base de conocimiento que resguarda todos los documentos y productos generados por los procesos.

En otras palabras, es un modelo que de manera más pragmática que otros presenta las mejores prácticas para la industria de software.

Conclusiones

El software es el factor que marca la diferencia, de tal forma que las grandes y pequeñas empresas se están convirtiendo en fábricas de software que envejecen con cientos de aplicaciones basadas en software que están en una situación crítica y necesitan ser renovados urgentemente.

Hoy por hoy y con base en mi experiencia profesional, no es suficiente reparar lo que está mal y dar una imagen moderna a éstas aplicaciones, sino más bien empezar a crear conciencia entre los estudiantes, desarrolladores y todas las personas dedicadas a la elaboración de software de cualquier tipo, sobre la responsabilidad profesional y ética que recae en ellos. Es decir, se espera que este documento ayude al lector y en particular a los estudiantes de licenciaturas en Matemáticas Aplicadas y Computación, Ingenieros en Computación, o carreras dedicadas al diseño y desarrollo de software, a tomar conciencia en la importancia de la documentación, misma que permitirá llevar de manera ordenada todo el proceso de desarrollo del sistema de información. Llegando así a los resultados que todos buscan: “la combinación de mínimo tiempo y mínimo esfuerzo”. Comprobando así que la productividad, que cualquier desarrollo de software busca, la encontrará cuando la combinación eficiencia y eficacia sean las asertivas.

Por otra parte este trabajo me permitió confirmar la relevancia que tienen las asignaturas de la licenciatura en Matemáticas Aplicadas y Computación del Plan 82, ya que al conjuntar materias del área de sistemas como Teoría de sistemas, Análisis y diseño de sistemas, Investigación de operaciones y Teoría de graficas entre otras con el pensamiento ordenado que me otorgaron las asignaturas del área matemática, me han llevado a visualizar necesidades comentadas en este trabajo, que actualmente no se les ha dado la importancia oportuna en la ardua tarea de desarrollo y creación de software.

Bibliografía

Referencias bibliográficas

- Cuevas, G., *Ingeniería del software: Práctica de la programación*, Serie Paradigma, México, 1993.
- Dolado Cosín, Javier, *Técnicas cuantitativas para la gestión en la ingeniería del software*, Net biblo, 2009.
- Fuentes, A., *El enfoque de sistemas en la solución de problemas la elaboración del modelo conceptual*, Facultad de Ingeniería UNAM, México, 1995.
- Hamlet y Maybee, *The engineering of software*, Addison Wesley, E.U.A., 2001.
- Kendall y Kendall, *Análisis y diseño de sistemas*, Prentice Hall, México, 1991.
- McConnell, S., *Desarrollo y gestión de proyectos informáticos*, McGraw Hill, México, 1996.
- Pressman R., *Ingeniería de software, un enfoque práctico*, McGraw Hill, México, 2001.
- Senn, James, *Análisis y diseño de sistemas de información. McGraw Hill, México, 1989.*
- Sommerville, Ian, *Ingeniería de software*, Pearson Education, México, 2005.
- Fuentes, A., *El enfoque de sistemas en la solución de problemas la elaboración del modelo conceptual*, Facultad de Ingeniería UNAM, México, 1995
- McConnell, S., *Desarrollo y gestión de proyectos informáticos*, McGraw Hill, México, 1996
- Suárez, R., *Un modelo cualitativo del proceso de solución de problemas. El modelo del diamante*, Facultad de Ingeniería UNAM, México, 1995

Referencias electrónicas

- “Entrevista con la Dra. Hanna Oktaba, presidenta de la Asociación Mexicana para la Calidad en Ingeniería de Software (AMCIS)”, www.iie.org.mx/boletin032003/ind.pdf citado el 13 de febrero, 2009
- Merle Vicent, Director del CEREQ, Revista Europea, “La evolución de los sistemas de validación y certificación”, Disponible en la World Wide Web http://dialnet.unirioja.es/servlet/dfichero_articulo?codigo=131217&orden=83432, [citado el 10 de febrero, 2009]
- Betarte Gustavo, Luna Carlos, Sierra Luis, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, “Especificación y Verificación Formal de Sistemas Críticos en el Instituto de Computación de la Universidad de la República (Uruguay)”. Disponible en la World Wide Web <http://www.fing.edu.uy/inco/pedeciba/biblioteca/reptec/TR0714.pdf>, [citado el 14 de marzo, 2009]
- Pino, Francisco J. García Félix, Facultad de Ingeniería, Electrónica y Telecomunicaciones, “Revisión sistemática de mejora continua de procesos de software, micro pequeñas y medianas empresas”, Revista Española de Innovación de calidad e Ingeniería de Software Vol.2 No. 1, 2006. Disponible en la World Wide Web <http://redalyc.uaemex.mx/redalyc/pdf/922/92220103.pdf>, [citado el 4 de octubre, 2009]
- Pikatza, Juan M., Larburu, Iker U., Sobrado Francisco J., García Juan J., “Modelo jerárquico de roles para organizaciones de pequeño tamaño”, Departamento de Lenguajes y Sistemas Informáticos, UPV/EHU. Disponible en la World Wide Web <http://ceur-ws.org/Vol-88/paper4.pdf>, [citado el 21 de julio, 2009]
- Ruiz Carreira, Mercedes, Dpto. de Lenguajes y Sistemas Informáticos, Universidad de Cádiz; Ramos Román, Isabel, Dpto. de Lenguajes y Sistemas Informáticos
- Universidad de Sevilla, Facultad de Informática y Estadística. “Aprendizaje de CMM a través de la simulación” Disponible en la World Wide Web <http://bioinfo.uib.es/~joemiro/aenui/procJenui/ProcWeb/actas2001/ruapr487.pdf>, [citado el 23 de julio, 2009]

Estimación de diferencias empresariales

Metodologías y
herramientas de la
Ingeniería de Software. Un
caso práctico

Fernando Hernández Martínez

Índice

Contextualización	3
Requerimientos. Identificación del problema.....	7
Requerimientos. Definición del problema	13
Diseño y modelado del sistema propuesto.....	35
Diagramas de flujo de datos	35
Modelo de Contexto	40
Diagrama funcional del sistema	43
Propuesta de diseños de entrada, salidas e interfaz	44
Diagramas y documentación técnica.....	48
Implementación y validación	57

Contextualización

Introducción

Jiménez Márquez y Asociados tiene 15 años participando en el desarrollo de auditorías y dictámenes de todo lo relacionado con el Instituto Mexicano de Seguro Social IMSS y el Instituto de Fondo Nacional para la Vivienda INFONAVIT, esto es, formula dictámenes que establecen lo que se necesitan pagar las empresas a estas instituciones, con la finalidad de mantenerse al corriente con la ley Mexicana.

Para el desarrollo de una auditoria se siguen los procedimientos y normas establecidas en los principios básicos de contabilidad. El IMSS y el INFONAVIT han desarrollado procedimientos básicos para el desarrollo de las auditorias y la elaboración de los dictámenes que establece la ley para efectos de los mismos. Los procedimientos están establecidos en sus aspectos generales en la ley del IMSS y en su reglamento, por lo que el despacho debe acatar con las normas y procedimientos que establecen estos mismos.

El campo de trabajo del despacho son todas las empresas que cuentan con trabajadores y tienen la obligación de registrarlos ante el instituto. Particularmente la ley establece como obligatorio que todas las empresas que cuenten con más de 300 trabajadores se dictaminen.

Jiménez Márquez y Asociados es una empresa con 5 trabajadores y con un amplio campo de trabajo, tal que dado el caso de que se pudieran facilitar o apresurar los procesos, podría crecer de una manera importante.

Visión de la empresa

Coadyuvar en el mejoramiento y a la buena aplicación de la seguridad social en México orientando la aplicación de la ley del Seguro Social en beneficio de los trabajadores y de las empresas mexicanas para lograr un crecimiento constante involucrando a más personal para poder abarcar a más empresas medianas.

Misión de la empresa

Participar como intermediario entre las instituciones que participan en la seguridad social en México (IMSS e INFONAVIT) y las empresas medianas para cumplir plenamente con las obligaciones que impone la ley mediante la formulación de auditorías y dictámenes.

Objetivo y políticas de la empresa

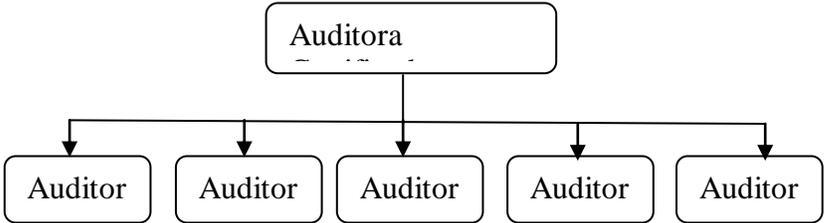
Formular dictámenes y auditorías que permitan a las empresas cumplir con las obligaciones establecidas por la ley en materia de seguridad social. El buen cumplimiento de la ley podrá beneficiar a los trabajadores, ya que recibirán sus prestaciones correspondientes, así como a la empresa, ya que esta no tendrá que pagar penalizaciones por el mal cumplimiento de los pagos a las instituciones. Esto se sustenta en la aplicación estricta de la LEY DEL SEGURO SOCIAL y de la LEY DEL INSTITUTO DEL FONDO NACIONAL DE LA VIVIENDA PARA LOS TRABAJADORES. Se aplican las normas de auditoría y procedimientos establecidos en sus reglamentos aplicables en materia de contabilidad y de seguridad social.

Estrategias o planes de desarrollo

La formulación de dictámenes y auditorías requiere de la elaboración y aplicación de modelos matemáticos que se realizan con el conocimiento de los conceptos fiscales referentes a los movimientos en los pagos que realizan los trabajadores y las empresas al IMSS e INFONAVIT. La buena formulación de auditorías permitirá que la empresa no pague innecesariamente penalizaciones.

Estructura organizacional de la empresa/ Estructura organizacional del departamento de interés

Despacho Jiménez Márquez Anexo de auditoría IMSS/INFONAVIT



Funciones de los puestos del departamento

Auditora Certificada

Elaborar los planes de trabajo, programas de actividades y estrategias a seguir de acuerdo a las peculiaridades de cada una de las empresas que se van a dictaminar. Suministrará de manera clara el trabajo que se debe realizar y tomará las decisiones importantes referentes al despacho.

Audidores

Suministrarán la información oportunamente para el cumplimiento del trabajo elaborando las visitas empresariales necesarias, revisando que la documentación y los procedimientos sustenten el cumplimiento de las normas en materia de seguridad social establecidas en el REGLAMENTO DE LA LEY DEL SEGURO SOCIAL EN MATERIA DE AFILIACIÓN, CLASIFICACIÓN DE EMPRESAS, RECAUDACIÓN Y FISCALIZACIÓN.

Actividades de los puestos del departamento

Auditora Certificada

Las actividades consisten en formular, analizar, revisar, ratificar y firmar los dictámenes para efectos del IMSS y del INFONAVIT.

Audidores

Elaborar las auditorias de las empresas y formular el dictamen correspondiente a esta, así como formular los anexos establecidos para la formulación del dictamen correspondiente.

Objetivo del sistema informático actual

El sistema informático actual se trabaja en hojas de cálculo e intenta evitar el traspaso de algunos datos de manera manual, aunque no se logre enteramente. Así mismo con el sistema actual es posible realizar unas funciones sencillas que son de ayuda para la realización de operaciones simples aplicadas a una cantidad considerable de datos, sin embargo los dictámenes que se realizan todavía requieren de mucho trabajo manual. El sistema actual contiene diversos datos del trabajador y nos permite establecer los resultados de cuotas pagadas por algunos conceptos exclusivos.

Estrictamente no se cuenta con un sistema informático para la formulación del dictamen. Se han desarrollado algunos programas aislados mediante hojas de cálculo que ayudan a la determinación de algunos procesos tendientes a la formulación del dictamen.

Requerimientos. Identificación del problema

Análisis de la situación actual.

En los últimos 10 años el IMSS ha hecho intentos para que las empresas formulen sus pagos en forma computarizada, pero hasta la fecha el crecimiento de las operaciones ha rebasado por mucho los intentos de sistematización del instituto. Esto se ve reflejado en el hecho de que aún a la fecha, para que una empresa realice su pago, tiene que hacer uso, en muchos casos, de disquetes de 3 ½ a pesar de que la mayoría de las computadoras no los aceptan. Otro elemento que nos indica el atraso del instituto es el hecho de que algunos de sus sistemas todavía requieren de impresoras de punto de las cuales a la fecha es muy difícil encontrar.

En los últimos 3 años han hecho intentos tanto el Instituto de Seguridad Social como e INFONAVIT de establecer una vía de comunicación con las empresas y los despachos a través del internet, pero estos sistemas no son eficientes y frecuentemente se ven bloqueados o no responden de acuerdo a las necesidades de las empresas.

El sistema informático actual se trabaja con hojas de cálculo utilizando información de diversas fuentes, dentro de las cuales se incluyen algunos datos de manera manual con la consecuente probabilidad de errores al transcribir la información, así mismo no se verifica o coteja la información de las fuentes lo cual frecuentemente genera errores textuales. Por lo anterior se hace necesaria la utilización de un sistema que permita la transferencia de información por medio de procedimientos informáticos.

Se están evaluando las variables y la identificación de la técnica más adecuada para mover los datos de una manera más eficiente, ya que hay que considerar que algunos procesos más viejos se siguen utilizando.

Será primordial la utilización y el manejo de una base de datos de acuerdo a las posibilidades del despacho. Así mismo será necesario contar con un mecanismo para comparar la información para poder verificar si la empresa está pagando lo debido. Esto, y las demás funciones que deberá de tener el software a utilizar deberán poder funcionar con los procedimientos anteriores, así como los actuales y futuros.

Identificación de las necesidades de información

Para poder llevar a cabo la realización de la nómina que define las cuotas obrero patronales se necesita información de dos fuentes. Primeramente es necesario que la empresa proporcione la información de sus trabajadores conforme ellos la conocen. Esta información contendrá las diversas cuotas, cargos, y abonos según y exactamente como la conoce la empresa. Por otra parte se requiere la información de la empresa, de los

trabajadores y sus cuotas respectivas por conceptos, según y exactamente como la conoce el IMSS. Con estas dos fuentes de (supuestamente) la misma información, se pueden verificar las diferencias que existen entre ellas para así determinar los datos y cuotas veraces que cumplan con la ley.

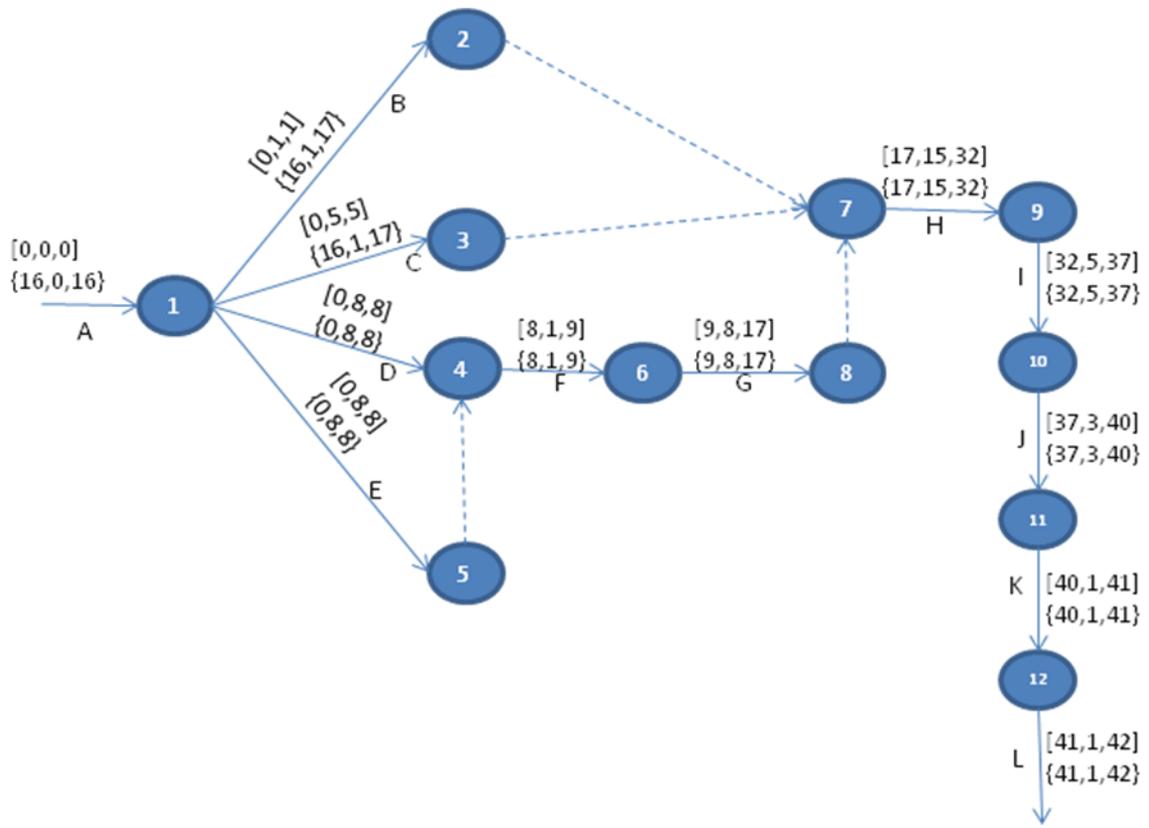
La información es una base de datos que contiene la información necesaria por trabajador como el CURP, los días trabajados, el salario diario, la cantidad pagada por seguridad social, la cantidad pagada para pagar el crédito de una vivienda, días inhábiles, primas vacacionales y dominicales, gratificaciones, entre otros datos que determinan el Salario Diario Integrado (SDI). El SDI conforma la parte más importante ya que los errores que se cometen comúnmente en las empresas se encuentran al formular este dato, y que ocasiona que no sea el mismo que afirma el Instituto.

Muchas veces no se hace el traspaso de los datos de forma electrónica, si no manual. Esto lleva a errores textuales que también llega a ocasionar problemas graves, ya que cualquier cambio en un nombre es considerado como una nueva persona. Por esta razón es indispensable la transferencia de datos electrónica y el cálculo del SDI con un proceso computacional. Esto podrá reducir de manera importante los errores y se podrá realizar el trabajo de una manera más eficiente.

Diagrama de PERT/CPM

El siguiente diagrama tiene como finalidad identificar los tiempos máximos y mínimos que se utilizan en la realización de las actividades señaladas anteriormente. También se identifican los tiempos de sobra o tiempos de holgura.

Clave de actividad	Descripción de actividad	Predecesor	Tiempo en días
A	Conseguir cliente	-	0
B	Identificación de conceptos	A	1
C	Consulta de la ley y del reglamento	A	5
D	Conseguir información de la empresa	A	8
E	Conseguir información del IMSS con respecto a la empresa	A	8
F	Crear la hoja de cálculo	D,E	1
G	Capturar la información	F	8
H	Realización de la nómina	B,C,G	15
I	Actualización de datos	H	5
J	Efectuar el análisis de diferencias	I	3
K	Cobrar las diferencias	J	1
L	Entregar costos	K	1



Formato de entrevistas, cuestionarios, documentos relacionados con el sistema de información.

Instrucciones: Responder el siguiente cuestionario

1. ¿Cuál es la función principal de despacho Jiménez Márquez?

2. ¿Quiénes son los clientes para los que trabaja el despacho?

3. A grandes rasgos, ¿cómo se lleva a cabo el proceso que realiza el despacho?

4. Por el momento, ¿Cuánto tiempo se tarda el llevar a cabo el trabajo para una empresa del tamaño habitual?

5. ¿Qué herramientas de trabajo se utilizan para llevar a cabo el proceso?

6. ¿Cuál es el factor común entre todos los procesos que llevan a cabo?

7. Indicar cuál es el problema que usted nota en la realización de los procesos del despacho.

8. Indicar que tipo de mejoras se podrían realizar en la realización de los procesos del despacho.

Análisis sobre la información recopilada.

Jiménez Márquez desarrolla auditorías y dictámenes para las empresas que cuentan con IMSS e INFONAVIT para cualquier empresa, normalmente de 300 trabajadores o más, los dictámenes son necesarios para poder determinar la cantidad a pagar que cumpla con la ley.

Dentro de los procedimientos establecidos para la formulación de dictámenes se cubren algunos aspectos fundamentales:

El primero está relacionado con el cumplimiento de las obligaciones que establece la ley y su reglamento, en todo lo relacionado con el cumplimiento de cuotas obrero-patronales.

Por otra parte el registro contable y los pagos específicamente correspondientes a la seguridad social.

Con el transcurso de los años el IMSS conforme ha aumentado el número de empresas en nuestro país, se ha visto en la necesidad de sistematizar sus procedimientos para controlar todo lo relacionado con el pago de cuotas por parte de las empresas, pero existen diversos problemas relacionados con el funcionamiento del sistema como sería la sobresaturación de su página electrónica al pasar información, o en otros casos, sus equipos y dispositivos de almacenaje no son los adecuados.

Por otra parte muchas empresas en México consideran que el cumplimiento de las obligaciones referentes al Seguro Social no son tan importantes como el cumplimiento de los compromisos fiscales. Por lo que frecuentemente el personal que desarrolla el dictamen inicial, y que mantiene al tanto las actualizaciones de los trabajadores, no tiene la preparación ni las herramientas necesarias para cumplir; no obstante los recursos que pagan las empresas mensualmente por conceptos de IMSS e INFONAVIT son mucho más grandes en el aspecto financiero que los fiscales.

En este contexto las empresas cuentan con diversos paquetes que se encargan del desarrollo de la contabilidad y de las nominas de las empresas, sin embargo no existen paquetes que consideren todo lo relacionado con el cumplimiento de las obligaciones en materia de seguridad social.

El despacho ha visualizado dentro de sus necesidades el desarrollo de un software que permita facilitar el desarrollo de una parte de la auditoria y que también sería útil para las empresas, puesto que les permitiría evitar pagos excesivos en sus cuotas, omisiones en el pago de cuotas, e importantes ahorros en el pago de multas y recargos que frecuentemente incurren por el incumplimiento de sus compromisos.

Para lograr lo anterior, se propone el desarrollo de un software que permita comparar la información que emite el IMSS relacionada con lo que espera que esta empresa pague contra la información que genera la propia empresa y ajustarla de tal manera que no incurra en el pago de diferencias, o cuotas por incumplimiento de sus cuotas.

Requerimientos. Definición del problema.

Introducción del análisis de factibilidad.

En el *capítulo 3. Ingeniería de requerimientos* que acompaña a éste Caso Practico, se hace referencia al Análisis de Factibilidad, en donde se mencionan los objetivos y alcances que se desean al final de este estudio. En la página 64, del capítulo 3 antes mencionado, se señala el objetivo de este tipo de análisis que a la letra dice:

“El **objetivo** de este estudio es realizar recomendaciones para el desarrollo del sistema. Se proponen cambios al alcance, el presupuesto y la calendarización del sistema. Se pueden sugerir requerimientos adicionales de alto nivel para los cambios. Las fuentes de información son los administradores, Ingenieros de software, Expertos en tecnología, Usuarios finales.”

En las siguientes secciones se desarrollan varios puntos importantes para dar un panorama general de los recursos que tiene, en este caso el Despacho contable Jiménez Márquez, de acuerdo al punto de vista objetivo de todas las personas involucradas en el desarrollo del software “Estimación de diferencias empresariales”, que nos han proporcionado.

1.a Declaración del problema

El despacho Jiménez Márquez y asociados ha trabajado durante muchos años transfiriendo datos del papel a la computadora de forma manual, tal que han ocasionado varios problemas básicos. Durante los últimos años el IMSS y el INFONAVIT han cambiado su forma de trabajo utilizando más medios electrónicos, ya que ha crecido la cantidad de datos que deben manejar. Este logro indujo a que Jiménez Márquez también haya tenido que cambiar su forma de trabajo para utilizar medios electrónicos. Sin embargo la utilización de los medios electrónicos todavía no se ha explotado lo suficiente.

Los problemas principales de dictaminar en Jiménez Márquez consisten en los errores de traslado de información, entre la impresión del dictamen realizado por la empresa a la computadora del asociado. También existen problemas al momento de las actualizaciones de las cuotas de los trabajadores, por ejemplo si renunciaron o si recibieron alguna bonificación. Las actualizaciones teóricamente las debería de realizar la empresa diariamente mediante una página electrónica, pero esto no se lleva a cabo, lo que ocasiona diferencias con la realidad de los datos. Así mismo existen algunas confusiones referentes a los diferentes medios de donde se obtiene la información.

La mayoría de estos problemas se podrían evitar con la ayuda de la computación, así como podrían diseñarse algunos modelos matemáticos que ayuden a formular dichos dictámenes. El hecho de que existan estos medios computacionales ayudar a que algunos procesos se faciliten, ayudando a disminuir los errores, y por lo tanto lograr una mayor eficiencia.

1.b Entorno de la implementación

El software será utilizado por los miembros de la empresa Jiménez Márquez para facilitar y hacer más eficiente el proceso. El software será diseñado como auxiliar de la dictaminación ya que trabajará con el traslado de información y con la formulación de cuotas necesarias para conjuntar la nómina y establecer cuáles son las diferencias en la información. El sistema cumplirá con la ley y la reglamentación (IMSS e INFONAVIT) con fines de un mejor enfoque que cumpla con la labor y con los objetivos que tiene Jiménez Márquez.

El sistema será capaz de integrarse y trabajar en conjunto con los sistemas que actualmente se están utilizando en la empresa, esto es, podrá trabajar con la información que se tiene en Excel, con el Sistema Único de Autodeterminación (SUA), y con el sistema IMSS Desde Su Empresa (IDSE).

Se dará una capacitación inicial a un trabajador de la empresa con la finalidad de poder conocer las funciones del software establecido, así como su uso. El software podrá ser utilizado por todos los empleados que requiera Jiménez Márquez, y la información se mantendrá confidencial conforme a lo indicado por la misma.

1.c Restricciones

El software está pensado para adaptarse fácilmente a los sistemas que se están usando actualmente por lo que las computadoras del momento cuentan con lo necesario para el funcionamiento del sistema, esto es Windows XP o superior, así como 256 MB de memoria RAM, 20 MB de disco duro y un procesador Intel o AMD que trabaje a 850 Mhz.

El software cumplirá con lo establecido por la ley, para facilitar la formulación de dictámenes empresariales. Existirá la posibilidad de que el sistema sea actualizado a causa de cambios en los conceptos o en la ley en el futuro, sin embargo, este proyecto en específico no tomará parte en las actualizaciones posteriores a la finalización del mismo. El tiempo para completar el funcionamiento del sistema en donde son estimadas las diferencias está estimado para finales de Noviembre del 2009.

Antes de la implementación del software, los procesos que se llevan a cabo en hoja de cálculo son poco eficientes, ya que estos se complementan con mucha verificación y refuerzo manual. Igualmente las hojas de cálculo formulan un costo alto por la obtención de las licencias y no se explota el paquete a su capacidad máxima, por lo que existe un gasto de recursos computacionales. El sistema que se va a implementar se acoplará a los recursos técnicos disponibles de la empresa.

2 Resumen de gestión y recomendaciones

2.a Objetivos

El software estará enfocado a la realización de un reporte de diferencias para estimar los recargos y multas que deben pagarle las empresas al Seguro Social e INFONAVIT procesando la información de una manera rápida y haciendo uso de los recursos disponibles para lograr una mayor eficiencia del proceso. Se hará uso de los recursos disponibles para minimizar los gastos, así como hacer que el proceso del cálculo de las diferencias se realice con más velocidad. El sistema será atractivo y sencillo de entender.

2.b Impacto

El impacto del software será importante, ya que la eficiencia del mismo le ahorrará mucho tiempo a la empresa que eventualmente se podrá enfocar hacia el crecimiento de la misma. Con mayor tiempo disponible, se podrían conseguir nuevos clientes empresariales y hasta estos mismos notaran una mejora en los resultados, así como será más cómoda la entrega de información para la empresa, y con menos complicaciones. Por lo que existirá una mayor satisfacción por los clientes de Jiménez Márquez.

La disminución de errores en el traslado de información, y la velocidad del proceso serán de primordial importancia para lograr esta eficiencia, que junto con la eficacia que tiene actualmente la empresa podría dar mejores resultados.

La implementación del software es el próximo paso que Jiménez Márquez tiene que realizar para estar al tanto de los recursos computacionales disponibles que se pueden explotar a favor del despacho. No solo esto, sino es necesario que exista un crecimiento en cuanto a los procesos computacionales, ya que hoy en día son indispensables. El IMSS y el INFONAVIT ya se percataron de esta necesidad y es necesario que el despacho no se quede atrás.

2.c Escenarios

A continuación se desarrollan tres diferentes escenarios para generar una propuesta de implementación al Despacho Contable Jiménez Márquez. El primer escenario definido "Lujo" se refiere a que el despacho contable, en un momento no muy lejano y con posibilidades económicas amplias, podrá invertir en el gasto para la adquisición de productos informáticos que ayudarán en la implementación y uso del software "Estimación de diferencias empresariales". El segundo y tercer escenario definidos "Medio" y "Básico" respectivamente, tienen el mismo objetivo que el primero, sólo que los costos recomendados no son tan altos como el primero, sino buscan un presupuesto en donde el cliente no gaste mucho y que en determinado momento se sigan ocupando los recursos informáticos que hasta el momento tiene el despacho.

En la siguiente tabla se describen los costos de los tres escenarios:

Tabla de Costos					
Lujo		Medio		Básico	
Concepto Tangible	Costos	Concepto Tangible	Intangible	Concepto Tangible	Intangible
Tarjeta de red (5)	\$845 c/u	Tarjeta de red (5)	\$400 c/u	Tarjeta de red (5)	\$160 c/u
Windows 7 Ultimate	\$4,320	Windows Vista Business	\$2,700	Windows XP Professional	\$1350.34
Quemador(5)	\$1,349	Quemador(5)	\$480	Quemador(5)	\$270
Laptop Toshiba (2) 61062	\$12,443	Laptop Toshiba (2)	\$8,399	Laptop Toshiba (1)	\$8,399
Multifuncional Brother (2) MFC290C	\$1,799	Multifuncional Brother (1) TX410	\$1,599	Multifuncional Deskjet (1) F4480HP	\$999
No-Break (5) Koblenz 3810	\$1,249	Regulador (5)	\$280	Regulador (5)	\$224
USB 8GB (1 por empresa, 6)	\$439.90 c/u	USB 4GB (1 por empresa,6)	\$249.90 c/u	USB 2GB (1 por empresa,6)	\$129.00 c/u
Total:	\$52,658.44	Total:	\$28,391.9	Total:	\$14,792.34

Costos de instalación

Tabla de Costos					
Lujo		Medio		Básico	
Concepto Tangible	Costos	Concepto Tangible	Intangible	Concepto Tangible	Intangible
Paquete de internet Telmex	\$999 c/mes	Paquete de internet Telmex	\$599 c/mes	Paquete de internet Telmex	\$389 c/mes
Norton Antivirus(5)	\$853.07 p/año	Panda Antivirus T&P	\$404 p/año	AVG Antivirus	Gratuito
Antispyware	Gratuito	Antispyware	Gratuito	Antispyware	Gratuito
Total:	\$1,852.03	Total:	\$1003	Total:	\$389

Costos repetidos

"OfficeMax.com.mx - Catálogo de Artículos." *OfficeMax.com.mx - Página Principal*. Web. Citado el 8 de octubre de 2009. <<http://www.officemax.com.mx/>>.

2.d Decisión para la mejor alternativa

Para poder tomar una decisión adecuada en cuanto a qué modelo de costos recomendar, en primera instancia analizaremos los costos, los beneficios, y su correlación.

Costos inicial de inversión:

- Escenario "lujo" = \$54,510.47
- Escenario "medio" = \$29,394.90
- Escenario "básico" = \$15,181.34

Costos por capacitación y otros:

En esta actividad se tiene por objetivo provocar un cambio positivo en la actitud mental, los conocimientos y habilidades de las personas que la reciben.

Los beneficios de la capacitación para el despacho contable, de acuerdo a la percepción de los usuarios y personas tomadoras de decisiones son:

- Crear una mejor imagen de la empresa
- Mejorar la relación entre el jefe y sus subordinados
- Incrementar la productividad y la calidad en el trabajo.

Se pretende que el Despacho Contable Jiménez Márquez, reciba un curso de capacitación formal en sus instalaciones con el fin de que todos los usuarios desarrollen sus habilidades, combinando la teoría y la práctica del software “Estimación de diferencias empresariales”. Se tiene contemplado que el curso se imparta en un máximo de 15 horas, distribuidas de acuerdo al tiempo disponible del despacho contable.

En el *capítulo 3. Ingeniería de requerimientos*, se definen varios métodos para medir el costo y el beneficio en la implementación del software.

3. Descripción del Sistema

El sistema consistirá en el cálculo de diversos conceptos y costos obrero-patronales que deben cumplir con la Ley de Seguridad Social. Este software será capaz de agilizar el proceso, reduciendo los tiempos de respuesta y la transferencia de información definitiva a los anexos que son obligatorios entregar al IMSS y al INFONAVIT. El sistema de información permitirá reducir errores de transferencia de información, así como la reducir errores de cálculos erróneos en los procesos de pagos a realizar.

Al agilizar los procesos, con un menor índice de errores el Despacho Contable podrá enfocarse en aspectos que generen mayores ganancias, como la actualización de la cartera de clientes, así como de la captación de nuevas empresas. Estos beneficios se podrán realizar dado que se incrementará la calidad del trabajo y la satisfacción de los clientes. Los clientes son empresas que cuentan con más de 300 trabajadores y estos están afiliados al Seguro Social e INFONAVIT, quienes de acuerdo a la Ley de prestaciones, se realiza la disminución correspondiente a las entidades respectivas para cumplir con dicha obligación.

Los clientes de Jiménez Márquez son:

- ETINCEL S.A. DE C.V
- INGENIERIA ESTRUCTURAL DE VANGUARDIA S DE R.L. DE C.V.
- IMPULSORA DE MATERIALES TENAYUCA S.A. DE C.V.
- ESCUELA SIERRA NEVADA S.C.
- INDUSTRIAS 3FM S.A DE C.V
- INSTITUTO BILINGÜE LIBERTAD S.C
- QUALITY KNITS S.A. DE C.V.

Los usuarios del sistema serán:

- los trabajadores del Despacho Jiménez Márquez

Los trabajadores actualmente trabajan con hojas de cálculo (EXCEL). Las computadoras que se utilizan trabajan con Windows XP y con Vista. Cuentan con conexión inalámbrica de internet, con copiadoras, multifuncionales, quemadores de disco, fax, teléfono. Aunque se trabaje con hojas de cálculo, muchos de los datos se traspasan de forma manual desde las nóminas en papel. Para evitar errores de traspaso de información, el sistema será capaz de hacer una conjunción entre los datos dados por la empresa, o la nómina, y los datos dados por el IMSS, o la liquidación. Estos se conjuntaran en una nueva estructura con únicamente los datos necesarios para calcular la cuota a pagar. Estos incluyen:

- Sueldos y salarios
- Aguinaldo
- Vacaciones
- Prima vacacional
- Prima dominical
- Bonos
- Suplencias
- Ajuste de sueldo
- Otras percepciones

Tomando estos valores directamente de las nóminas y de las liquidaciones, se determinará la cuota obrero patronal que deberá ser pagada al IMSS o INFONAVIT. Esta podrá ser desplegada para entenderlo detalladamente, pero también consistirá en un estimado final de las multas y recargos que deben ser pagados. Se podrá obtener un recibo para que se puedan pagar estas diferencias obtenidas entre los valores calculados por la institución y por la empresa. A la empresa auditada se le asignará un ejecutivo del Despacho el cual realizará el dictamen directamente en las instalaciones de la misma, para lo cual la aplicación será instalada en las oficinas de la empresa auditada. Para su implementación se dispondrán de los recursos computacionales con los que cuente el cliente.

El sistema mejorará la manera en la que se trabaja el proceso de calcular diferencias entre los cálculos, y disminuirá errores de ingreso de información. Se espera que el sistema disminuya el tiempo en lo que se realiza un dictamen final, por lo que podrá tener un alcance importante referente al crecimiento de la empresa para obtener más clientes.

4. Análisis Costo-Beneficio

El análisis costo-beneficio también puede referir a Análisis Económico, cuyo objetivo sea realizar una valoración de la justificación económica para un proyecto de sistema basado en computadora, en donde se pueden identificar costos tangibles e intangibles que se deben tomar en cuenta.

Para efectos de este Caso Practico, solo se analizará el impacto esperado del software SEDE, con base en los procesos que actualmente el Despacho Contable está realizando, destacando procesos del Software SEDE.

Algunos de los parámetros que se manejarán para el software se explican en la Unidad 3. Ingeniería de requerimientos, en la página 65, y se señalan a continuación:

Software

- Efectividad del desempeño.
- Eficiencia de desempeño.
- Facilidad de uso.
- Manuales, tutoriales y soporte técnico del fabricante.

Métricas para la eficiencia

- Reducción de errores
- Incremento de flexibilidad
- Aumento en la productividad
- Mejora en la planificación o control de gestión

a. Beneficios del sistema de información

- i. La siguiente tabla muestra un análisis sobre los beneficios obtenidos al mejorar los procesos de cálculo e impresión, es decir sobre la manera en que el software podrá realizar las tareas enlistadas en la primer columna:

	Reducción de errores	Incremento de flexibilidad	Aumento en la productividad	Mejora en la planificación o control de gestión
Mayor precisión en las tareas de cálculo	✓	✓	✓	✓
Capacidad de cambiar rápidamente parámetros del sistema	✓	✓	✓	✓
Mayor velocidad de cálculo e impresión			✓	

Si las tareas y los cálculos del software son más precisos, la disminución de errores ocurrirá, y se podrán realizar otras actividades haciendo uso de los resultados obtenidos, mejorando de forma inmediata el control de los procesos de cálculo.

- ii. La siguiente tabla muestra un análisis sobre los beneficios de acceso y captura de registros:

	Reducción de errores	Incremento de flexibilidad	Aumento de la productividad	Mejora en la planificación o control de gestión
Captura ágil de registros		✓	✓	✓
Mejor acceso a registros en bases de datos grandes	✓	✓	✓	✓
Búsqueda de registros	✓	✓	✓	✓

La tabla de búsqueda de registros es de las más importantes, ya que el Software deberá realizar estas actividades de forma eficaz. El hecho de poder capturar rápidamente los registros nos permite disminuir los tiempos y de forma simultánea nos permite búsquedas más efectivas. En caso de contar con bases de datos grandes, dichos proceso de búsqueda tendrán mayor impacto, ya que se evitará confusiones de información, minimizará los tiempos para permitirnos llevar un control óptimo.

- iii. La siguiente tabla muestra un análisis de los beneficios en el control de recursos:

	Reducción de errores	Incremento de flexibilidad	Aumento del ritmo de actividad	Mejora en la planificación o control de gestión
Reducción en la fuerza de trabajo	✓	✓	✓	✓

Mientras el sistema de información sea eficiente con los puntos anteriores a éste, se tendrá hasta cierto momento, la garantía de no requerir otro tipo de personal para el manejo y obtención de información oportuna.

b. Costos del sistema de información

- i. La siguiente tabla muestra un análisis de los costo de compra e instalación de equipo de cómputo:

	Justificación de costos	Reducción de errores	Aumento de la productividad	Mejora en la planificación o control de gestión
Costo de alquiler o compra de equipo	✓		✓	✓
Costo de instalación de los equipos	✓	✓	✓	✓
Seguridad	✓		✓	

Si se compra equipo de cómputo, también se involucran costos de instalación, sin embargo, se evitan costos como la espera de tiempos largos en la respuesta de procesamiento de información o bien en la espera de adecuaciones para interactuar con otras aplicaciones diarias como lo es Internet. Finalmente invertir en la seguridad de información nos permitirá aumentar la productividad.

ii. La siguiente tabla muestra un análisis de costos de los procesos

	Aumento del ritmo de actividad	Mejora en la planificación o control de gestión
Costos de suministros (electricidad, renta de internet, renta de inmueble, etc.)	✓	✓
Costo de depreciación del hardware		✓

Si se cuentan con suministros de cierta calidad, digamos por ejemplo un servicio de internet adecuado para las necesidades de la empresa, se podría aumentar el ritmo de actividad de la empresa. El hardware, actualmente se ha convertido en algo que seguidamente está cambiando y mejorando, por lo que es importante tener en mente que el equipo que se tiene ahorita, puede que no sea el mejor en 3 años, y aunque no sea indispensable cambiar el equipo se requiere tener en mente una planificación futura de la actualización del equipo.

Haciendo un resumen de los beneficios y costos descritos, podemos decir que:

	Reducción de errores	Incremento de flexibilidad	Aumento del ritmo de actividad	Mejora en la planificación o control de gestión
Reducción en tiempo de captura	✓	✓	✓	✓
Reducción del tiempo de entrega de resultados	✓	✓	✓	✓
Continuidad en el proceso	✓	✓	✓	✓
Reducción en gastos de recursos técnicos		✓	✓	
Mejora en el respaldo de información		✓	✓	

La reducción en el tiempo de captura nos hace posible comenzar a analizar la información en un plazo de tiempo menor, y con la confianza de que no existen errores en los datos procesados. Así mismo, la manera en la que se maneja la información se vuelve más accesible. La reducción del tiempo de entrega de los resultados se puede realizar ya que existe una reducción de los errores desde la captura. La información final se podrá presentar de una manera más clara para el cliente. La continuidad en el proceso permite que la cómoda reanudación de los cálculos por concepto, que se han detenido momentáneamente sin invertir demasiado tiempo en ello. La reducción en los gastos de los recursos técnicos se da ya que no se tendrá que hacer una inversión mayor con la compra de licencias, así mismo el sistema está adaptado a la capacidad del hardware con el que cuenta la empresa. La mejora en el respaldo de la información evitará repetir un proceso a causa de una inactividad no prevista, así como el acceso a históricos de información que en un momento dado podrían requerirse.

5. Evaluación del riesgo técnico

Las innovaciones en los recursos electrónicos y computacionales, actualmente son muy frecuentes, por lo que el sistema, así como todos los demás recursos computacionales y electrónicos tiene un periodo de tiempo en donde serán catalogados como vigentes o innovadores, pero inevitablemente se tendrán que hacer las actualizaciones respectivas conforme a las necesidades del mismo.

6. Consideraciones Legales

Para la realización de las nóminas es importante tomar en cuenta diferentes conceptos y cifras del IMSS y del INFONAVIT. Estos conceptos son importantes ya que establecen los derechos y las

cuotas obrero-patronales. A continuación se desglosan las leyes y reglas importantes:

REGLAMENTO DE LA LEY DEL SEGURO SOCIAL EN MATERIA DE AFILIACIÓN, CLASIFICACIÓN DE EMPRESAS, RECAUDACIÓN Y FISCALIZACIÓN

Nuevo Reglamento publicado en el Diario Oficial de la Federación el 1º de noviembre de 2002

TEXTO VIGENTE

Última reforma publicada DOF 15-07-2005

Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República.

VICENTE FOX QUESADA, Presidente de los Estados Unidos Mexicanos, en ejercicio de la facultad que me confiere el artículo 89, fracción I, de la Constitución Política de los Estados Unidos Mexicanos, con fundamento en los artículos 11 a 15, 16, 70 a 76, 240 a 245 y 304 A a 304 D, de la Ley del Seguro Social; 27, 31, 32, 39 y 40 de la Ley Orgánica de la Administración Pública Federal, he tenido a bien expedir el siguiente

“REGLAMENTO DE LA LEY DEL SEGURO SOCIAL EN MATERIA DE AFILIACIÓN, CLASIFICACIÓN DE EMPRESAS, RECAUDACIÓN Y FISCALIZACIÓN

TÍTULO PRIMERO DISPOSICIONES GENERALES

CAPÍTULO ÚNICO

Artículo 1. El presente Reglamento establece las normas para:

- I. El registro de los patrones y demás sujetos obligados, así como la inscripción de los trabajadores y demás sujetos de aseguramiento del Régimen Obligatorio;
- II. El aseguramiento de los sujetos de continuación o incorporación voluntaria al Régimen Obligatorio y del Seguro de Salud para la Familia;
- III. La determinación y pago de las cuotas, capitales constitutivos, actualización y recargos, a cargo de patrones, demás sujetos obligados y, en su caso, de trabajadores; de los gastos por inscripciones improcedentes y los demás conceptos que el Instituto tenga derecho a exigir a personas no derechohabientes, de conformidad con lo dispuesto por la Ley del Seguro Social y demás disposiciones legales o reglamentarias aplicables;
- IV. La clasificación de las empresas y la determinación de la prima para la cobertura del Seguro de Riesgos de Trabajo, a que se refiere la Ley del Seguro Social;
- V. El dictamen y la corrección sobre el cumplimiento de las obligaciones de los patrones ante el Instituto;
- VI. La comprobación del cumplimiento de las obligaciones legales, reglamentarias y administrativas ante el Instituto, y
- VII. La determinación, imposición y pago de multas, y aplicación de otras sanciones, por infracciones a las disposiciones de la Ley del Seguro Social y sus reglamentos.

Artículo 2. Para efectos de este Reglamento, serán aplicables las definiciones establecidas en el artículo 5 A de la Ley del Seguro Social, así como las siguientes:

- I. Patrón del campo: persona física o moral que realiza actividades agrícolas, ganaderas, forestales o mixtas, independientemente de su naturaleza jurídica o económica y que contrata trabajadores para la explotación de dichas actividades;

II. (Se deroga).

Fracción derogada DOF 15-07-2005

- III. Jornada reducida: tiempo que labora el trabajador, inferior a los máximos establecidos por la Ley Federal del Trabajo, en el cual el salario se determina por unidad de tiempo;
- IV. Semana reducida: número de días que labora el trabajador, inferior al establecido como una semana por la Ley Federal del Trabajo, en el cual el salario se determina por día trabajado;
- V. Unidad de tiempo laborado: medida que utiliza el patrón para la contratación de sus trabajadores y pago de salarios a éstos, como puede ser, entre otros, por hora, día, semana, decena, catorcena, quincena y mes;
- VI. Trabajador doméstico: persona física que presta servicios de aseo, asistencia y demás propios o inherentes al hogar de una persona o familia;
- VII. Riesgos de trabajo terminados: siniestro concluido por alta médica de un trabajador que ha sido declarado apto para continuar sus labores; por el inicio de una incapacidad permanente parcial o total o por la muerte del trabajador siniestrado;
- VIII. Dictamen: documento elaborado por contador público autorizado que consigna la opinión sobre el cumplimiento de las obligaciones del patrón ante el Instituto;
- IX. Corrección: documento que elabora el patrón o sujeto obligado para regularizar el cumplimiento de sus obligaciones ante el Instituto;
- X. Programa informático: el medio de captura, transmisión y recepción de información, que permite a los patrones o sujetos obligados cumplir, a través de medios remotos de comunicación electrónica, con sus obligaciones previstas en la Ley, y
- XI. Entidad receptora: persona moral autorizada para recibir el pago de cuotas y sus accesorios, así como aportaciones voluntarias a los Sistemas de Ahorro para el Retiro, y para efectuar la verificación aritmética de los importes parciales y totales de las cédulas de determinación presentadas por los patrones.

Artículo 3. El registro de los patrones y demás sujetos obligados, la inscripción de los trabajadores y demás sujetos de aseguramiento, la clasificación de empresas y la determinación de la prima de riesgo de trabajo, la determinación y pago de los créditos fiscales a cargo de patrones, trabajadores y demás sujetos obligados y de aseguramiento y en general cualquier otro sujeto de obligaciones establecidas en la Ley y en este Reglamento, así como la comunicación de sus modificaciones salariales y bajas, el registro del contador público autorizado, el aviso para dictaminar, los modelos de opinión y la carta de presentación del dictamen y los demás de cualquier otra índole, se harán en los formatos impresos autorizados que deberán ser publicados en el Diario Oficial de la Federación por el Instituto. Salvo cuando la obligación se cumpla a través de un medio de los señalados en el artículo 5 de este Reglamento.

La reproducción y presentación de dichos formatos podrá realizarse en la forma y términos que señale el Instituto, o en cualquiera de los medios previstos en el último párrafo del artículo 15 de la Ley, de acuerdo con las especificaciones establecidas por el mismo.

En el caso de que se omita presentar la información a que se refieren los párrafos anteriores, en los formatos o medios señalados, no se dará trámite a la solicitud, excepto cuando no se hayan publicado por el Instituto dichos formatos, en cuyo supuesto, se realizará mediante escrito reuniendo todos y cada uno de los requisitos previstos en la Ley y este Reglamento para el cumplimiento de las obligaciones.

Cuando el último día de los plazos señalados en este Reglamento para el cumplimiento de obligaciones, sea día inhábil o viernes se prorrogará el plazo hasta el día hábil siguiente. No se prorrogará el plazo para la presentación de avisos afiliatorios.

Artículo 4. El Instituto podrá conservar en medios magnéticos, digitales, electrónicos, ópticos, magneto ópticos o de cualquier otra naturaleza, la información contenida en la documentación a que se refiere el artículo anterior, presentada en formatos impresos, en relación con el registro de patrones y demás sujetos obligados; inscripción, modificación de salario y baja de trabajadores y demás sujetos de aseguramiento.

El Instituto podrá expedir certificaciones de la información así conservada, en términos de las disposiciones legales aplicables.

Artículo 5. Los patrones y demás sujetos obligados que en los términos del artículo 15 de la Ley, realicen los trámites correspondientes para el cumplimiento de sus obligaciones a través de medios magnéticos, digitales, electrónicos, ópticos, magneto ópticos o de cualquier otra naturaleza, deberán utilizar el número patronal de identificación electrónica, como llave pública de sistemas criptográficos a que se refiere el artículo 15 de este Reglamento en sustitución de su firma autógrafa. Este número se tramitará de conformidad con los lineamientos de carácter general que emita el Consejo Técnico del Instituto, los cuales se publicarán en el Diario Oficial de la Federación.

Para los efectos del párrafo anterior, el Instituto establecerá un sistema de identificación electrónica de tecnología criptográfica.

La información a que se refiere el primer párrafo de este artículo en la que se utilice el número patronal de identificación electrónica en sustitución de la firma autógrafa, así como las certificaciones que de ésta expida el Instituto producirán los mismos efectos que las leyes otorgan a los documentos firmados autógrafamente y, en consecuencia, tendrán el mismo valor probatorio que las disposiciones aplicables les otorgan a éstos.

El Instituto requerirá nuevamente el envío de la información remitida a que se refiere este artículo, en caso de que no se pueda tener acceso a la misma por problemas técnicos.

Para los efectos del párrafo anterior, el patrón o sujeto obligado deberá enviar nuevamente la información en un plazo no mayor de cinco días hábiles contado a partir del requerimiento, a fin de que se le respete la fecha de presentación original. En caso de no hacerlo se tendrá por no presentada.

Artículo 6. En todos los casos el Instituto o las oficinas autorizadas por el mismo, entregarán al patrón o sujeto obligado la constancia correspondiente del trámite realizado, ya sea en forma impresa o en cualquiera de los medios de comunicación electrónica a que se refiere la Ley, en este último caso, dicha constancia tendrá para todos los efectos legales, igual valor probatorio que la impresa.

Tratándose de la constancia que se entregue a través de los medios de comunicación electrónica, ésta deberá reunir los requisitos que establezcan los lineamientos de carácter general que emita el Consejo Técnico del Instituto, los cuales se publicarán en el Diario Oficial de la Federación.

El patrón, en un plazo de tres días hábiles contados a partir de la entrega de la mencionada constancia, por el mismo medio en que le fue entregada, confirmará que la información que proporcionó al Instituto concuerda con la registrada por éste, o bien, realizará las aclaraciones que procedan. En caso contrario, se tendrá por consentida.

De proceder la aclaración, se tendrá por realizado el trámite en la fecha en que fue efectuado originalmente.

LEY DEL SEGURO SOCIAL

Nueva Ley publicada en el Diario Oficial de la Federación el 21 de diciembre de 1995
TEXTO VIGENTE

Última reforma publicada DOF 26-05-2009

Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República.

ERNESTO ZEDILLO PONCE DE LEON, Presidente de los Estados Unidos Mexicanos, a sus habitantes sabed:

Que el H. Congreso de la Unión, se ha servido dirigirme el siguiente

D E C R E T O

"EL CONGRESO DE LOS ESTADOS UNIDOS MEXICANOS, DECRETA:

LEY DEL SEGURO SOCIAL

TITULO PRIMERO

DISPOSICIONES GENERALES

CAPITULO UNICO

Artículo 29. Para determinar la forma de cotización se aplicarán las siguientes reglas:

- I. El mes natural será el período de pago de cuotas;
- II. Para fijar el salario diario en caso de que se pague por semana, quincena o mes, se dividirá la remuneración correspondiente entre siete, quince o treinta respectivamente. Análogo procedimiento será empleado cuando el salario se fije por períodos distintos a los señalados, y
- III. Si por la naturaleza o peculiaridades de las labores, el salario no se estipula por semana o por mes, sino por día trabajado y comprende menos días de los de una semana o el asegurado labora jornadas reducidas y su salario se determina por unidad de tiempo, en ningún caso se recibirán cuotas con base en un salario inferior al mínimo.

Artículo 30. Para determinar el salario diario base de cotización se estará a lo siguiente:

- I. Cuando además de los elementos fijos del salario el trabajador percibiera regularmente otras retribuciones periódicas de cuantía previamente conocida, éstas se sumarán a dichos elementos fijos;
- II. Si por la naturaleza del trabajo, el salario se integra con elementos variables que no puedan ser previamente conocidos, se sumarán los ingresos totales percibidos durante los dos meses inmediatos anteriores y se dividirán entre el número de días de salario devengado en ese período. Si se trata de un trabajador de nuevo ingreso, se tomará el salario probable que le corresponda en dicho período, y
Fracción reformada DOF 20-12-2001
- III. En los casos en que el salario de un trabajador se integre con elementos fijos y variables, se considerará de carácter mixto, por lo que, para los efectos de cotización, se sumará a los elementos fijos el promedio obtenido de los variables en términos de lo que se establece en la fracción anterior.

Artículo 31. Cuando por ausencias del trabajador a sus labores no se paguen salarios, pero subsista la relación laboral, la cotización mensual se ajustará a las reglas siguientes:

- I. Si las ausencias del trabajador son por períodos menores de ocho días consecutivos o interrumpidos, se cotizará y pagará por dichos períodos únicamente en el seguro de enfermedades y maternidad. En estos casos los patrones deberán presentar la aclaración correspondiente, indicando que se trata de cuotas omitidas por ausentismo y comprobarán la falta de pago de salarios respectivos, mediante la exhibición de las listas de raya o de las nóminas correspondientes. Para este efecto el número de días de cada mes se obtendrá restando del total de días que contenga el período de cuotas de que se trate, el número de ausencias sin pago de salario correspondiente al mismo período.
Si las ausencias del trabajador son por períodos de ocho días consecutivos o mayores, el patrón quedará liberado del pago de las cuotas obrero patronales, siempre y cuando proceda en los términos del artículo 37;
Fracción reformada DOF 20-12-2001
- II. En los casos de las fracciones II y III del artículo 30, se seguirán las mismas reglas de la fracción anterior;
- III. En el caso de ausencias de trabajadores comprendidos en la fracción III del artículo 29, cualquiera que sea la naturaleza del salario que perciban, el reglamento determinará lo procedente conforme al criterio sustentado en las bases anteriores, y
- IV. Tratándose de ausencias amparadas por incapacidades médicas expedidas por el Instituto no será obligatorio cubrir las cuotas obrero patronales, excepto por lo que se refiere al ramo de retiro.

Artículo 32. Si además del salario en dinero el trabajador recibe del patrón, sin costo para aquél, habitación o alimentación, se estimará aumentado su salario en un veinticinco por ciento y si recibe ambas prestaciones se aumentará en un cincuenta por ciento.

Cuando la alimentación no cubra los tres alimentos, sino uno o dos de éstos, por cada uno de ellos se adicionará el salario en un ocho punto treinta y tres por ciento.

Artículo 33. Para el disfrute de las prestaciones en dinero, en caso que el asegurado preste servicios a varios patrones se tomará en cuenta la suma de los salarios percibidos en los distintos empleos, cuando ésta sea menor al límite superior establecido en el artículo 28 los patrones cubrirán separadamente los aportes a que estén obligados con base en el salario que cada uno de ellos pague al asegurado.

Cuando la suma de los salarios que percibe un trabajador llegue o sobrepase el límite superior establecido en el artículo 28 de esta Ley, a petición de los patrones, éstos cubrirán los aportes del salario máximo de cotización, pagando entre ellos la parte proporcional que resulte entre el salario que cubre individualmente y la suma total de los salarios que percibe el trabajador.

Artículo 34. Cuando encontrándose el asegurado al servicio de un mismo patrón se modifique el salario estipulado, se estará a lo siguiente:

- I. En los casos previstos en la fracción I del artículo 30, el patrón estará obligado a presentar al Instituto los avisos de modificación del salario diario base de cotización dentro de un plazo máximo de cinco días hábiles, contados a partir del día siguiente a la fecha en que cambie el salario;
- II. En los casos previstos en la fracción II del artículo 30, los patrones estarán obligados a comunicar al Instituto dentro de los primeros cinco días hábiles de los meses de enero, marzo, mayo, julio, septiembre y noviembre, las modificaciones del salario diario promedio obtenido en el bimestre anterior, y
- III. En los casos previstos en la fracción III del artículo 30, si se modifican los elementos fijos del salario, el patrón deberá presentar el aviso de modificación dentro de los cinco días hábiles siguientes de la fecha en que cambie el salario. Si al concluir el bimestre respectivo hubo modificación de los elementos variables que se integran al salario, el patrón presentará al Instituto el aviso de modificación en los términos de la fracción II anterior. El salario diario se determinará, dividiendo el importe total de los ingresos variables obtenidos en el bimestre anterior entre el número de días de salario devengado y sumando su resultado a los elementos fijos del salario diario.

En todos los casos previstos en este artículo, si la modificación se origina por revisión del contrato colectivo, se comunicará al Instituto dentro de los treinta días naturales siguientes a su celebración. Las sociedades cooperativas deberán presentar los avisos de modificación de las percepciones base de cotización de sus socios, de conformidad con lo establecido en este artículo.

Artículo reformado DOF 20-12-2001

Artículo 35. Los cambios en el salario base de cotización derivados de las modificaciones señaladas en el artículo anterior, así como aquellos que por Ley deben efectuarse al salario mínimo, surtirán efectos a partir de la fecha en que ocurrió el cambio, tanto para la cotización como para las prestaciones en dinero.

Artículo 36. Corresponde al patrón pagar íntegramente la cuota señalada para los trabajadores, en los casos en que éstos perciban como cuota diaria el salario mínimo.

Artículo 37. En tanto el patrón no presente al Instituto el aviso de baja del trabajador, subsistirá su obligación de cubrir las cuotas obrero patronales respectivas; sin embargo, si se comprueba que dicho trabajador fue inscrito por otro patrón, el Instituto devolverá al patrón omiso, a su solicitud, el importe de las cuotas obrero patronales pagadas en exceso, a partir de la fecha de la nueva alta.

Artículo 38. El patrón al efectuar el pago de salarios a sus trabajadores, deberá retener las cuotas que a éstos les corresponde cubrir.

Cuando no lo haga en tiempo oportuno, sólo podrá descontar al trabajador cuatro cotizaciones semanales acumuladas, quedando las restantes a su cargo.

El patrón tendrá el carácter de retenedor de las cuotas que descuenta a sus trabajadores y deberá determinar y enterar al Instituto las cuotas obrero patronales, en los términos establecidos por esta Ley y sus reglamentos.

Artículo 39. Las cuotas obrero patronales se causan por mensualidades vencidas y el patrón está obligado a determinar sus importes en los formatos impresos o usando el programa informático, autorizado por el Instituto. Asimismo, el patrón deberá presentar ante el Instituto las cédulas de determinación de cuotas del mes de que se trate, y realizar el pago respectivo, a más tardar el día diecisiete del mes inmediato siguiente.

La obligación de determinar las cuotas deberá cumplirse aun en el supuesto de que no se realice el pago correspondiente dentro del plazo señalado en el párrafo anterior.

Los capitales constitutivos tienen el carácter de definitivos al momento de notificarse y deben pagarse al Instituto, en los términos y plazos previstos en esta Ley.

Artículo reformado DOF 20-12-2001

Artículo 39 A. Sin perjuicio de lo dispuesto en el primer párrafo del artículo anterior, el Instituto, en apoyo a los patrones, podrá entregar una propuesta de cédula de determinación, elaborada con los datos con que cuente de los movimientos afiliatorios comunicados al Instituto por los propios patrones y, en su caso, por sus trabajadores en los términos de la presente Ley.

La propuesta a que se refiere el párrafo anterior podrá ser entregada por el Instituto en documento impreso, o bien, previa solicitud por escrito del patrón o su representante legal, en medios magnéticos, digitales, electrónicos o de cualquier otra naturaleza.

En el caso de los patrones que reciban la propuesta a través de medios magnéticos, digitales, electrónicos, ópticos, magneto óptico o de cualquier otra naturaleza, y opten por usarla para cumplir con su obligación fiscal, invariablemente, para efectos de pago, deberán utilizar el programa informático previamente autorizado por el Instituto, a que se refiere el artículo 39.

Cuando los patrones opten por usar la propuesta en documento impreso para cumplir la obligación fiscal a su cargo, bastará con que la presenten y efectúen el pago de la misma en la oficina autorizada por el Instituto, dentro del plazo señalado en el artículo 39 de esta Ley.

Si los patrones deciden modificar los datos contenidos en las propuestas entregadas, deberán apegarse a las disposiciones de esta Ley y sus reglamentos y anotarán en documento impreso o en el archivo de pago que genere el programa autorizado, todos los elementos necesarios para la exacta determinación de las cuotas, conforme al procedimiento señalado en el reglamento correspondiente.

El hecho de que el patrón no reciba la propuesta de cédula de determinación emitida por el Instituto, no lo exime de cumplir con la obligación de determinar y enterar las cuotas, ni lo libera de las consecuencias jurídicas derivadas del incumplimiento de dichas obligaciones.

Artículo adicionado DOF 20-12-2001

Artículo 39 B. Las cédulas de determinación presentadas al Instituto por el patrón, tendrán para éste el carácter de acto vinculatorio.

Artículo adicionado DOF 20-12-2001

Artículo 39 C. En el caso en que el patrón o sujeto obligado no cubra oportunamente el importe de las cuotas obrero patronales o lo haga en forma incorrecta, el Instituto podrá determinarlas presuntivamente y fijarlas en cantidad líquida, con base en los datos con que cuente o con apoyo en los hechos que conozca con motivo del ejercicio de las facultades de comprobación de que goza como autoridad fiscal o bien a través de los expedientes o documentos proporcionados por otras autoridades fiscales. Esta determinación deberá considerar tanto los saldos a favor del Instituto como los que pudiera haber a favor del patrón debido a errores en lo presentado por este último.

En la misma forma procederá el Instituto, en los casos en que al revisar las cédulas de determinación pagadas por los patrones, detecte errores u omisiones de los que se derive incumplimiento parcial en el pago de las cuotas.

Las cédulas de liquidación que formule el Instituto deberán ser pagadas por los patrones, dentro de los quince días hábiles siguientes a la fecha en que surta efectos su notificación, en los términos del

Código.

En el caso de que el patrón o sujeto obligado, espontáneamente opte por regularizar su situación fiscal, conforme a los programas de regularización que en su caso se establezcan, el Instituto podrá proporcionarle, previa solicitud por escrito, la emisión correspondiente sea de manera impresa, o bien, a través de medios magnéticos, digitales, electrónicos, ópticos, magneto ópticos o de cualquier otra naturaleza.

Artículo adicionado DOF 20-12-2001

Artículo 39 D. Respecto de las cédulas de liquidación emitidas por el Instituto en el supuesto señalado en el segundo párrafo del artículo anterior, el patrón podrá, dentro de los cinco días hábiles siguientes a la fecha en que surta sus efectos la notificación, formular aclaraciones ante la oficina que corresponda a su registro patronal, las que deberán estar debidamente sustentadas y sólo podrán versar sobre errores aritméticos, mecanográficos, avisos afiliatorios presentados previamente por el patrón al Instituto, certificados de incapacidad expedidos por éste o situaciones de hecho que no impliquen una controversia jurídica.

La aclaración administrativa en ningún caso suspenderá o interrumpirá el plazo establecido para efectuar el pago hasta por la suma reconocida. El Instituto contará con veinte días hábiles para resolver la aclaración administrativa que presente el patrón. Si transcurrido este plazo no se resolviera la aclaración, se suspenderá la cuenta de días hábiles señalada en el párrafo anterior.

El Instituto podrá aceptar las aclaraciones debidamente sustentadas que presente el patrón fuera del plazo señalado en este artículo, siempre que, respecto de dicha cédula no se encuentre en trámite de efectividad la garantía otorgada, se haya interpuesto recurso de inconformidad o cualquier otro medio de defensa, o que habiéndolo interpuesto, medie desistimiento.

LEY DEL INSTITUTO DEL FONDO NACIONAL DE LA VIVIENDA PARA LOS TRABAJADORES

Nueva Ley publicada en el Diario Oficial de la Federación el 24 de abril de 1972

TEXTO VIGENTE

Última reforma publicada DOF 01-06-2005

Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República.

LUIS ECHEVERRIA ALVAREZ, Presidente Constitucional de los Estados Unidos Mexicanos, a sus habitantes, sabed:

Que el H. Congreso de la Unión se ha servido dirigirme el siguiente

DECRETO

"El Congreso de los Estados Unidos Mexicanos, Decreta:

LEY DEL INSTITUTO DEL FONDO NACIONAL DE LA VIVIENDA PARA LOS TRABAJADORES

Artículo 29.- Son obligaciones de los patrones:

- I. Proceder a inscribirse e inscribir a sus trabajadores en el Instituto y dar los avisos a que se refiere el Artículo 31 de esta Ley;
Los patrones estarán obligados, siempre que contraten un nuevo trabajador, a solicitarle su número de Clave Unica de Registro de Población.

Párrafo adicionado DOF 06-01-1997

Los patrones inscribirán a sus trabajadores con el salario que perciban al momento de su inscripción;

Párrafo adicionado DOF 06-01-1997

- II. Determinar el monto de las aportaciones del cinco por ciento sobre el salario de los trabajadores a su servicio y efectuar el pago en las entidades receptoras que actúen por cuenta y orden del Instituto, para su abono en la subcuenta de vivienda de las cuentas individuales de los trabajadores previstas en los sistemas de ahorro para el retiro, en los términos de la presente Ley y sus reglamentos, así como en lo conducente, conforme a lo previsto en la Ley del Seguro Social y en la Ley Federal del Trabajo. En lo que corresponde a la integración y cálculo de la base y límite superior salarial para el pago de aportaciones, se aplicará lo contenido en la Ley del Seguro Social.

Estas aportaciones son gastos de previsión de las empresas y forman parte del patrimonio de los trabajadores.

Los patrones, al realizar el pago, deberán proporcionar la información relativa a cada trabajador en la forma y con la periodicidad que al efecto establezca la presente Ley y, en lo aplicable, la Ley del Seguro Social y la Ley de los Sistemas de Ahorro para el Retiro.

El registro sobre la individualización de los recursos de la subcuenta de vivienda de las cuentas individuales de los sistemas de ahorro para el retiro, estará a cargo de las administradoras de fondos para el retiro, en los términos que se establecen en la Ley de los Sistemas de Ahorro para el Retiro y su Reglamento. Lo anterior, independientemente de los registros individuales que determine llevar el Instituto.

Es obligación del patrón pagar las aportaciones por cada trabajador mientras exista la relación laboral y subsistirá hasta que se presente el aviso de baja correspondiente. Si se comprueba que dicho trabajador fue inscrito por otro patrón, el Instituto devolverá al patrón omiso, a su solicitud, el importe de las aportaciones pagadas en exceso, a partir de la fecha de la nueva alta;

Fracción reformada DOF 24-02-1992, 22-07-1994, 06-01-1997

- III.** Hacer los descuentos a sus trabajadores en sus salarios, conforme a lo previsto en los artículos 97 y 110 de la Ley Federal del Trabajo, que se destinen al pago de abonos para cubrir préstamos otorgados por el Instituto, así como enterar el importe de dichos descuentos en las entidades receptoras que actúen por cuenta y orden del Instituto, en la forma y términos que establece esta Ley y sus disposiciones reglamentarias. La integración y cálculo de la base salarial para efectos de los descuentos será la contenida en la fracción II del presente artículo.

A fin de que el Instituto pueda individualizar dichos descuentos, los patrones deberán proporcionarle la información relativa a cada trabajador en la forma y periodicidad que al efecto establezcan esta Ley y sus disposiciones reglamentarias;

Fracción reformada DOF 07-01-1982, 06-01-1997

- IV.** Proporcionar al Instituto los elementos necesarios para precisar la existencia, naturaleza y cuantía de las obligaciones a su cargo, establecidas en esta Ley y sus disposiciones reglamentarias;

Fracción adicionada DOF 06-01-1997

- V.** Permitir las inspecciones y visitas domiciliarias que practique el Instituto, las que se sujetarán a lo establecido por esta Ley, el Código Fiscal de la Federación y sus disposiciones reglamentarias. A efecto de evitar duplicidad de acciones, el Instituto podrá convenir con el Instituto Mexicano del Seguro Social la coordinación de estas acciones fiscales;

Fracción adicionada DOF 06-01-1997

- VI.** Atender los requerimientos de pago e información que les formule el Instituto, de conformidad con las disposiciones legales y reglamentarias correspondientes;

Fracción adicionada DOF 06-01-1997

- VII.** Expedir y entregar, semanal o quincenalmente, a cada trabajador constancia escrita del número de días trabajados y del salario percibido, conforme a los períodos de pago establecidos, tratándose de patrones que se dediquen en forma permanente o esporádica a la actividad de la construcción.

Asimismo, deberán cubrir las aportaciones, aun en el caso de que no sea posible determinar el o los trabajadores a quienes se deban aplicar, en cuyo caso su monto se depositará en una cuenta específica que se manejará en los mismos términos que los recursos individualizados del Fondo Nacional de la Vivienda, hasta en tanto se esté en posibilidad de individualizar los pagos a favor de sus titulares, en los términos de esta Ley. Lo anterior, sin perjuicio de que aquellos trabajadores que acrediten sus derechos, se les abonen a sus cuentas individuales de los sistemas de ahorro para el retiro, los importes que les correspondan.

La administradora de fondos para el retiro en la que el trabajador se encuentre registrado tendrá a petición del mismo, la obligación de individualizar las aportaciones a que se refiere esta fracción contra la presentación de las constancias mencionadas;

Fracción adicionada DOF 06-01-1997

- VIII.** Presentar al Instituto copia con firma autógrafa del informe sobre la situación fiscal del contribuyente con los anexos referentes a las contribuciones por concepto de aportaciones patronales de conformidad con lo dispuesto en el Reglamento del Código Fiscal de la Federación, cuando en los términos de dicho Código, estén obligados a dictaminar por contador público autorizado sus estados financieros.

Cualquier otro patrón podrá optar por dictaminar por contador público autorizado el cumplimiento de sus obligaciones ante el Instituto en los términos de las disposiciones reglamentarias correspondientes, y

Fracción adicionada DOF 06-01-1997

- IX.** - Las demás previstas en la Ley y sus reglamentos.

Fracción adicionada DOF 06-01-1997

La obligación de efectuar las aportaciones y hacer los descuentos a que se refieren las fracciones II y III anteriores, se suspenderá cuando no se paguen salarios por ausencias en los términos de la Ley del Seguro Social, siempre que se dé aviso oportuno al Instituto, en conformidad al artículo 31. Tratándose de incapacidades expedidas por el Instituto Mexicano del Seguro Social, subsistirá la obligación del pago de aportaciones.

Párrafo adicionado DOF 06-01-1997

En caso de sustitución patronal, el patrón sustituido será solidariamente responsable con el nuevo de las obligaciones derivadas de esta Ley, nacidas antes de la fecha de la sustitución, hasta por el término de dos años, concluido el cual todas las responsabilidades serán atribuibles al nuevo patrón.

Párrafo adicionado DOF 06-01-1997

Artículo 30.- Las obligaciones de efectuar las aportaciones y enterar los descuentos a que se refiere el Artículo anterior, así como su cobro, tienen el carácter de fiscales.

El Instituto del Fondo Nacional de la Vivienda para los Trabajadores, en su carácter de organismo fiscal autónomo, está facultado, en los términos del Código Fiscal de la Federación, para:

I.- Determinar, en caso de incumplimiento, el importe de las aportaciones patronales y de los descuentos omitidos, así como calcular su actualización y recargos que se generen, señalar las bases para su liquidación, fijarlos en cantidad líquida y requerir su pago. Para este fin podrá ordenar y practicar, con el personal que al efecto designe, visitas domiciliarias, auditorías e inspecciones a los patrones, requiriéndoles la exhibición de libros y documentos que acrediten el cumplimiento de las obligaciones que en materia habitacional les impone esta Ley.

Párrafo reformado DOF 22-07-1994

Las facultades del Instituto para comprobar el cumplimiento de las disposiciones de esta Ley, así como para determinar las aportaciones omitidas y sus accesorios, se extinguen en el término de cinco años no sujeto a interrupción contado a partir de la fecha en que el propio Instituto tenga conocimiento del hecho generador de la obligación. El plazo señalado en este párrafo sólo se suspenderá cuando se interponga el recurso de inconformidad previsto en esta Ley o se entable juicio ante el Tribunal Federal de Justicia Fiscal y Administrativa.

Párrafo reformado DOF 31-12-2000

La prescripción de los créditos fiscales correspondientes se sujetará a lo dispuesto en el Código Fiscal de la Federación;

Fracción adicionada DOF 24-02-1992

II.- Recibir en sus oficinas o a través de las entidades receptoras, los pagos que deban efectuarse conforme a lo previsto por este artículo.

Las entidades receptoras son aquellas autorizadas por los institutos de seguridad social para recibir el pago de las cuotas del seguro de retiro, cesantía en edad avanzada y vejez, previsto en la Ley del Seguro Social, de aportaciones y descuentos de vivienda al Fondo Nacional de la Vivienda y de aportaciones voluntarias.

El Instituto deberá abonar a la subcuenta de vivienda del trabajador el importe de las aportaciones recibidas conforme a este artículo, así como los intereses determinados de conformidad a lo previsto en el artículo 39, que correspondan al período de omisión del patrón. En caso de que no se realice el abono dentro de los diez días hábiles siguientes a la fecha de cobro efectivo, los intereses se calcularán hasta la fecha en que éste se acredite en la subcuenta de vivienda del trabajador;

Fracción adicionada DOF 24-02-1992. Reformada 06-01-1997

III. Realizar por sí o a través de la Secretaría de Hacienda y Crédito Público el cobro y la ejecución correspondiente a las aportaciones patronales y a los descuentos omitidos, sujetándose a las normas del Código Fiscal de la Federación;

Fracción adicionada DOF 24-02-1992

IV.- Resolver en los casos en que así proceda, los recursos previstos en el Código Fiscal de la Federación relativos al procedimiento administrativo de ejecución, así como las solicitudes de prescripción y caducidad planteadas por los patrones;

Fracción adicionada DOF 24-02-1992. Reformada 06-01-1997

V.- Requerir a los patrones que omitan el cumplimiento de las obligaciones que esta Ley establece, la información necesaria para determinar la existencia o no de la relación laboral con las personas a su servicio, así como la que permita establecer en forma presuntiva y conforme al procedimiento que al efecto el Instituto señale, el monto de las aportaciones omitidas.

La Secretaría de Hacienda y Crédito Público y el Instituto, indistintamente, sancionarán aquellos casos en que el incumplimiento de las obligaciones que esta Ley establece, origine la omisión total o parcial en el pago de las aportaciones y el entero de los descuentos, en los términos del Código Fiscal de la Federación.

Párrafo adicionado DOF 24-02-1992. Reformado DOF 22-07-1994

Previa solicitud del Instituto, la Secretaría de Hacienda y Crédito Público, el Instituto Mexicano del Seguro Social y las autoridades fiscales locales, en los términos de los convenios de coordinación que al efecto se celebren, indistintamente y conforme a las disposiciones legales aplicables, están facultados para determinar, en caso de incumplimiento, el importe de las aportaciones patronales y de los descuentos omitidos. Para estos efectos, podrán ordenar y practicar visitas domiciliarias, auditorías e inspecciones a los patrones y requerir la exhibición de los libros y documentos que acrediten el cumplimiento de las obligaciones que en materia habitacional les impone esta Ley.

Párrafo adicionado DOF 22-07-1994

Fracción adicionada DOF 24-02-1992

VI.- Determinar la existencia, contenido y alcance de las obligaciones incumplidas por los patrones y demás sujetos obligados en los términos de esta Ley y demás disposiciones relativas, para lo cual podrá aplicar los datos con los que cuente, en función del último mes cubierto o con apoyo en los hechos que conozca con motivo del ejercicio de las facultades de comprobación de que goza como autoridad o bien a través de los expedientes o documentos proporcionados por otras autoridades fiscales;

Fracción adicionada DOF 06-01-1997

VII.- Ordenar y practicar, en los casos de sustitución patronal, las investigaciones correspondientes así como emitir los dictámenes respectivos;

Fracción adicionada DOF 06-01-1997

VIII.- Revisar los dictámenes formulados por contadores públicos sobre el cumplimiento de las disposiciones contenidas en esta Ley y sus disposiciones reglamentarias respectivas;

Fracción adicionada DOF 06-01-1997

IX.- Hacer efectivas las garantías del interés fiscal ofrecidas a favor del Instituto, incluyendo fianza, en los términos del Código Fiscal de la Federación;

Fracción adicionada DOF 06-01-1997

X.- Conocer y resolver las solicitudes de devolución y compensación de cantidades pagadas indebidamente o en exceso, de conformidad a lo previsto en las disposiciones legales y reglamentarias, y

Fracción adicionada DOF 06-01-1997

XI.- Las demás previstas en la Ley.

Fracción adicionada DOF 06-01-1997

Párrafo reformado DOF 24-02-1992

Párrafos derogados DOF 24-02-1992

Artículo reformado DOF 30-12-1983

Artículo 31.- Para la inscripción de los patrones y de los trabajadores se deberá proporcionar la información que se determine en esta Ley y sus disposiciones reglamentarias correspondientes.

Los patrones deberán dar aviso al Instituto de los cambios de domicilio y de denominación o razón social, aumento o disminución de obligaciones fiscales, suspensión o reanudación de actividades, clausura, fusión, escisión, enajenación y declaración de quiebra y suspensión de pagos. Asimismo harán del conocimiento del Instituto las altas, bajas, modificaciones de salarios, ausencias e incapacidades y demás datos de los trabajadores, necesarios al Instituto para dar cumplimiento a las obligaciones contenidas en este artículo. El Instituto podrá convenir con el Instituto Mexicano del Seguro Social los términos y requisitos para simplificar y unificar los procesos antes descritos.

El registro de los patrones y la inscripción de los trabajadores, así como los demás avisos a que se refieren los párrafos anteriores, deberán presentarse al Instituto dentro de un plazo no mayor de cinco días hábiles, contados a partir de que se den los supuestos a que se refiere el párrafo anterior.

Los cambios en el salario base de aportación y de descuentos, surtirán efectos a partir de la fecha en que éstos ocurran.

La información a que se refiere este artículo, podrá proporcionarse en dispositivos magnéticos o de telecomunicación, en los términos que señale el Instituto.

Los documentos, datos e informes que los trabajadores, patrones y demás personas proporcionen al Instituto en cumplimiento de las obligaciones que les impone esta Ley, serán estrictamente confidenciales y no podrán comunicarse o darse a conocer en forma nominativa e individual, salvo cuando se trate de juicios y procedimientos en que el Instituto fuere parte y en los casos previstos por Ley.

Artículo reformado DOF 06-01-1997

Artículo 32.- En el caso de que el patrón no cumpla con la obligación de inscribir al trabajador, o de enterar al Instituto las aportaciones y descuentos a los salarios, los trabajadores tienen derecho de acudir al Instituto y proporcionarle los informes correspondientes; sin que ello releve al patrón del cumplimiento de su obligación y lo exima de las sanciones en que hubiere incurrido.

Artículo reformado DOF 06-01-1997

Artículo 33.- El Instituto podrá registrar a los patrones e inscribir a los trabajadores y precisar su salario base de aportación, aun sin previa gestión de los interesados y sin que ello releve al patrón de su obligación y de las responsabilidades y sanciones por infracciones en que hubieren incurrido.

Artículo reformado DOF 06-01-1997

Artículo 34.- El trabajador tendrá derecho, en todo momento, a solicitar información a las administradoras de fondos para el retiro sobre el monto de las aportaciones registradas a su favor.

La información anterior, también podrá solicitarla el trabajador a través del Instituto o del patrón al que preste sus servicios.

Tratándose de los trabajadores que reciban crédito de vivienda por parte del Instituto, tendrán derecho a solicitar y obtener información directa de éste o a través del patrón al que preste sus servicios sobre el monto de los descuentos, incluyendo las aportaciones aplicadas a cubrir su crédito, y el saldo del mismo.

Al terminarse la relación laboral, el patrón deberá entregar al trabajador una constancia de la clave de su registro.

Artículo reformado DOF 07-01-1982, 06-01-1997

Artículo 35.- El pago de las aportaciones y descuentos señalados en el artículo 29 será por mensualidades vencidas, a más tardar los días diecisiete del mes inmediato siguiente.

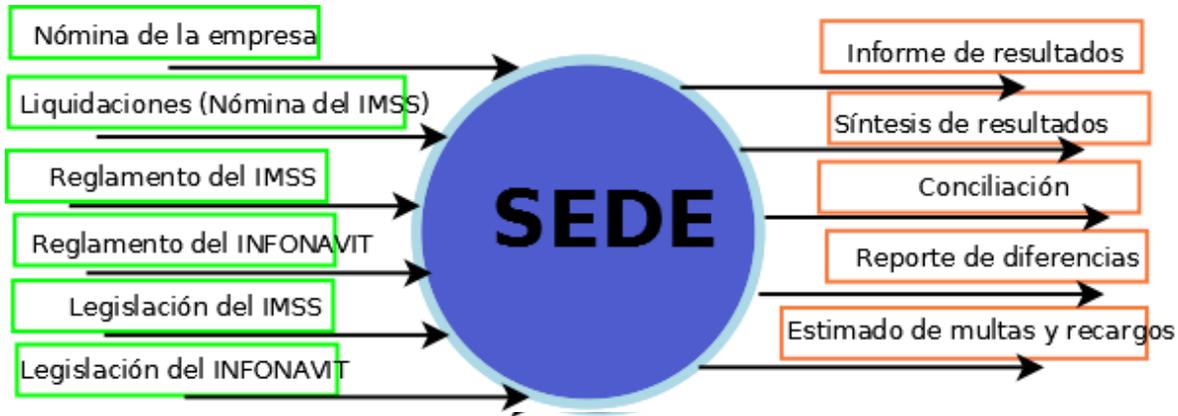
El Instituto podrá emitir y notificar liquidaciones para el cobro de las aportaciones y descuentos a que se refiere el artículo 29. Estas liquidaciones podrán ser emitidas y notificadas por el Instituto Mexicano del Seguro Social conjuntamente con las liquidaciones del seguro de retiro, cesantía en edad avanzada y vejez, previo convenio de coordinación entre ambas instituciones.

Artículo reformado DOF 28-12-1989, 24-02-1992, 06-01-1997

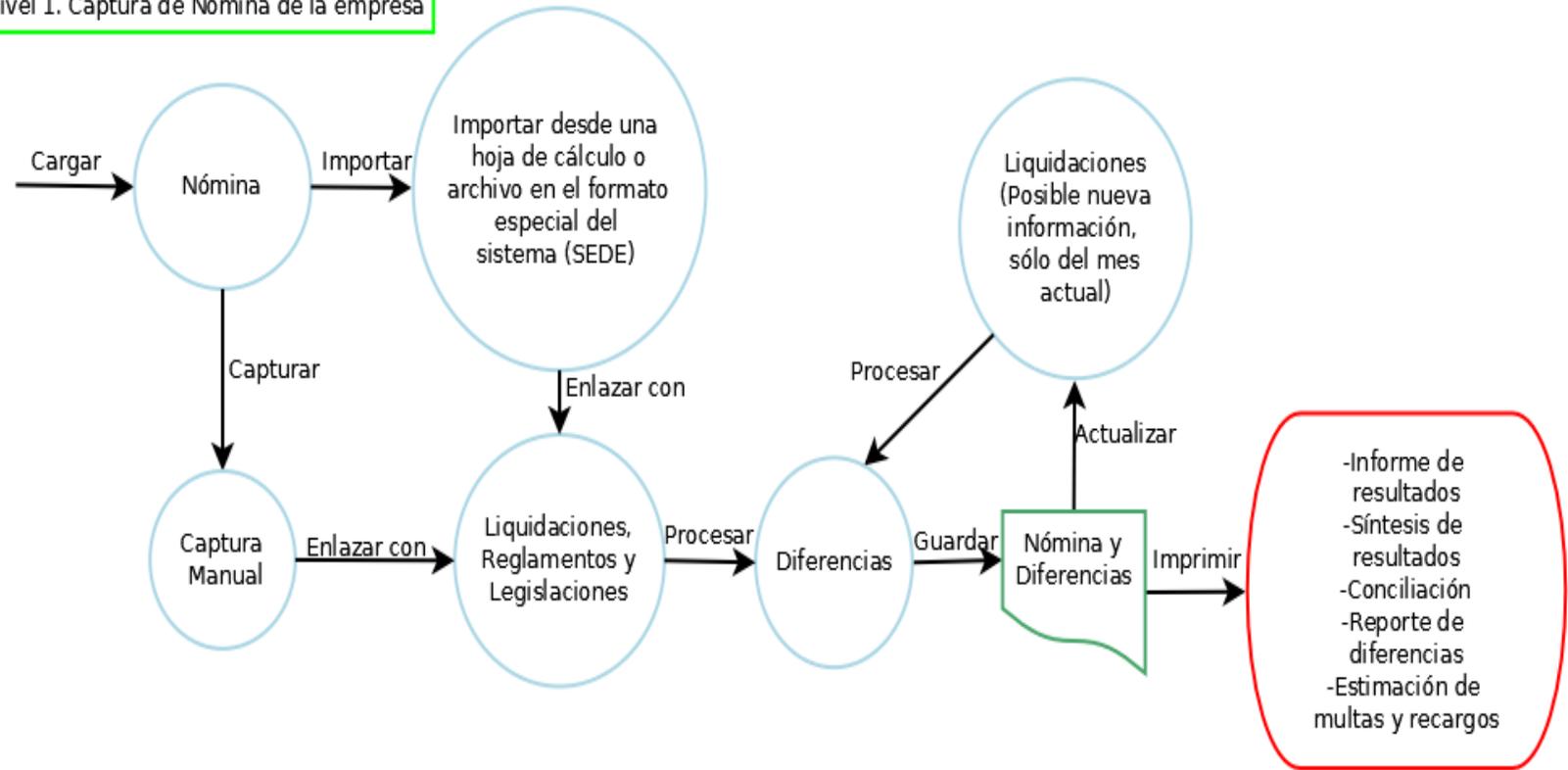
Diseño y modelado del sistema propuesto

Diagrama de flujo de datos del sistema

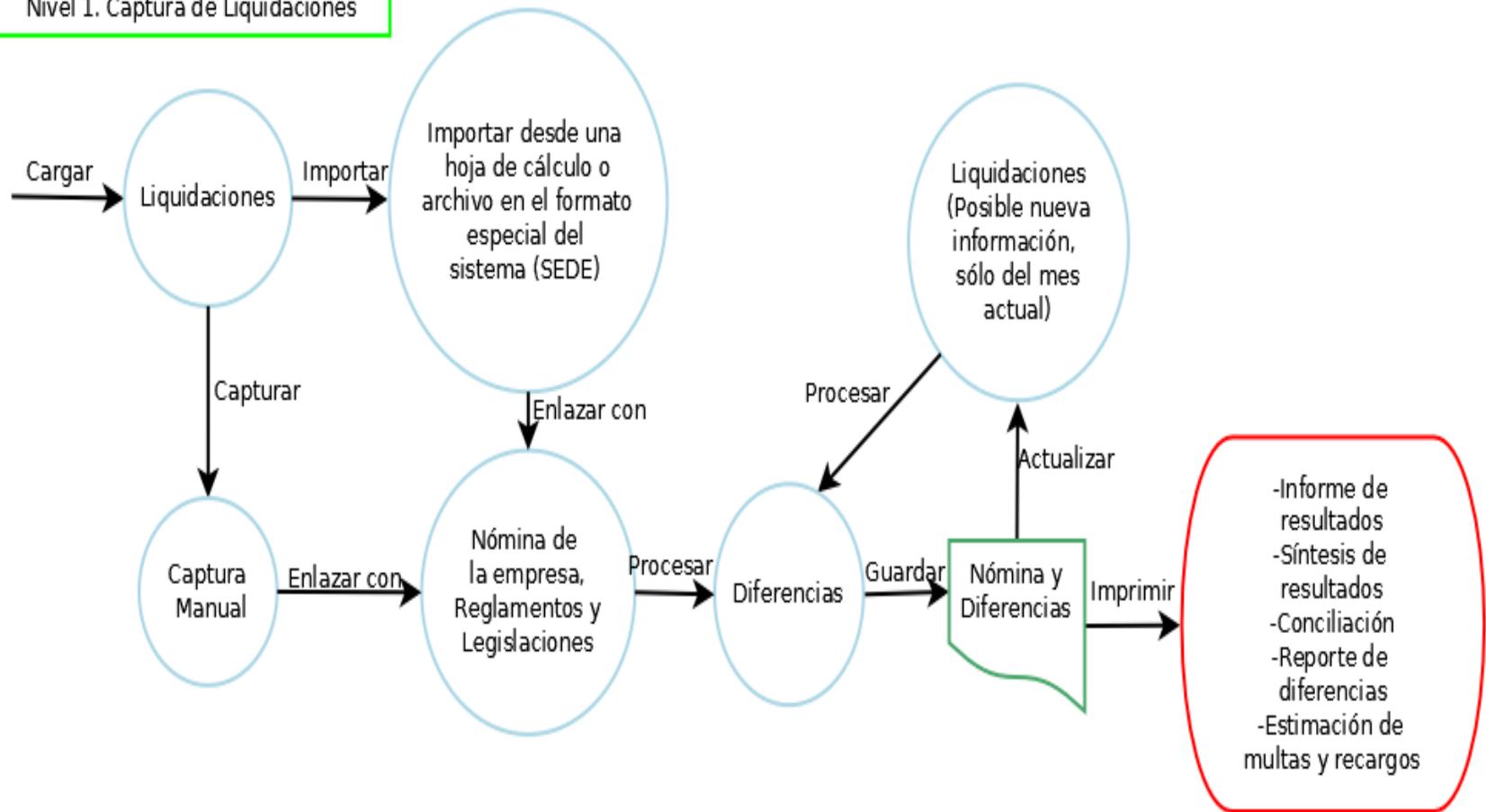
Nivel 0.



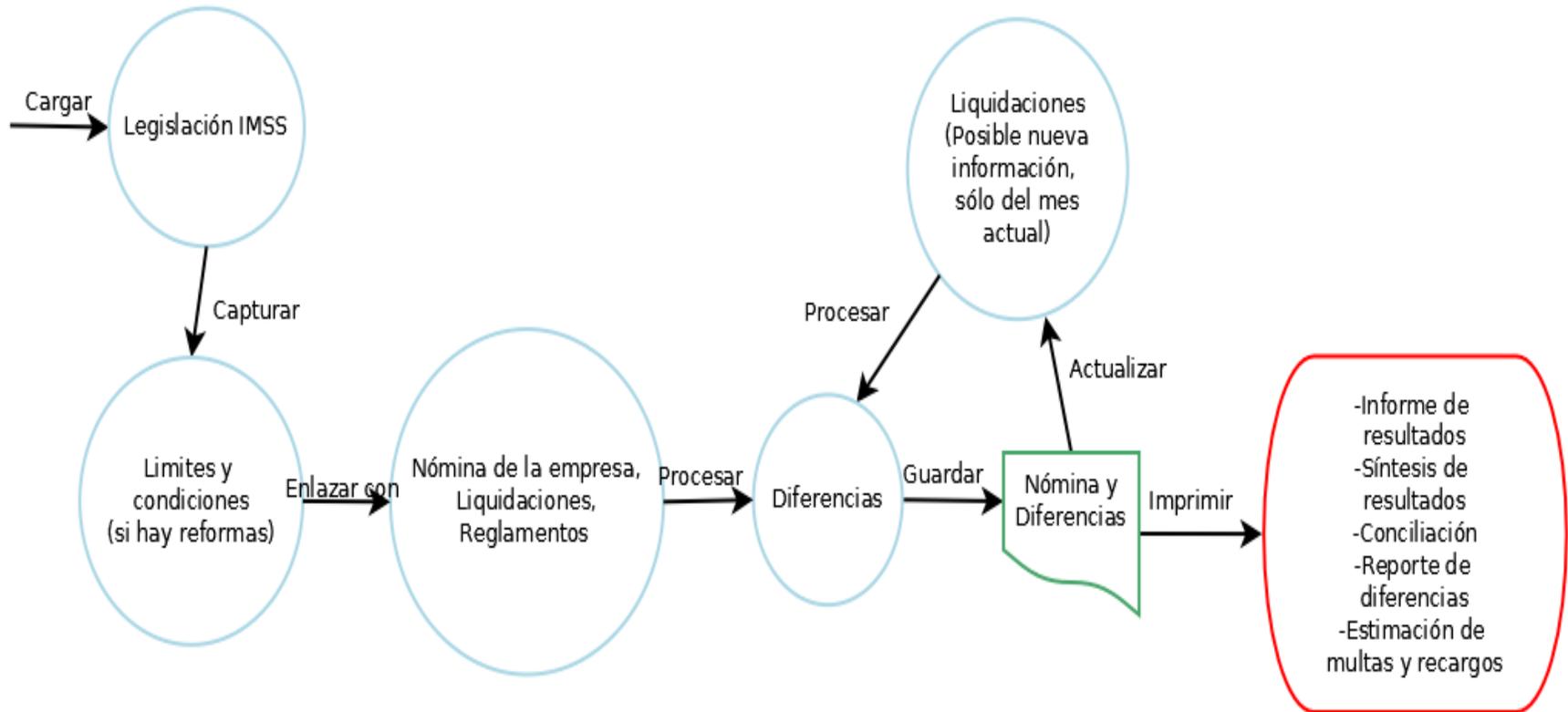
Nivel 1. Captura de Nómina de la empresa



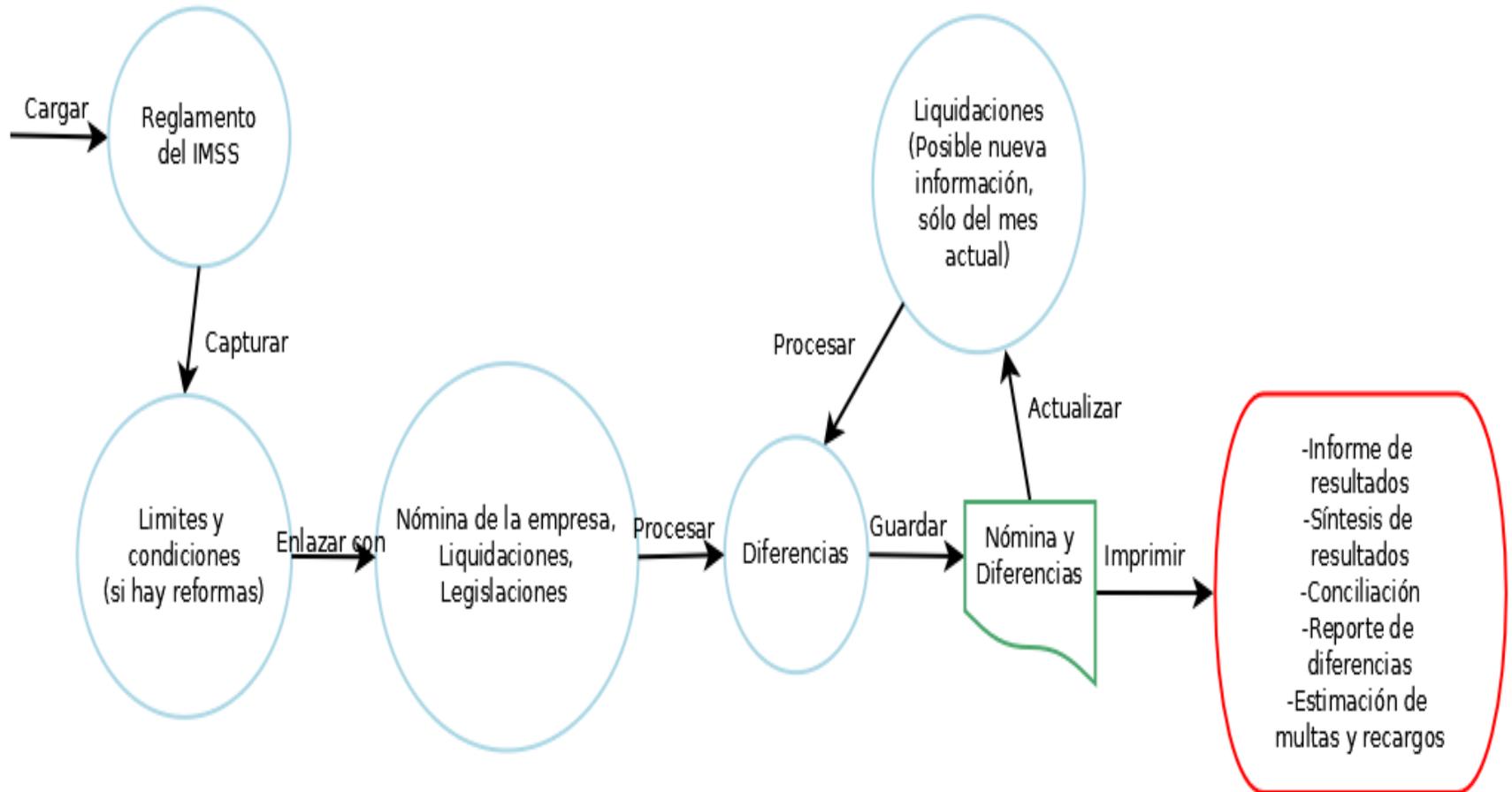
Nivel 1. Captura de Liquidaciones



Nivel 1. Captura de Legislación del IMSS



Nivel 1. Captura de Reglamento del IMSS



Modelos de contexto. Tabla visual de contenidos.

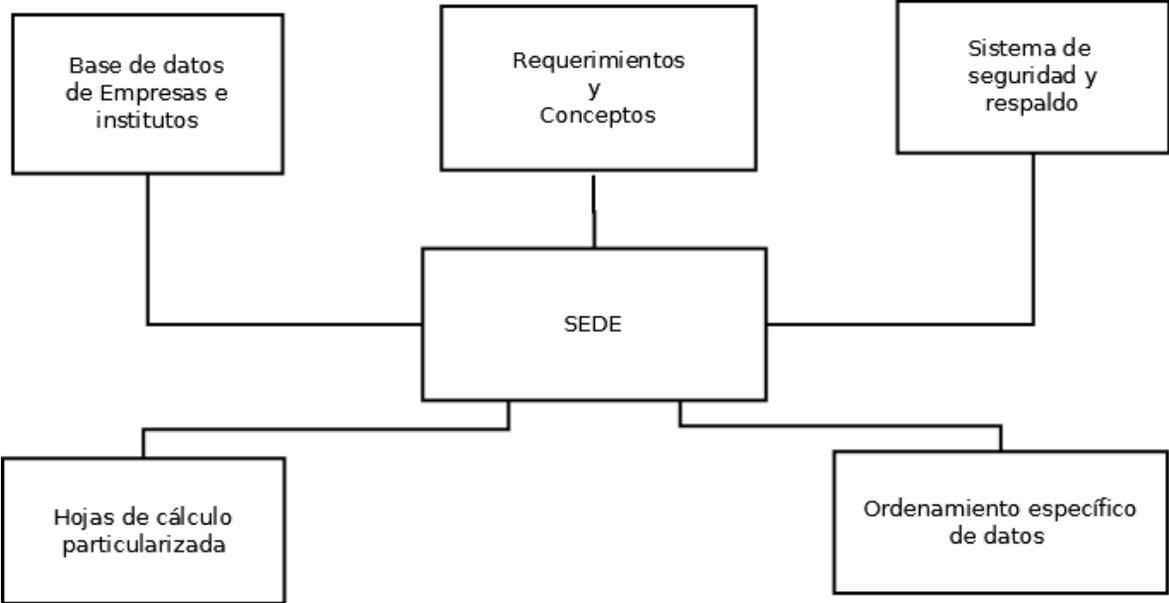


Definiciones:

1. SEDE: (Sistema de Estimación de Diferencias Empresariales) Es el sistema que estará encargado de calcular de forma rápida y eficiente los pagos de seguro (IMSS) por parte de una empresa a partir de su nómina.
2. Captura de nómina: Es la interfaz a través de la cual se captura la nómina de una empresa con información a la fecha sobre cotizaciones ideales y presentadas.
 - 2.1. Importación de documentos: Tiene como objetivo importar la información de la nómina a partir de un archivo, hoja de cálculo o separado por comas.
 - 2.2. Captura manual: En caso que la información no esté en formato digital, esta se capturará manualmente; éste módulo tiene como objetivo facilitar esta tarea presentando campos claros y detallando la información capturada, con el fin de evitar equivocaciones o duplicidad de la información.
3. Captura de Liquidaciones: Es la interfaz en la que se capturará de forma detallada la información entorno al IMSS y/o INFONAVIT.
 - 3.1. Importación de documentos: Tiene como objetivo importar la información de las liquidaciones a partir de un archivo, hoja de cálculo o separado por comas.
 - 3.2. Captura manual: En caso que la información no esté en formato digital, esta se capturará manualmente; éste módulo tiene como objetivo facilitar esta tarea presentando campos claros y detallando la información capturada, con el fin de evitar equivocaciones o duplicidad de la información.
4. Definición de legislación y reglamentos del IMSS e INFONAVIT: Tiene como objetivo añadir las restricciones y nuevas variables requeridas por la legislación para calcular pagos.
 - 4.1. Modificación de reformas: En caso de cambios en restricciones, permite actualizar estos cambios al proceso actual.
 - 4.2. Consulta: Verifica que el proceso y los datos calculados sean correctos de acuerdo a la fecha y reformas.
5. Administración de cuentas: La gestión de las personas que pueden usar el sistema.

- 5.1. Gestión de usuarios: Administra los nombres y claves de los usuarios que pueden usar el sistema y la información que se maneja.
- 5.2. Asociación: Gestiona qué usuario puede dar seguimiento a determinado proceso de una empresa, así como quién o quienes pueden consultar y sólo consultar esta información en un momento dado.
6. Generación de informes: Tiene como objetivo simplificar la forma en que se presenta la información así como mostrar los resultados del proceso, cada módulo de este programa tiene como objetivo presentar la información indicada por su nombre:
 - 6.1. Informes de resultados
 - 6.2. Conciliación
 - 6.3. Reporte de diferencias
 - 6.4. Estimación de multas y recargos

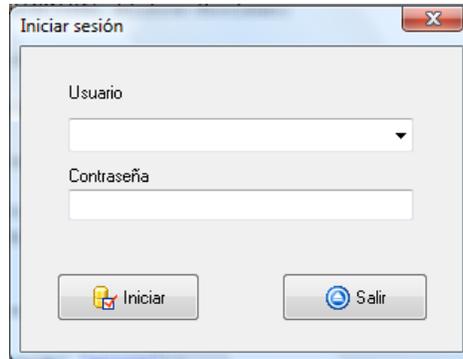
Modelo arquitectónico sencillo del sistema



PROPUESTA DE DISEÑOS DE ENTRADA Y SALIDAS

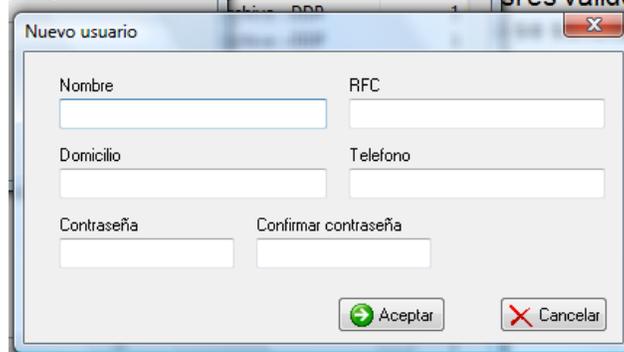
Diseño de entradas

La primer entrada es el nombre de usuario y contraseña, si es válido permite continuar, de lo contrario en caso de no existir el usuario se solicita crearlo.



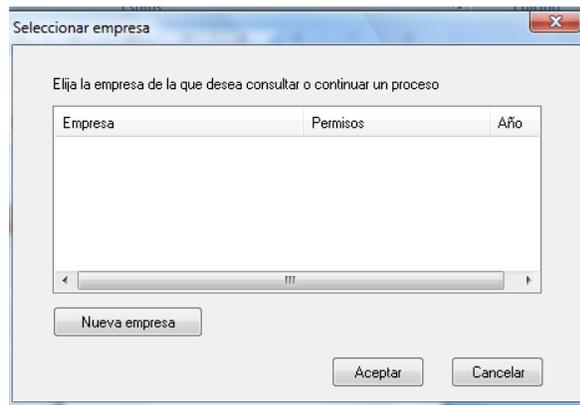
The screenshot shows a dialog box titled "Iniciar sesión". It has a standard Windows-style title bar with a close button. The main area contains two text input fields. The first is labeled "Usuario" and the second is labeled "Contraseña". Below these fields are two buttons: "Iniciar" with a key icon and "Salir" with a circular arrow icon.

Esta es la ventana de entrada de un nuevo usuario, se solicita su nombre, rfc, domicilio, teléfono, y la contraseña que le permitirá iniciar sesión. El usuario principal es agregado durante la instalación del software.



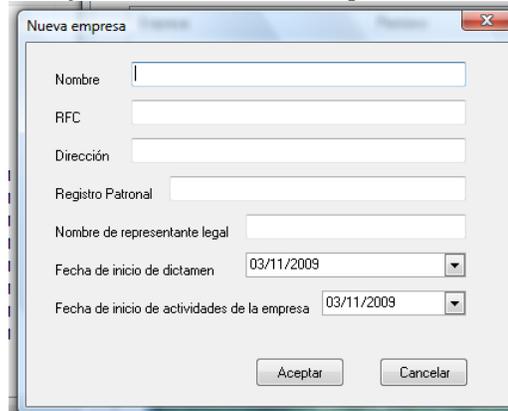
The screenshot shows a dialog box titled "Nuevo usuario". It has a standard Windows-style title bar. The main area contains six text input fields arranged in three rows. The first row has "Nombre" and "RFC". The second row has "Domicilio" and "Telefono". The third row has "Contraseña" and "Confirmar contraseña". At the bottom right are two buttons: "Aceptar" with a green checkmark icon and "Cancelar" with a red X icon.

A cada usuario le corresponde una empresa o varias empresas, concretamente sus procesos, por año, así un usuario dispone de hacer modificaciones y actualizaciones a un proceso o simplemente sólo puede consultar los datos. Aparece una ventana con una lista de procesos disponibles, el usuario seleccionará uno.



The screenshot shows a dialog box titled "Seleccionar empresa". It has a standard Windows-style title bar. The main area contains a text prompt: "Elija la empresa de la que desea consultar o continuar un proceso". Below this is a table with three columns: "Empresa", "Permisos", and "Año". The table is currently empty. Below the table is a horizontal scrollbar. At the bottom left is a button labeled "Nueva empresa". At the bottom right are two buttons: "Aceptar" and "Cancelar".

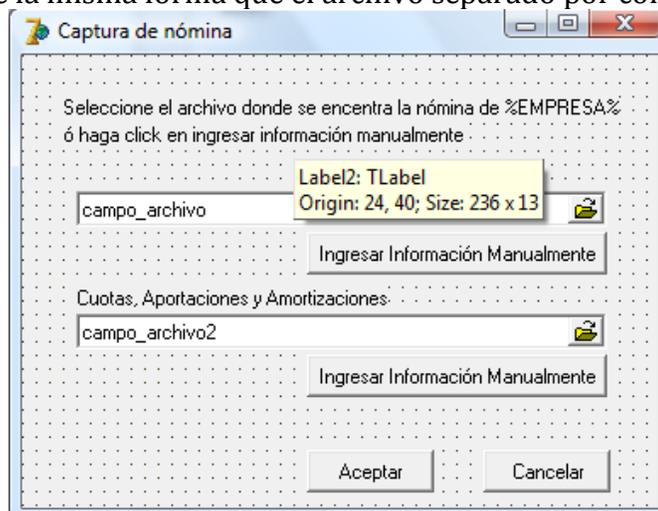
Si se inicia un nuevo proceso, se selecciona la opción de nueva empresa, donde se registrará a esta, si la empresa ya existe pero se desea un nuevo proceso, bastará con hacer dobleclick en la empresa de la lista para llenar los campos de forma automática y continuar con la captura de nómina.



Cada vez que una nueva es creada, o se inicia un nuevo proceso, o se calculan los procesos en el mes siguiente, se procede a la captura de nómina correspondiente al mes.

El software SEDE, trabaja sólo con dos tipos entrada, que es una tabla de nomina de la empresa y otra tabla que contiene la información sobre cuotas y aportaciones, información dada por el IMSS e INFONAVIT, la información dada es por cada empleado de la empresa, y no necesariamente es la misma cada mes.

La entrada, puede ser por archivo, cuya estructura se basa en tokens separados por comas (,) que permiten un “acomodo” sencillo de los datos en el sistema. Sin embargo, existe la posibilidad de que la empresa no aporte la información en este formato, así que la entrada será una captura manual de la información que generará eventualmente un “stream” en memoria que funcionará de la misma forma que el archivo separado por comas.



La tabla muestra un ejemplo de la presentación a modo de hoja de cálculo el contenido de un archivo recibido por el software.

RFC	#IMSS	NOMBRE	SUELDO	PERCEPCIONESVAR	PERCEP1	...	CUOTAS

Automáticamente, se modelará la base de datos particular para el proceso, de acuerdo con las percepciones variables y políticas de la empresa de la que se está llevando a cabo un cálculo de diferencias.

Diseño de salidas

Las salidas están dadas en primera instancia en la ventana principal, se muestran los datos que han sido recibidos directamente por la captura de nómina además de información, como el factor de integración, el total de percepciones, y el total de cuotas.

Y se muestra un resumen con las diferencias calculadas.

The screenshot displays the SEDE software interface. It features a top navigation bar with options like 'Almacenar cambios', 'Captura de datos del mes', 'Búsqueda rápida', 'Búsqueda', 'Criterio', 'Búsqueda avanzada', 'Cerrar proceso', and 'Cerrar sesión'. The main area is divided into several sections:

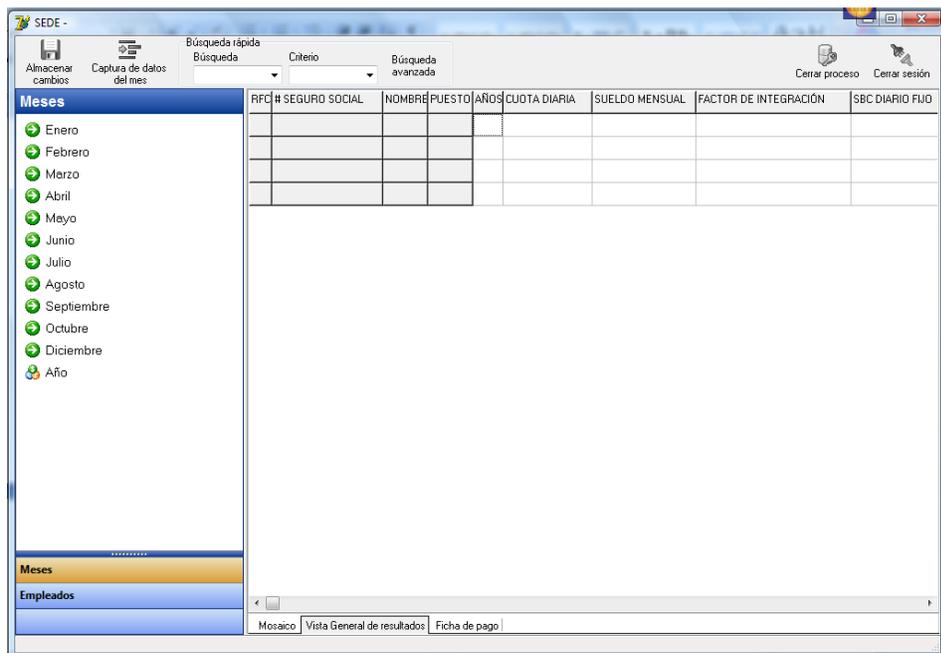
- Empleados:** A table with columns for RFC, Nombre, and No. S... (Social Security Number).
- Sección Nómina de la empresa:** A table with columns for SUELDO MENSUAL, SUELDO DIARIO, and PERCEPCIONES.
- Sección Cuotas, Aportaciones, Amortizaciones:** A table with columns for DIAS, CUOTA RCV, No. CREDITO DE VIVIENDA, and SUMA INFO.
- Resumen de diferencias:** A table with columns for AÑOS, FACTOR DE INTEGRACION, SUELDO MENSUAL, SUELDO DIARIO, PERCEPCIONES, SUMA CUOTAS, IVCV (DIAS), and IVCV (CLAV).
- Datos de empleado%:** A section for employee details including RFC, No. Seguro social, Puesto, Fecha de alta, No. Tra., Cuota Diaria, and Años trabajados (%primerdialmes%).
- Meses:** A section for monthly data.
- Empleados:** A section for employee selection.

At the bottom, there are view options: 'Mosaico', 'Vista General de resultados', and 'Ficha de pago'.



Los datos pueden filtrarse de acuerdo con el mes seleccionándolo de la lista en el lado izquierdo.

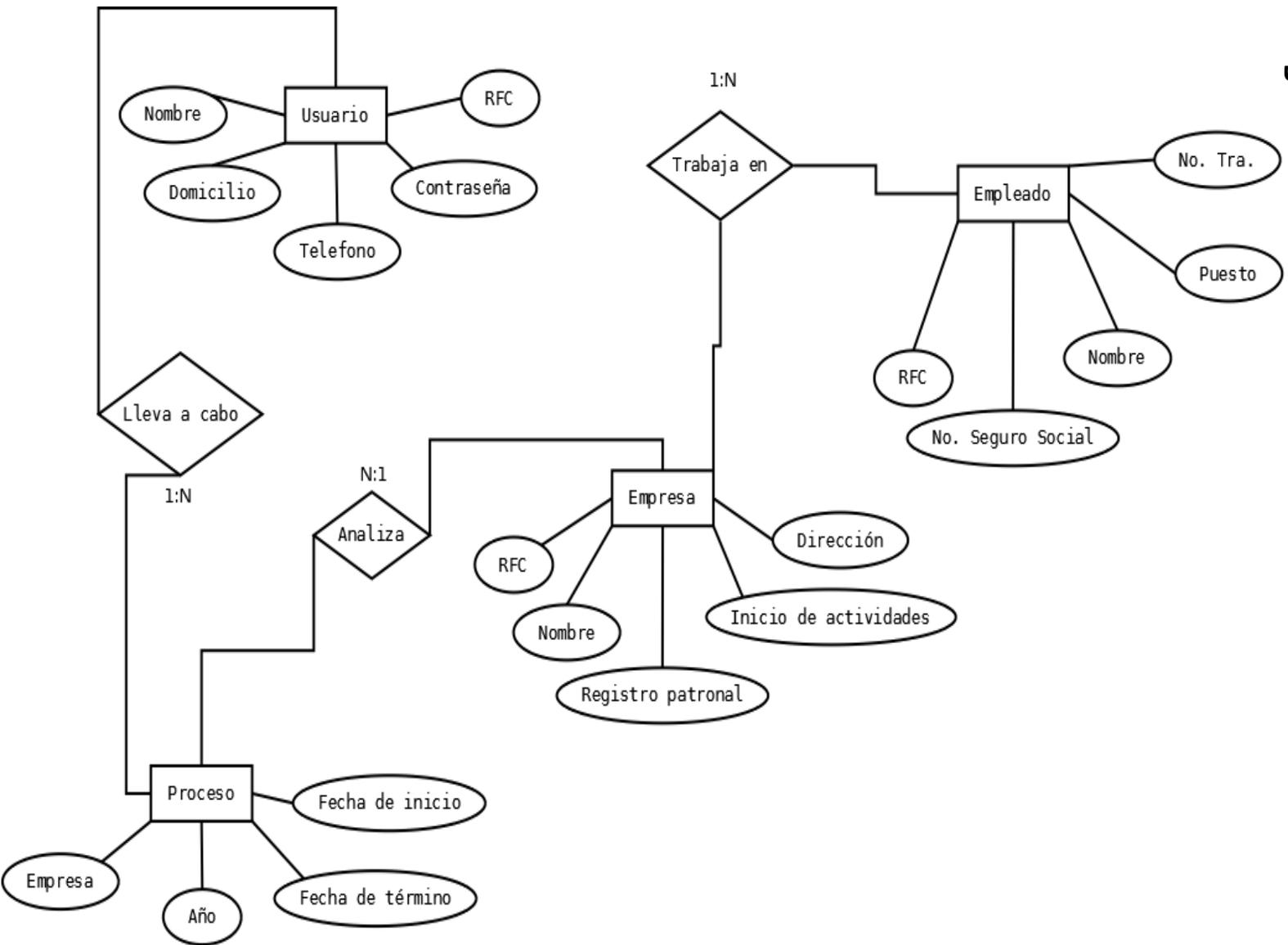
Una muestra detallada de los resultados por mes o por conjunto se da al seleccionar la opción vista general de resultados, donde se muestra de forma explícita cada dato de entrada, los cálculos de totales y las diferencias que se traducen en el importe que debe ser pagado por la empresa al IMSS y/o al INFONAVIT.



Debido al alcance (limitado por el tiempo), aún no se contempla la generación de reportes impresos de estos resultados.

DIAGRAMAS Y DOCUMENTACIÓN TÉCNICA

Diagrama Entidad-Relación



Diccionario de datos
Tabla de usuarios

Número de identificación	U1	U2	U3	U4
Nombre descriptivo	Identificación interna del usuario	Nombre del usuario	Clave o contraseña de acceso	Foto del usuario
Alíás	ID_USUARIO	NOMBRE	CLAVE	FOTO
Campo llave	Si	No	No	No
Descripción general del dato	Consiste en un identificador alfanumérico para cada usuario del Software.	Contiene el nombre de la persona que usara el Software	Contiene la clave personal correspondiente a cada usuario.	Consiste en mostrar la fotografía del usuario.
Longitud	10 caracteres	100 caracteres	15 caracteres	80 megabytes
Tipo de dato	Alfanumérico	Alfanumérico	Alfanumérico	Stream
Origen del flujo de dato	Proviene de una función generadora cada vez que un usuario se da de alta conteniendo su identificador.	Proviene del registro inicial del usuario y contiene su nombre.	Proviene del registro inicial del usuario.	Proviene del registro inicial del usuario.
Destino del flujo de información	El programa solicita la información desde una base datos para iniciar una sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.
Detallar el tipo de dato	Se maneja de manera interna y es un dato de entrada.	Se maneja como dato de entrada, así como dato de salida	Se maneja como dato de entrada.	Se utiliza como dato de entrada y de salida, se mostrará en pantalla.
Control de validación de datos, valores por omisión	A cada usuario le corresponde un identificador correspondiente a los usuarios que se han dado de alta.	Al momento del registro de usuario, se considera como válido el nombre del usuario, puesto que es un empleado del despacho. No hay valores por omisión.	Se valida al momento de dar de alta un usuario. Éste debe contener un mínimo de cuatro caracteres. No hay valores por omisión.	Se valida al momento de dar de alta un usuario, aunque no es obligatoria. No hay valores por omisión.
Área para comentarios generales	La forma como se almacena el dato es la siguiente "U0...0x".	Debe incluir nombre(s) y apellidos escrito todo en mayúsculas.	Todos los usuarios deben contar con una contraseña para la iniciación de sesión.	El formato de la foto debe estar en formato jpg, con un máximo de 0.5 megapíxeles.

Número de identificación	U5	U6	U7	U8
Nombre descriptivo	Domicilio del usuario	Registro Federal de Causantes	Teléfono del usuario	Nivel de administración
Alíás	DOMICILIO	RFC	TELEFONO	ADMON
Campo llave	No	No	No	No
Descripción general del dato	Contiene el domicilio personal del usuario.	Contiene el RFC correspondiente a la persona que usara el Software	Contiene el número telefónico personal correspondiente a cada usuario.	Consiste en un número que identifica el nivel de datos que puede acceder el usuario.
Longitud	120 caracteres	15 caracteres	15 caracteres	32 bits
Tipo de dato	Alfanumérico	Alfanumérico	Alfanumérico	Numérico
Origen del flujo de dato	Proviene el registro inicial del usuario.	Proviene del registro inicial del usuario.	Proviene del registro inicial del usuario.	Proviene del registro inicial del usuario.
Destino del flujo de información	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.
Detallar el tipo de dato	Se maneja como dato de entrada, así como dato de salida	Se maneja como dato de entrada, así como dato de salida	Se maneja como dato de entrada, así como de salida.	Se utiliza de manera interna.
Control de validación de datos, valores por omisión	Se valida al momento de dar de alta un usuario, aunque no es obligatoria. No hay valores por omisión.	Al momento del registro de usuario en su empresa, y se considera como válido. No hay valores por omisión.	Se valida al momento de dar de alta un usuario. Éste debe contener un máximo de quince caracteres. No hay valores por omisión.	El valor por omisión es 3 que consiste en el nivel más bajo y que no tiene acceso a toda la información.
Área para comentarios generales	Debe de consistir máximo de 120 caracteres y no es un dato obligatorio.	Debe ser único dentro de la empresa.	No es un dato obligatorio.	Existen 4 niveles de acceso, en los que el 0 es el más alto y tiene acceso a toda la información y 4 el más bajo y tiene acceso restringido.

Número de identificación	E1	E2	E3	E4
Nombre descriptivo	Identificación interna de la empresa	Registro Federal de Contribuyentes	Nombre de la empresa	Dirección de la empresa
Alíás	ID_EMPRESA	RFC	NOMBRE	DIRECCIÓN
Campo llave	Si	No	No	No
Descripción general del dato	Consiste en un identificador alfanumérico para cada empresa dada de alta.	Contiene el RFC correspondiente a la empresa registrada.	Contiene el nombre de la empresa dada de alta.	Contiene la ubicación física de la empresa.
Longitud	12 caracteres	15 caracteres	100 caracteres	200 caracteres
Tipo de dato	Alfanumérico	Alfanumérico	Alfanumérico	Alfanumérico
Origen del flujo de dato	Proviene de una función generadora cada vez que una empresa se da de alta, conteniendo su identificador.	Proviene del registro inicial de la empresa.	Proviene del registro inicial de la empresa y contiene su nombre.	Proviene el registro inicial de la empresa.
Destino del flujo de información	El programa solicita la información desde una base de datos para iniciar una sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.
Detallar el tipo de dato	Se maneja de manera interna y es un dato de entrada.	Se maneja como dato de entrada, así como dato de salida	Se maneja como dato de entrada, así como dato de salida	Se maneja como dato de entrada, así como dato de salida
Control de validación de datos, valores por omisión	A cada empresa le corresponde un identificador correspondiente a las empresas que se han dado de alta.	Se valida al momento del registro de la empresa. No hay valores por omisión.	Al momento del registro de la empresa, se considera como válido. No hay valores por omisión.	Se valida al momento de dar de alta a una empresa, y si es obligatoria. No hay valores por omisión.
Área para comentarios generales	La forma como se almacena el dato es la siguiente "EO...0x".	Debe ser única según la ley.	Debe incluir el nombre completo de la empresa, escrito en mayúsculas.	Debe de consistir máximo de 200 caracteres y es un dato obligatorio.

Número de identificación	E5	E6	E7	E8
Nombre descriptivo	Registro Patronal	Nombre del Representante Legal	Fecha de iniciación de dictamen	Fecha de iniciación de actividades de la empresa
Alíás	REGISTRO_PATRONAL	REP_LEGAL	INICIO_DICTAMEN	INICIO_ACTIV
Campo llave	No	No	No	No
Descripción general del dato	Consiste en el número de Registro Patronal de la empresa.	Contiene el nombre del representante legal.	Contiene la fecha en que se inicia a dictaminar la empresa.	Contiene la fecha en que se inician las actividades dentro de la empresa.
Longitud	100 caracteres	100 caracteres	32 bits	32 bits
Tipo de dato	Alfanumérico	Alfanumérico	Numérico tipo fecha	Numérico tipo fecha
Origen del flujo de dato	Proviene del registro inicial de la empresa.	Proviene del registro inicial de la empresa.	Proviene del registro inicial de la empresa.	Proviene del registro inicial de la empresa.
Destino del flujo de información	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.
Detallar el tipo de dato	Se maneja de manera interna y es un dato de entrada y de salida.	Se maneja como dato de entrada, así como dato de salida	Se maneja como dato de entrada, y como dato de salida. Funciona de manera interna.	Se maneja como dato de entrada, y como dato de salida. Funciona de manera interna.
Control de validación de datos, valores por omisión	A cada empresa le corresponde un registro patronal único dado legalmente.	Se valida al momento del registro de la empresa. No hay valores por omisión.	Se valida al momento del registro de la empresa. El valor por omisión consiste en la fecha actual.	Se valida al momento del registro de la empresa. No tiene valor por omisión.
Área para comentarios generales	Es un dato obligatorio.	Debe contar con un representante único.	Con esta fecha se calculan los periodos de dictaminación.	Con esta fecha se calculan los factores de integración por empleado.

TABLA DE PROCESOS

Número de identificación	P1	P2	P3
Nombre descriptivo	Identificador del proceso	Empresa a la que corresponde el proceso.	Año al que corresponde el proceso
Alíás	ID_PROCESO	EMPRESA	ANUAL
Campo llave	Si	No	No
Descripción general del dato	Consiste en el número identificador del proceso.	Contiene el ID interno de la empresa de la que se lleva a cabo el proceso	Contiene el año en que se lleva a cabo el proceso de cálculo de diferencias.
Longitud	12 caracteres	12 caracteres	32 bits
Tipo de dato	Alfanumérico	Alfanumérico	Númérico
Origen del flujo de dato	Proviene del registro de un nuevo proceso.	Proviene del registro inicial de la empresa, debe existir la empresa.	Proviene del registro inicial del proceso.
Destino del flujo de información	El programa solicita la información desde una base de datos para iniciar un proceso o continuarlo.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.
Detallar el tipo de dato	Se maneja de manera interna y es un dato de entrada y de salida.	Se maneja como dato de entrada, así como dato de salida.	Es un dato de entrada.
Control de validación de datos, valores por omisión	A cada empresa le corresponde uno o varios procesos, de acuerdo al año de dictamen.	Es válido, sólo si el ID de la empresa existe. No hay valores por omisión.	Por omisión se toma el año actual, no debe ser mayor a este.
Área para comentarios generales	La forma como se almacena el dato es la siguiente "PO...0x".	El dato debe pertenecer al dominio de empresas dadas de alta.	Con esta fecha se termina el período anual de dictaminación.

TABLA DE PROCESOS, CONTINUACIÓN

Número de identificación	P4	P5
Nombre descriptivo	Fecha de inicio del proceso	Fecha en la que termina el proceso
Alíás	INICIO_PROCESO	FIN_PROCESO
Campo llave	No	No
Descripción general del dato	Contiene la fecha en que se inician las actividades cálculo de diferencias.	Contiene la fecha en que se terminan las actividades cálculo de diferencias.
Longitud	32 bits	100 caracteres
Tipo de dato	Numérico tipo fecha	Alfanumérico
Origen del flujo de dato	Proviene de fecha en la que se registró un proceso.	Proviene de fecha en la que se termina el proceso.
Destino del flujo de información	Se consulta continuamente para determinar los cambios con respecto a la fecha.	Se almacena en la base de datos y es consultado cada referencias futuras.
Detallar el tipo de dato	Se maneja como dato de entrada.	Es un dato de entrada y de salida.
Control de validación de datos, valores por omisión	Se valida al momento del registro de la empresa. La fecha por omisión es la actual (registro).	Es la fecha en la que se determina el fin del proceso.
Área para comentarios generales	Permite llevar un control sobre la nómina.	

TABLA DE SESIONES

Número de identificación	S1	S2	S3	S4
Nombre descriptivo	Identificador de sesión	Usuario que inicia sesión	Proceso que corresponde a la sesión.	Permisos de un usuario de acceso a la información.
Alías	ID_SESION	USUARIO	NPROCESO	NIVEL_ACCESO
Campo llave	Si	No	No	No
Descripción general del dato	Identifica un proceso asociado a un usuario.	Contiene el id del usuario que usará su sesión para abrir el proceso.	Contiene del proceso.	Contiene un nivel de acceso.
Longitud	12 caracteres	12 caracteres	12 caracteres	32 bits
Tipo de dato	Alfanumérico	Alfanumérico	Alfanumérico	Numérico
Origen del flujo de dato	Proviene del registro inicial del proceso.	Proviene del registro del proceso.	Proviene del registro del proceso.	Proviene del registro inicial del proceso.
Destino del flujo de información	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.	Se almacena en la base de datos y es consultado cada vez que se inicia la sesión.
Detallar el tipo de dato	Se maneja de manera interna y es un dato de entrada.	Se maneja como dato de entrada.	Se maneja como dato de entrada. Es un único proceso por sesión.	Se maneja como dato de entrada, y como dato de salida. Determina a los usuarios específicos por sesión.
Control de validación de datos, valores por omisión	A cada sesión le corresponde un proceso y un usuario con permisos para acceder a él.	Se valida al momento del registro del proceso, pues está en el dominio de usuarios.	Se valida al momento del registro del proceso. Se basa en el dominio de procesos.	Numero entero, entre 0 y 4.
Área para comentarios generales	Es un dato obligatorio.	En un proceso puede haber más de un usuario.		Permite la consulta de información por otros usuarios de un proceso.

IMPLEMENTACIÓN Y VALIDACIÓN

Estrategias y planes de pruebas del software:

Las pruebas inmediatas serán realizadas cubriendo todos los aspectos establecidos en el alcance del proyecto, utilizando datos proporcionados por el Despacho.

En primer lugar se analizará la exactitud con la que el software presenta la información, con el fin de evaluar la capacidad de cómputo. Para ello se usará la prueba estructural, de ejecución y de carga, antes de implementar el software.

Después se probará la tolerancia a fallas del software, con el fin de verificar la capacidad del software para mantener un nivel de rendimiento específico ante un error. Para ello se efectuarán las siguientes pruebas:

Prueba de Valores Límites: Evaluar valores frontera; es decir, los valores mínimos y máximos que la unidad puede aceptar.

Prueba Bajo Stress: Evaluar la capacidad del sistema para seguir operando apropiadamente ante una baja disponibilidad de recursos.

Prueba de Volumen: Someter al software a una gran cantidad de datos para determinar si los límites alcanzados hacen que este falle.

Estrategias y planes de capacitación

Como se estableció en el Análisis de Factibilidad, la capacitación a los usuarios se realizará en un máximo de 15 horas en las instalaciones del cliente. Se pretende que el Despacho Contable Jiménez Márquez, reciba un curso de capacitación formal con el fin de que todos los usuarios desarrollen sus habilidades, combinando la teoría y la práctica del software “Estimación de diferencias empresariales”.

Estrategias y planes de conversión

Posterior a la capacitación, durante la semana de pruebas de uso, se iniciará la migración de los datos de la aplicación anterior al sistema SEDE

Así mismo se analizarán las computadoras en las que será usado el software para detectar posibles configuraciones o estados del sistema operativo que pueden afectar el rendimiento del software para que este no sea obstáculo a la hora de migrar a este nuevo sistema.

En esta fase, no se dispondrá de todos los usuarios ni de todos los equipos, solo algunos participarán en esta fase, para reducir lo más posible el retraso en operaciones de la empresa.

Estrategias y planes de mantenimiento

El mantenimiento directo abordará aquellos aspectos que estén dentro del alcance del software, y aquellos aspectos operativos que afectan directamente su funcionamiento.

El alcance del software “en su etapa inicial” limita ciertos aspectos que quizá eventualmente sean requeridos para su pleno uso. En este sentido los planes de desarrollo para esos aspectos se programarán para fechas a largo plazo.

Calendarización propuesta

La calendarización siguiente, se basa en el tiempo en el que la carga de operaciones de la empresa disminuye, por lo tanto se tiene mayor disponibilidad para la atención del proyecto.

Estrategia/fecha	9-16 Nov.	23 o 24 Nov.	14-18 Dic.
Pruebas	■		
Capacitación		■	
Conversión		■	■
Mantenimiento		■	■

Manual del usuario

Sistema y Estimaciones
de Diferencias
Empresariales

Fernando Hernández Martínez

CONTENIDO

INTRODUCCIÓN	3,8
INSTALACIÓN.....	4
PRESENTACIÓN DE LAS INTERFACES.....	9
AGREGAR UN USUARIO E INICIAR SESIÓN	9
AGREGAR UNA EMPRESA Y CONTINUAR UN PROCESO.....	9
INTRODUCIR INFORMACIÓN DE NÓMINA	12
FORMATO DEL ARCHIVO DE NÓMINA.....	11
PRESENTACIÓN FILTRADA DE LA INFORMACIÓN	13
EL USO DE LA LISTA DE EMPLEADOS Y LA LISTA DE MESES	15
CONCLUSIÓN.....	..20

INTRODUCCIÓN

Antecedentes

EL sistema de seguridad ha tomado una parte importante en el desarrollo de México. El Instituto de Seguridad Social es uno de los pilares que sustentan la seguridad social y es uno de los más grandes e importantes en Latinoamérica. Desde sus 66 años de haberse establecido se ha encargado de proporcionarle a la población mexicana atenciones médica, así como con un fondo para el retiro, guarderías, etc. Las empresas aportan periódicamente unas cuotas por trabajador que permiten al IMSS cumplir con sus objetivos. Estas empresas deben cumplir con el pago de sus cuotas obligatoriamente, pero estas cuotas que se pagan al IMSS a veces resultan tener costos igual o más importantes que el pago de algunos impuestos, por lo que se les debería de prestar más atención a los pagos de seguridad social de la que actualmente tienen.

Necesidades

Las empresas cuentan internamente con áreas dedicadas a la contabilidad, a los impuestos, a los recursos humanos, pero la parte relacionada con la seguridad social pasa a ser un área de segundo o tercer nivel que depende de otras áreas. Resultado de lo anterior, las personas que se dedican a preparar los pagos correspondientes para el IMSS, por una parte no tienen los estudios necesarios, y por otra parte tampoco cuentan con las herramientas (computacionales) necesarias para desarrollar de una manera más eficiente su trabajo. Por lo cual muy frecuentemente se cometen muchos errores que inciden directamente en las utilidades de las empresas, y estos gastos contribuyen a hacer más ineficientes a nuestras empresas mexicanas.

Perspectivas y Ventajas

El presente trabajo fue desarrollado pensando en proporcionar una herramienta informática que permitiera a las empresas eficientar el proceso del cálculo de pagos que se efectúan al IMSS por concepto de cuotas obrero patronales, tal que permita filtrar la información antes de que las unidades administrativas del IMSS detecten fallos en el entero de cuotas ocasionado un cobro por concepto de actualizaciones y recargos.

El IMSS ha delegado en despachos privados la facultad de efectuar revisiones y auditorias a las empresas privadas, es por ello que esta herramienta sirve para facilitar y eficientar el trabajo de los despachos que se encargan de revisar la formulación e integración de las cuotas que se pagan al IMSS.

Este trabajo se ha orientado en su primera fase al cálculo de las diferencias exclusivamente para el IMSS, pero puede ser de gran utilidad también para el INFONAVIT ya que utilizan ambos institutos el mismo sistema de procesamiento de cuotas.

Las *ventajas* que presenta esta herramienta se pueden resumir en lo siguiente:

- Ayuda a sistematizar el ordenamiento de la información que interviene en el cálculo de las diferencias.
- Formula procesos matemáticos que reducen la posibilidad de errores que tiene el personal que procesa la información.
- El sistema mejora sustancialmente la eficiencia y rapidez en la formulación del trabajo.
- Ayuda a que las empresas no incurran en gastos innecesarios por concepto de actualizaciones y recargos.
- Al reducirse los errores en los pagos, evita que las autoridades del IMSS manden una revisión especial a la empresa, revisiones que frecuentemente terminan en embargos de bienes, o en congelación de cuentas bancarias.

Propósito

Este manual tiene el propósito de explicar la forma de utilizar el sistema para que no presente problemas al momento de su uso. Le permite instalar, utilizar y mantener el software de forma sencilla.

INSTALACIÓN

A continuación se explicará el proceso de instalación del software SEDE. Antes de proceder, se presentan los requisitos de hardware y software que deben considerarse para instalar el software.

Hardware	Software
Procesador Intel o AMD a 800 MHz o superior	Windows 2000 sp 6, o superior
256 Mb de memoria (1GB en Windows vista, debido a los requerimientos de este)	Firebird 2.1 o superior (Incluido en la instalación)
20 MB de espacio en disco duro (Para instalación)	MS OFFICE (manual en formato word) ó Adobe reader(Manual en format pdf)
Pantalla con resolución de 1024*768 pixeles, o mayor (Para facilitar la navegación)	
Teclado y mouse.	
Unidad lectora de CD	

Proceso de instalación:

1. Inserte el CD de instalación en la unidad de cd, aparecerá una ventana como la siguiente:



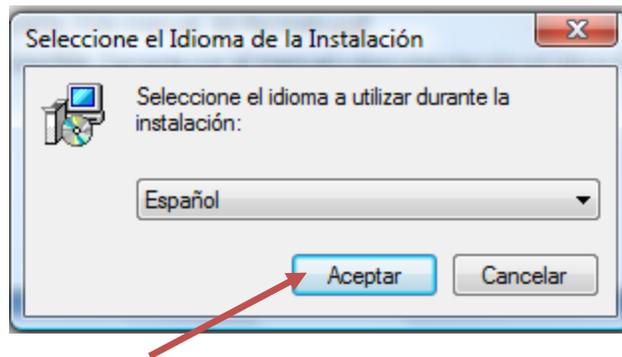
Se mostrarán tres opciones:

- Instalar: Inicia el asistente de instalación

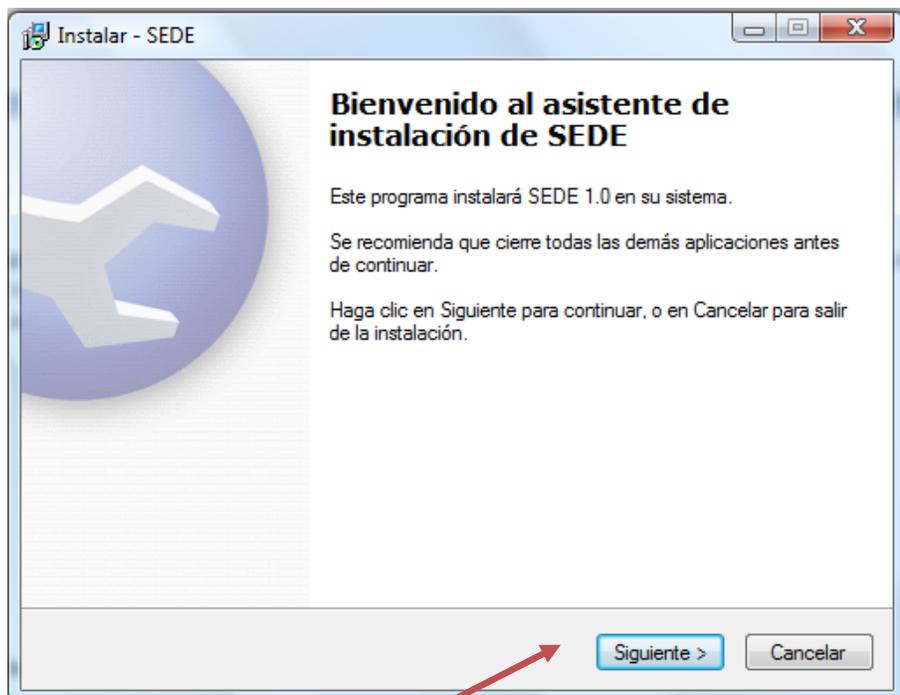
- Manual de usuario: Este manual, en formato pdf
- Carpeta documentos: Carpeta con el manual y documentación en otros formatos.

2. Haga clic en Instalar, Se iniciará el asistente de instalación, bastante similar a la instalación de otros programas en Windows. (Nota: En Windows Vista, debe iniciar el proceso como Administrador)

En la primera ventana, seleccione el Idioma (español) y haga clic en aceptar.



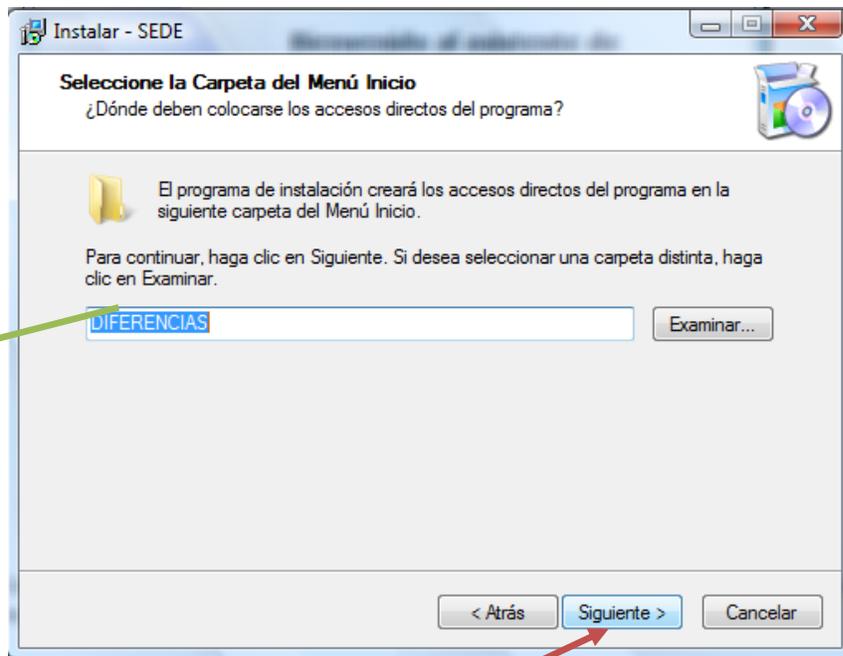
Esta es la pantalla de bienvenida del asistente de instalación, de clic en siguiente.



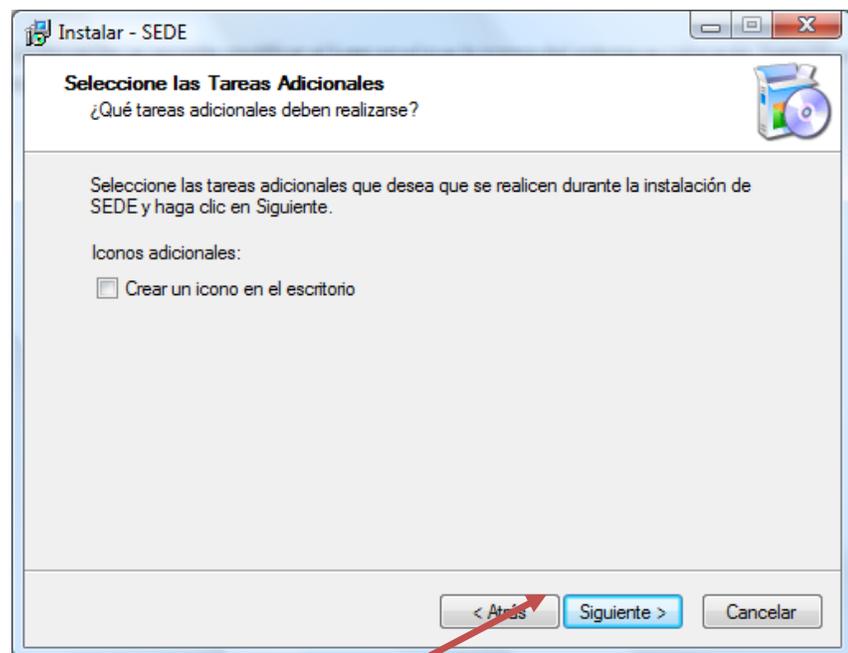
A

Continuación se permite modificar el lugar en el que los iconos del sistema se colocarán, haga clic en siguiente, o modifique si lo desea la ubicación.

Seleccionar ubicación

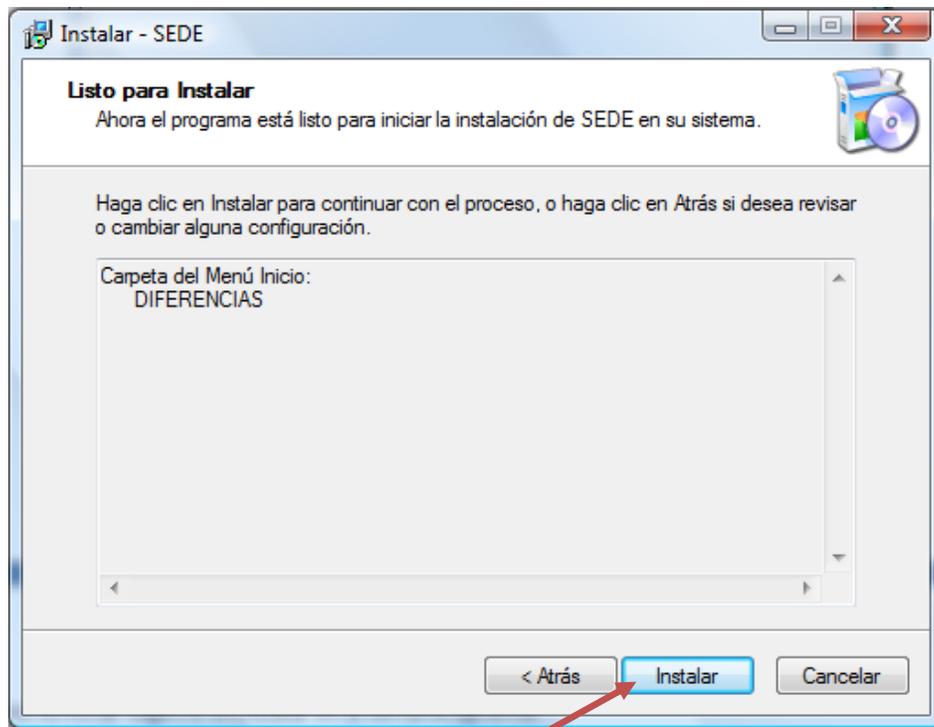


Active la casilla “Crear icono en el escritorio”, si desea que aparezca un acceso directo en el escritorio.



Para Terminar en Instalar en la ventana siguiente.

haga clic



A continuación se copiarán los archivos necesarios, y se configurará el gestor de base de datos. Si lo desea, al terminar puede ejecutar el software.

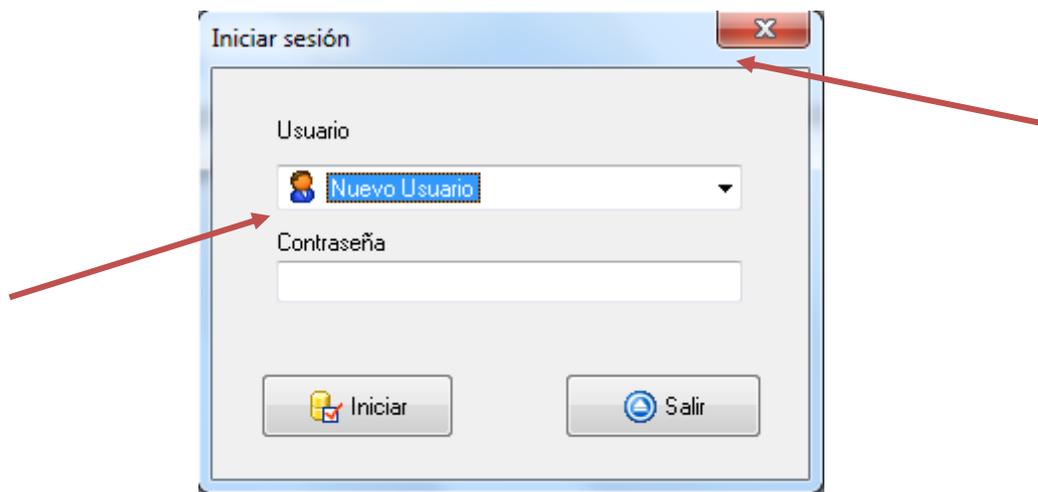


PRESENTACIÓN DE LAS INTERFACES

A continuación se mostraran las interfaces con las que se va a estar trabajando una vez realizada la instalación del software.

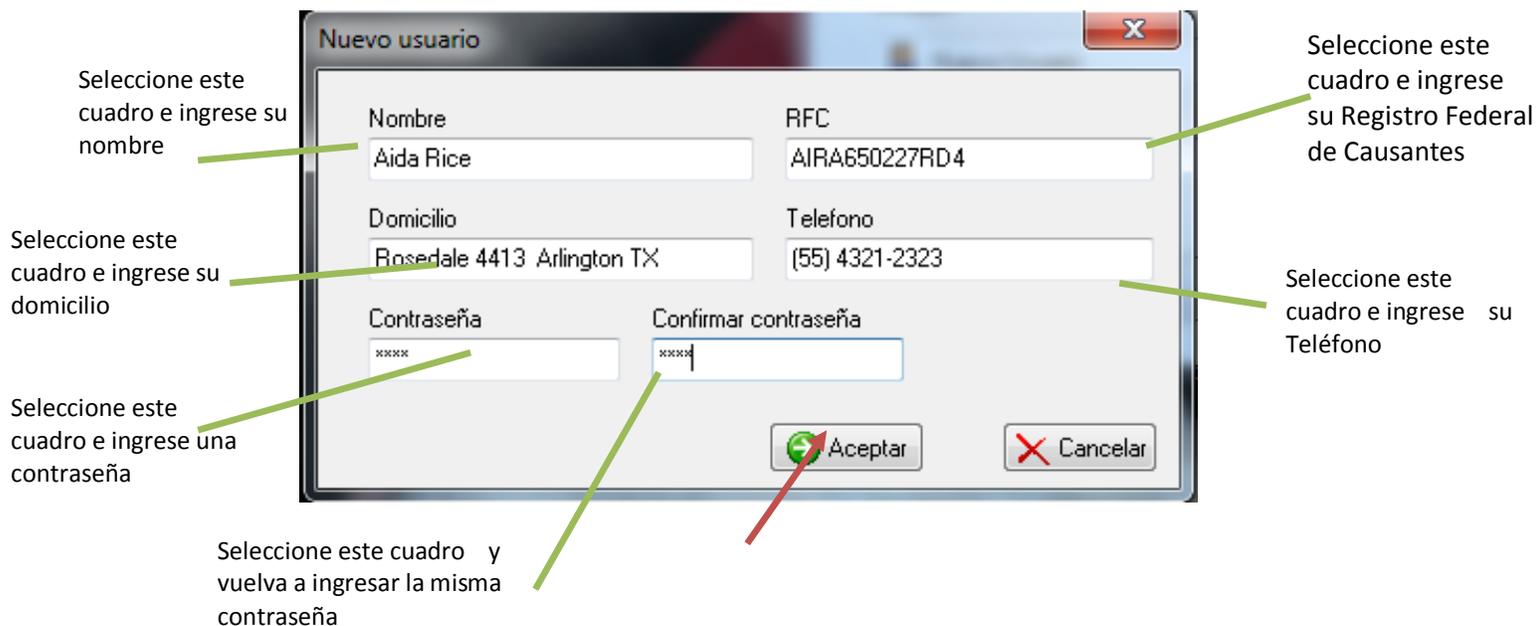
AGREGAR UN USUARIO E INICIAR SESIÓN

Al iniciar SEDE, aparece el cuadro de dialogo de inicio de sesión en el que se presenta un *Usuario* y *Contraseña*. En el cuadro correspondiente al Usuario dé un clic en la flecha de despliegue y seleccione “Nuevo Usuario”. Seleccione “Iniciar”.



Al momento de registrar un Nuevo Usuario y dar clic en “Iniciar” aparece el siguiente cuadro de dialogo en donde se va a dar de alta un Nuevo Usuario. Seleccione el cuadro y escriba el dato Correspondiente. Al finalizar seleccione “Aceptar”.

Se utilizaron datos ficticios para ejemplificar:



Después de dar de alta al Usuario y Contraseña, en el cuadro correspondiente al Usuario, escriba su nombre, o un nombre de inicio. En el cuadro correspondiente a la Contraseña, escriba una contraseña de inicio de sesión. Elija “Iniciar”.

Seleccione este cuadro y después escriba su contraseña

Iniciar sesión

Usuario
Aida Rice

Contraseña
xxxxxx

Iniciar Salir

Seleccione este cuadro y escoja su nombre que previamente fue dado de alta

AGREGAR UNA EMPRESA Y CONTINUAR UN PROCESO

Se presentará un cuadro de dialogo correspondiente a la Selección de Empresas. Para dar de alta una nueva empresa, se debe dar clic en el botón de “Nueva empresa”.

Seleccionar empresa

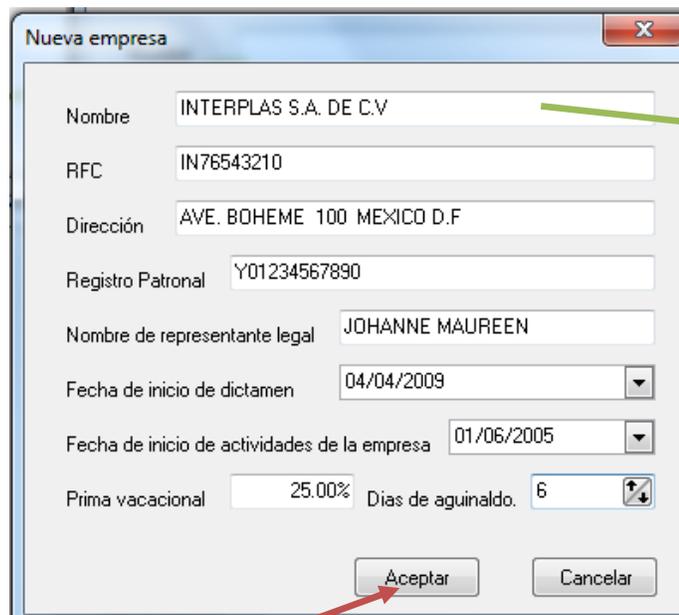
Elija la empresa de la que desea consultar o continuar un proceso

Empresa	Permisos	Año
---------	----------	-----

Nueva empresa

Aceptar Cancelar

El cuadro de dialogo que surgirá corresponde a dar de alta a la empresa. Se deben seleccionar las casillas y llenar con el dato correspondiente. Al finalizar dar clic en “Aceptar”.



The image shows a Windows-style dialog box titled "Nueva empresa". It contains several input fields and two buttons at the bottom. A green arrow points to the "Nombre" field, and a red arrow points to the "Aceptar" button.

Nombre	INTERPLAS S.A. DE C.V
RFC	IN76543210
Dirección	AVE. BOHEME 100 MEXICO D.F
Registro Patronal	Y01234567890
Nombre de representante legal	JOHANNE MAUREEN
Fecha de inicio de dictamen	04/04/2009
Fecha de inicio de actividades de la empresa	01/06/2005
Prima vacacional	25.00%
Dias de aguinaldo	6

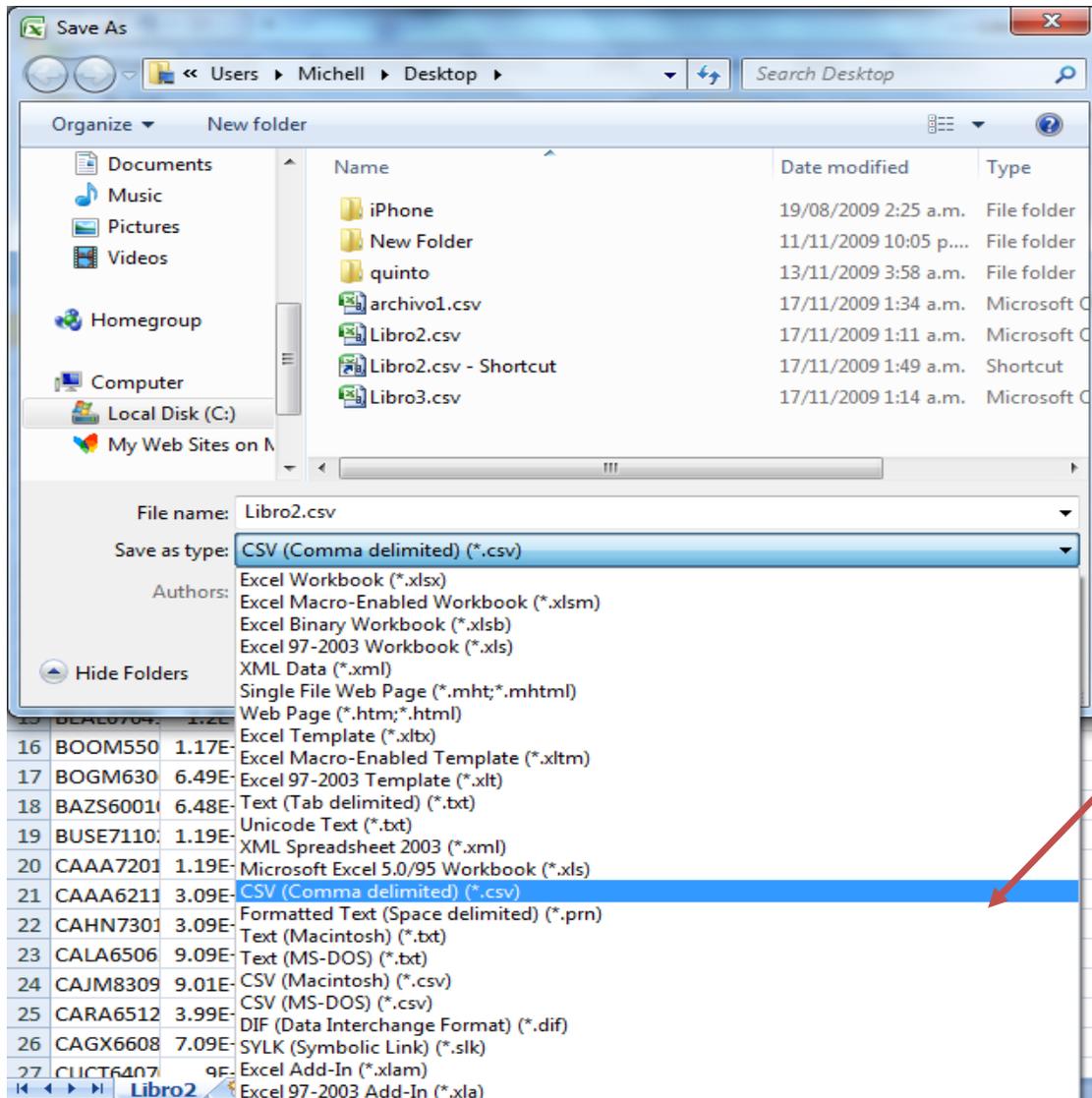
Seleccione este cuadro y escriba el nombre de la empresa

En el cuadro de dialogo siguiente se debe seleccionar el archivo en donde se encuentra la nómina de la empresa, y adicionalmente se selecciona el archivo en donde se encuentran las cuotas, aportaciones y amortizaciones.

FORMATO DEL ARCHIVO DE NÓMINA

Para poder ingresar la información que está contenida en una nómina por cada empresa, es necesario asegurar un formato que acepte el SEDE. Este formato es *csv* (comma separated values/ comma delimited) o valores separados por comas.

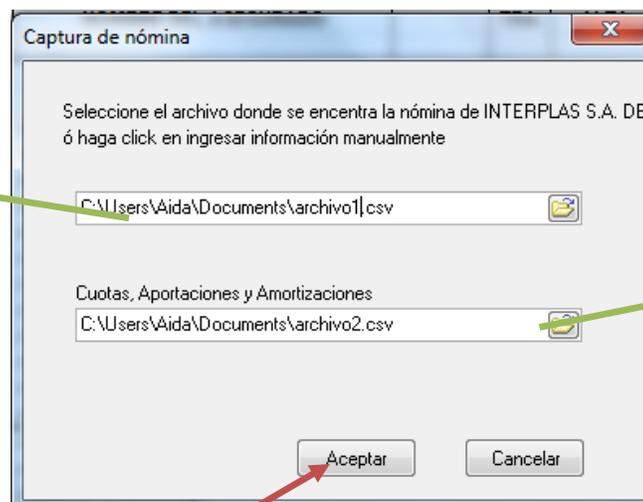
Note que normalmente los archivos que se reciben de las empresas vendrán en este formato, pero en caso de que no sea así, al momento de guardar el archivo se debe seleccionar el formato especificado *csv*.



INTRODUCIR INFORMACIÓN DE NÓMINA

Para introducir información se seleccionan los archivos correspondientes con el formato csv.

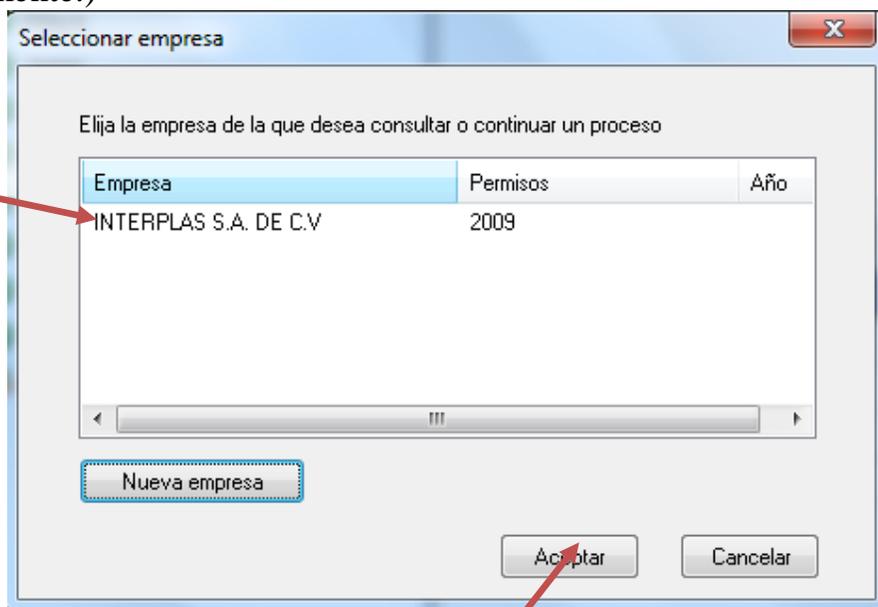
Seleccione el archivo en donde se encuentra la nómina de la empresa



Seleccione el archivo en donde se encuentran las cuotas, aportaciones, y amortizaciones

En el cuadro de dialogo emergente se muestran las empresas que han sido dadas de alta. Seleccionar la empresa con la que va a trabajar, y de clic en “Aceptar”.

(Note que para agregar una nueva empresa se debe seleccionar “Nueva empresa” como se hizo anteriormente.)



PRESENTACIÓN FILTRADA DE LA INFORMACIÓN

Cuando el usuario ingrese los archivos con la información correspondiente, se acomodara la información como se muestra en la siguiente imagen, en donde existen dos tablas de la integración de todos los datos. La parte superior en la presentación de los datos está conformada por los datos más comúnmente referenciados para el análisis de la información. La tabla de la parte inferior, está compuesta por todos los datos disponibles, así como los cálculos que fueron realizados para desplegar algunos campos de información. Para cada una de estas tablas existen sus barras desplegables para el acceso al resto de la información que no está visible en el espacio actual.

En la esquina superior izquierda de la pantalla se encuentra el botón para “almacenar los cambios realizados”. En esta misma esquina también se encuentra una opción para la “captura de datos del mes” en donde se puede seleccionar, de tal manera que conforme vaya avanzando el año se puedan ir agregando los datos del mes siguiente.

La búsqueda rápida tiene la función de localizar a un trabajador mediante su número de Seguro Social, R.F.C, nombre, o su puesto. En el campo de búsqueda se escribe el numero o nombre, y en el campo del criterio se selecciona el criterio (número de Seguro Social, R.F.C, etc.).

En la esquina superior derecha se encuentra la opción para “cerrar el proceso” lo que llevará al cierre del programa.

Guardar los cambios realizados

Agrega archivos con nueva información

Realiza búsquedas conforme a un criterio

Barra de despliegue

Cierra el proceso

MESES	AÑOS	FACTOR DE INTEGRACION	SUELDO MENSUAL	SUELDO DIARIO	PERCEPCIONES
7	1.0918		\$12,483.90	\$416.13	\$4,502.54
3	1.0890		\$6,813.00	\$227.10	\$19,243.34
9	1.0918		\$15,843.90	\$528.13	\$14,083.74
9	1.0918		\$14,628.00	\$487.60	\$14,144.14
5	1.0918		\$13,412.10	\$447.07	\$5,223.14
8	1.0918		\$13,020.90	\$434.03	\$8,005.34
7	1.0918		\$11,310.90	\$377.03	\$7,798.90
8	1.0918		\$10,593.90	\$353.13	\$1,520.14
8	1.0918		\$29,191.44	\$973.05	\$1,690.14
9	1.0918		\$11,558.10	\$385.27	\$3,228.64
4	1.0904		\$7,554.00	\$251.80	\$3,050.14

Resumen de diferencias	AÑOS	FACTOR DE INTEGRACION	SUELDO MENSUAL	SUELDO DIARIO	PERCEPCIONES	IVCV (DEBIÓ COZAR)	IVCV (COTIZÓ)	IVCV (DIFERENCIA)
7	1.0918		\$12,483.90	\$416.13	\$4,502.54	\$528.13	\$530.56	-\$2.43
3	1.0890		\$6,813.00	\$227.10	\$19,243.34	\$562.79	\$563.18	-\$0.39
9	1.0918		\$15,843.90	\$528.13	\$14,083.74	\$807.48	\$812.46	-\$4.98
9	1.0918		\$14,628.00	\$487.60	\$14,144.14	\$764.22	\$771.19	-\$6.97
5	1.0918		\$13,412.10	\$447.07	\$5,223.14	\$573.73	\$573.52	\$0.21
8	1.0918		\$13,020.90	\$434.03	\$8,005.34	\$605.10	\$606.11	-\$1.01
7	1.0918		\$11,310.90	\$377.03	\$7,798.90	\$539.48	\$540.50	-\$1.02
8	1.0918		\$10,593.90	\$353.13	\$1,520.14	\$410.46	\$411.47	-\$1.01
8	1.0918		\$29,191.44	\$973.05	\$1,690.14	\$1,090.06	\$1,006.65	\$83.41
0	1.0910		\$11,558.10	\$385.27	\$2,228.64	\$472.56	\$474.57	-\$1.01

Tabla de información selecta

Tabla de información completa

EL USO DE LA LISTA DE EMPLEADOS Y LA LISTA DE MESES

La presentación de la información inicial se muestra en relación a los meses correspondientes así como al año como se mostro anteriormente. De manera similar, existe una lista de empleados en donde también se muestran las tablas de la parte derecha, pero a diferencia de la presentación por meses, esta muestra la información de los empleados que le corresponden a la empresa. Esto es, muestra su nombre, número de Seguro Social, R.F.C., puesto, fecha de alta, años trabajados, y la cuota que recibe diariamente. Al seleccionar un nombre desde la lista, se mostrará su información en la parte inferior, y se destacará en color azul la información de este trabajador en las tablas de la parte derecha.

Tabla de información selecta

Tabla de información completa

The screenshot shows the SEDE software interface with the following components:

- Empleado List:** A table listing employees with columns for RFC, Nombre, and No. Seguro Social. The employee **ASPANI ALFIE RACHEL** (RFC: AAAR611019BD5) is highlighted in blue.
- Employee Details:** A panel on the left showing details for the selected employee, including RFC, No. Seguro social, Puesto (Profesor), Fecha de alta (01/09/2004), Años trabajados (5), and Cuota Diaria (\$447.07).
- Payroll Tables:** Two tables on the right showing payroll data. The top table, 'Sección Nómina de la empresa', has columns: AÑOS, FACTOR DE INTEGRACION, SUELDO MENSUAL, SUELDO DIARIO, and PERCEPCIONES. The bottom table, 'Resumen de diferencias', has columns: AÑOS, FACTOR DE INTEGRACION, SUELDO MENSUAL, SUELDO DIARIO, PERCEPCIONES, IVCV (DEBIO COTIZAR), IVCV (COTIZÓ), and IVCV (DIFE). The row for the selected employee is highlighted in blue.

Lista de Empleados
Lista de Meses

Información de Empleados

Lista de todos los empleados de la empresa

Selección azul correspondiente al empleado elegido

CONCLUSIÓN

En primera instancia, este software pretende servir como herramienta para realizar diversos cálculos en relación al Instituto Mexicano de Seguridad Social que se requieren en un despacho de contabilidad; sin embargo, dadas las necesidades de diversas empresas por realizar los pagos adecuados al Instituto, se puede pensar en un mayor alcance.

Este software podría llegar a adecuarse para su utilización dentro de diversas empresas, para que las personas encargadas de los movimientos en las cuotas de los trabajadores puedan realizar el proceso sin complicaciones. Los mismos administradores podrían ingresar la información respectiva de su empresa, actualizando los datos mensualmente, para así asegurar que lo pagado al IMSS sea lo correspondiente, y no se tengan que arriesgar a pagar intereses o atrasos de cantidades importantes. Este alcance podría sonar ambicioso, sin embargo la necesidad en el mundo empresarial existe, y también existen las herramientas para poder realizar un software que le ahorre a las empresas una cantidad importante de dinero, que simplemente sería producto de la implementación de un buen software.