



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE
INGENIERÍA**

**SISTEMA PARA LA
AUTOMATIZACIÓN DE FUERZA DE
VENTAS REMOTA**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN:**

PRESENTAN:

**MARCO ANTONIO ALFARO FLORES
REBECA ALEJANDRA SÁNCHEZ LABASTIDA
JOSÉ ANTONIO VELÁZQUEZ VIDRIO**

DIRECTOR: AURELIO ADOLFO MILLÁN NÁJERA

CIUDAD UNIVERSITARIA

JUNIO, 2007



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.

Agradecimientos

Este trabajo representa el fruto del esfuerzo y dedicación de mis padres, a quienes agradezco su apoyo para el logro de este objetivo, así como a mi esposa e hijas quienes motivaron la culminación de este esfuerzo. Agradezco también a mi profesor de tesis por su apoyo y seguimiento.

Marco Antonio

A Carmen Fernández por sus cuidados y todo el cariño que siempre me ha dado.

A Carmen Vidrio por su apoyo, motivación y por siempre ayudarme a ser mejor.

A mi familia y amigos que siempre han tenido confianza y han creído en mí.

A Rebeca Sánchez Labastida por su apoyo, motivación, confianza y por siempre estar ahí cuando te he necesitado.

A la Familia Sánchez por su apoyo y amistad en todo momento.

A mis maestros y director de tesis por sus enseñanzas, paciencia y apoyo.

José Antonio

Agradezco mis padres el cariño, esfuerzo y apoyo que me han dado en todo momento.

A mi hermano, Gina, Yunuen y tías por estar siempre conmigo.

A José Antonio por siempre estar a mí lado y ayudarme a crecer en este camino.

A mis amigos que me han brindado sus consejos.

A mi familia por su apoyo incondicional.

Gracias a nuestro director de tesis por su apoyo y ayuda para esta tesis.

Rebeca

Índice

Prólogo

Capítulo I. Antecedentes	1
Objetivo del capítulo:.....	1
I.1 Panorama general del sector manufactura en México	2
I.2 Modelo de comercialización de productos.....	3
I.3 Situación del Mercado.....	5
I.4 Descripción general de los sistemas de automatización de la fuerza de ventas	11
I.5 Respuesta del sistema propuesto a la problemática del sector	13
Capítulo II. Metodología de desarrollo	15
Objetivo del capítulo:.....	16
II.1 Problemática	16
II.2 MSF	17
II.2.1 Modelo de procesos de MSF (Process Model).....	18
II.2.1.1 Fase de planeación	19
II.2.1.2 Fase de análisis	21
II.2.2.3 Fase de diseño.....	23
II.2.2.4 Fase de desarrollo.....	28
II.2.2.5 Fase de aseguramiento de calidad (Estabilización).....	30
II.3 UML (Unified Modeling Language).....	34
Capítulo III. Planeación y análisis	38
Objetivo del capítulo:.....	39
III.1 Planeación	39
III.1.1 Estructura del equipo de trabajo.....	40
III.1.2 Visión del Proyecto.....	41
III.1.3 Evaluación de Riesgos	44
III.2 Análisis.....	46
III.2.1 Definición de Requerimientos.....	46
III.2.2 Arquitectura de Desarrollo.....	49
Capítulo IV. Diseño y desarrollo	59
Objetivo del capítulo:.....	60
IV.1 Fase de diseño	60
IV.1.1 Diseño conceptual	60
IV.1.2 Diseño lógico	61
IV.1.3 Diseño físico	64
IV.2.1 Fase de desarrollo	65
IV.2.1.1 Capa de datos	66
IV.2.1.2 Capa de Negocios	67
IV.2.1.3 Capa de presentación	68
Capítulo V. Aseguramiento de la calidad	71
Objetivo del capítulo:.....	72
V.1 Detección y corrección de fallas	72
V.2 Pruebas finales y liberación de versión 1.0	75
V. 3 Elaboración de manuales	77
Capítulo VI. Implantación y liberación a producción.....	81
Objetivo del capítulo.....	82
VI.1 Planeación:.....	82
VI. 2 Estructura del equipo de trabajo de implantación.....	85
Conclusiones Generales	87

Prólogo

El presente trabajo ilustra el proceso para el desarrollo de un sistema informático, tomando como ejemplo un “Sistema para la Automatización de Fuerza de Ventas Remota”, sustentado en la metodología empleada y descrita en los capítulos de esta tesis.

Se describe el proceso realizado para la detección de necesidades, análisis, diseño y desarrollo del sistema ejemplo de esta tesis, el que resolvería las necesidades detectadas en el sector manufactura con el objeto de construir una herramienta que apoye la comercialización y distribución de sus productos. Proveer también información oportuna y actualizada sobre sus clientes para la mejor toma de decisiones.

En el sector de manufactura cuyo mercado meta es el de autoservicios, mayoristas y distribuidores, se han detectado una serie de problemas relativos a la actividad comercial que desempeña la fuerza de ventas de campo, mismos que se describen a continuación:

- Procesos de venta largos
- Medio ineficiente para la consulta de información histórica
- Inadecuada planeación de las estrategias de ventas
- Falta de control en las actividades de los ejecutivos de ventas
- Pérdidas económicas por errores en la generación de pedidos duplicados.
- Producción insuficiente para el surtido de pedidos por una ineficiente proyección de ventas.
- Falta de un medio de comunicación eficaz para el intercambio de información entre los diferentes niveles del área de ventas.
- Pérdida de información relativa al negocio por falta de una base de datos unificada y centralizada.

Se propone la solución a los problemas anteriormente citados, a través del sistema ejemplo, que plantea automatizar los procesos de venta, para:

- Mejorar la atención a clientes
- Disminuir los costos de operación
- Incrementar el volumen de ventas
- Incrementar la participación en el mercado
- Reducir los ciclos de venta
- Disminuir las pérdidas en la distribución del producto, por entregas erróneas y pedidos duplicados.

El contenido de cada capítulo se define a continuación:

En el capítulo I se dará el panorama general del sector manufactura en México, se hablará de su modelo de comercialización, de su situación en el mercado, de los sistemas de automatización como estándar de la industria y de la problemática del sector que se propone resolver con esta tesis.

En el capítulo II se hablará de la metodología que se empleó a lo largo del proyecto, se describirá cada una de las fases que la componen, así como de las herramientas adicionales que se emplearon con esta metodología.

En el capítulo III se describirá el proceso de planeación y análisis seguido para la construcción del sistema, empleando como guía la metodología seleccionada y descrita en el capítulo anterior.

En el capítulo IV se describirán las actividades realizadas en la fase de diseño y desarrollo, que forman parte de la metodología empleada para el desarrollo del sistema, así como de las herramientas que sirvieron de apoyo.

En el capítulo V se mencionarán las actividades de aseguramiento de la calidad seguidas en el proyecto, así como de los procesos de documentación para usuarios y administradores del sistema.

En el capítulo VI se mencionarán las actividades realizadas para la implantación y liberación a producción del sistema desarrollado en esta tesis.

Capítulo I. Antecedentes

Objetivo del capítulo:

El objetivo de este capítulo es presentar un panorama del Sector Manufactura en México, su evolución en los años más recientes, la importancia que representa en la economía nacional, así como los elementos más representativos de su desarrollo.

Consideramos importante incluir esta sección para contextualizar el trabajo aquí presentado, el impacto de un sistema de automatización de la fuerza de ventas en este sector, así como la contribución en términos de negocio que justifica esta inversión.

Como primer paso, se incorporó un análisis del sector por cada segmento que lo compone, su aportación en la economía y los órganos que la regulan. Se explica asimismo la segmentación que se aplica de acuerdo al tamaño de la empresa, su clasificación, el valor porcentual de participación en el mercado de cada segmento, así como las estadísticas de personal empleado en cada una.

Seguido de esto, se presenta un esquema conceptual y operativo de la cadena de valor del sector manufactura y su relación con el sector comercial. Se describen las áreas e interrelaciones que las componen, así como la forma en que el sistema de automatización de la fuerza de ventas propuesto en este trabajo se relaciona con este entorno.

Más adelante se describen los elementos más representativos del sector manufactura al que está orientado el desarrollo del sistema. Asimismo, se presenta una perspectiva del sector manufactura, sus necesidades, indicadores de eficiencia y eficacia, así como las tendencias en la cadena de valor hacia los clientes que está haciendo que las empresas incorporen como parte de su modelo de negocio herramientas que les permitan incrementar su productividad, mejoren el servicio a sus clientes y se orienten hacia una economía apoyada en la tecnología que favorezca su ventaja competitiva en el mercado.

La última sección de este capítulo está orientada a describir cómo los sistemas de automatización de la fuerza de ventas apoyan el desarrollo empresarial, y específicamente del sector manufactura, a través de una incorporación de herramientas sistematizadas que optimizan la comercialización de productos, favorecen la planeación e incrementan la productividad de las empresas. En específico, se presentarán los elementos que el sistema propuesto en este trabajo ofrecerá como una respuesta a las necesidades del sector manufactura del país en materia comercial, destacando los indicadores de negocio cualitativos y cuantitativos de las empresas y sus clientes que justifican este desarrollo.

I.1 Panorama general del sector manufactura en México

El mercado al que se dirigirá este sistema es hacia las empresas medianas y grandes, y con foco principal al sector industrial con enfoque a manufactura de productos de consumo.

Actualmente la segmentación de las empresas de acuerdo al INEGI es la siguiente:

Tamaño	Clasificación por Número de Empleados		
	Industria	Comercio	Servicios
Micro Empresa	0 – 30	0 - 5	0 - 20
Pequeña Empresa	31 – 100	6 - 20	21 – 50
Mediana Empresa	101 – 500	21 - 100	51 – 100
Gran Empresa	501 en adelante	101 en adelante	101 en adelante

Figura 1.1 Segmentación de empresas

El sector manufacturero de acuerdo con su tamaño se clasifica en micro, pequeño, mediano o grande. Se consideran para su clasificación, cortes a nivel nacional, por entidad, subsector, las 10 principales ramas de actividad, por tamaño, y después de esto se ordenan conforme al nivel de empleo.

La actividad manufacturera por subsector y tamaño se ubica con una operación de 344 mil 118 establecimientos, de los cuales el de alimentos, bebidas y tabaco participó en 1999 con el mayor número: 117 mil 616 establecimientos (34.1% del sector) y 791 mil 297 personas (18.7%). De estos establecimientos, el 97.8% corresponde a la microempresa.

En el subsector de textiles, prendas de vestir e industria del cuero, la participación conjunta de la mediana y gran industria representa el 77.1% de las remuneraciones, el 79.1% de los activos fijos netos y el 72.5% del valor agregado censal bruto con tan sólo el 3.3% de los establecimientos. En el subsector de madera y sus productos la micro industria generó el 34.7% del valor agregado censal bruto, mientras que la grande aportó sólo el 9.9% dentro del mismo.

En el subsector de industrias metálicas básicas, la micro industria tiene la menor participación en el valor agregado censal bruto que cualquier otro subsector, con 0.4 por ciento. Por su parte, el subsector que mayor empleo generó fue el de productos metálicos, maquinaria y equipo, con un millón 318 mil 808 personas, destacando los establecimientos grandes que ocupan al 54.5% del personal, empleado en solamente el 1.0% de los establecimientos.

En la figura 1.2 se muestra esta composición por subsector manufacturero, que incluye por cada uno su Valor Agregado Censal Bruto, Personal Ocupado y Número de Establecimientos.

Establecimientos, Personal Ocupado y Valor Agregado Censal Bruto del Sector Manufacturero, según subsector

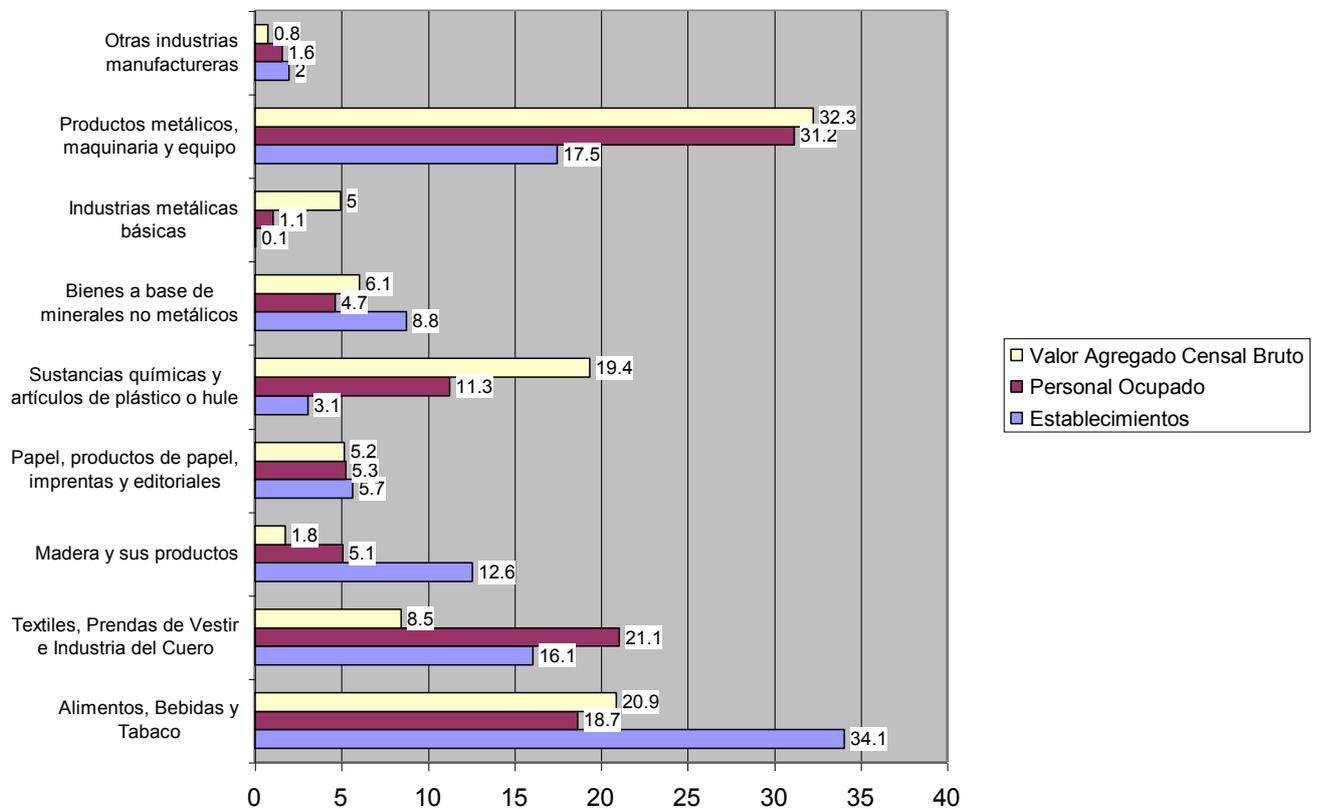


Figura 1.2 Composición de los subsectores Fuente INEGI 1999

Con este escenario podemos determinar en qué subsectores el sistema puede tener mayor impacto, basándonos en su tamaño de mercado, la manera en que realizan la comercialización de sus productos, así como su participación en la economía nacional. Para entender mejor el negocio de las empresas a las que se dirige este sistema, se ha integrado a continuación una sección donde se presenta un análisis de las necesidades específicas del sector manufactura, sus tendencias y la perspectiva de integración tecnológica en su modelo de trabajo.

1.2 Modelo de comercialización de productos

Las empresas de manufactura siguen un modelo de negocios que parte desde la planeación hasta el embarque de sus productos, para después ligarse con los sistemas de planeación, aprovisionamiento y administración de las empresas comercializadoras. En este caso, las empresas de manufactura como se explica en la figura 1.3, se consideran proveedoras de bienes de consumo, y las comercializadoras son los clientes a los que estará ligada la labor comercial de la fuerza de ventas de las manufacturas.

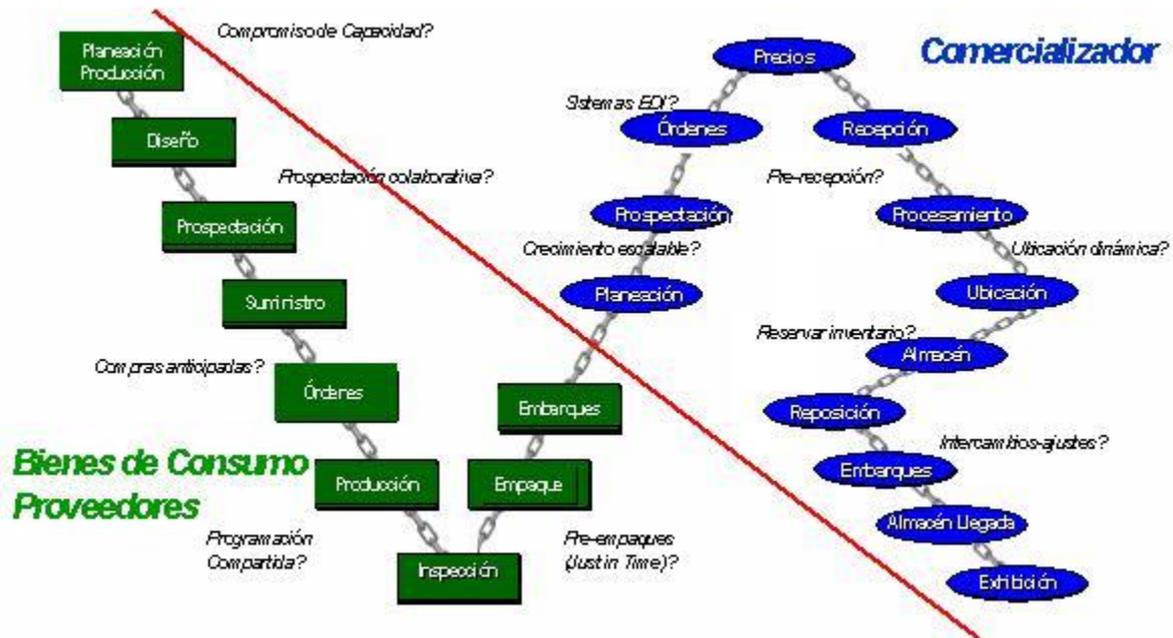


Figura 1.3 Flujo de comercialización

Para este fin, se ha definido que el sistema aunque está orientado al sector manufactura, tiene un impacto directo en las empresas comercializadoras, ya que al llevar una planeación de ventas con base a existencias, se impacta directamente a éstas últimas para su eficiencia en la cadena general de aprovisionamiento.

De esta misma forma, existe una coordinación eficiente para las empresas comercializadoras, ya que se puede administrar la cadena completa desde la planeación hasta los embarques.

En este modelo, las empresas de manufactura pueden planear y diseñar su producción con base a lo que el mercado les demanda. Esta información indispensable para planear, es obtenida directamente por la información que los vendedores proveen al sistema mediante su interacción con los clientes. En este caso, la demanda del mercado se podrá alimentar a los sistemas de producción desde los sistemas de predicción y ventas como es el caso del sistema propuesto en esta tesis, o con una interfaz que pueda establecerse entre el sistema de planeación de recursos y suministros con el de ventas.

Siguiendo el diagrama anterior, una vez generada la producción con base a la demanda del mercado, se procederá entre otros pasos, al embarque de productos a los clientes del sector comercial. Una vez recibidos estos productos, los que fueron generados a través de una planeación basada en la demanda y control de inventarios, las empresas comercializadoras administran la venta y distribución de estos productos hacia el mercado.

La última etapa de este proceso es la exhibición y venta de productos. Para cerrar este ciclo, será importante tener disponible el estado que guarda la actividad de ventas, la que será uno de los principales factores de éxito del sistema, ya que la eficiencia y contribución del mismo a las empresas dependerá de la calidad de información que se alimente, su actualización y difusión oportuna.

En resumen, el nivel de información y eficiencia de un sistema de automatización de la fuerza de ventas, permitirá y favorecerá la planeación eficiente y eficaz de la producción en las empresas manufactureras, así como la posibilidad de incrementar sus ventas y penetración en nuevos mercados al extender las capacidades de la organización hacia sus clientes. De esta forma, los clientes entendidos como mercado meta, son partícipes de la planeación de la producción, e influyen en la oportunidad de que los productos lleguen al mercado bajo las condiciones requeridas, tanto en precio como características del mismo.

I.3 Situación del mercado

Las empresas de manufactura de bienes de consumo, así como las comercializadoras están reestructurando sus presupuestos para enfocarlos a los proyectos que están directamente relacionados con su negocio. Estas inversiones están principalmente orientadas a las tecnologías de almacenamiento y atención a clientes/ desarrollo comercial. En esta área, y mayormente la de atención a clientes- desarrollo comercial, es donde el sistema propuesto en esta tesis está directamente relacionado.

Estos presupuestos según Forrester Research (ver figura 1.4), se mantendrán al menos en un 80% en estos proyectos, y por lo que la relevancia de desarrollar este sistema se incrementa en forma significativa, ya que será una de las iniciativas con mayor impulso dentro de las organizaciones de estos sectores.

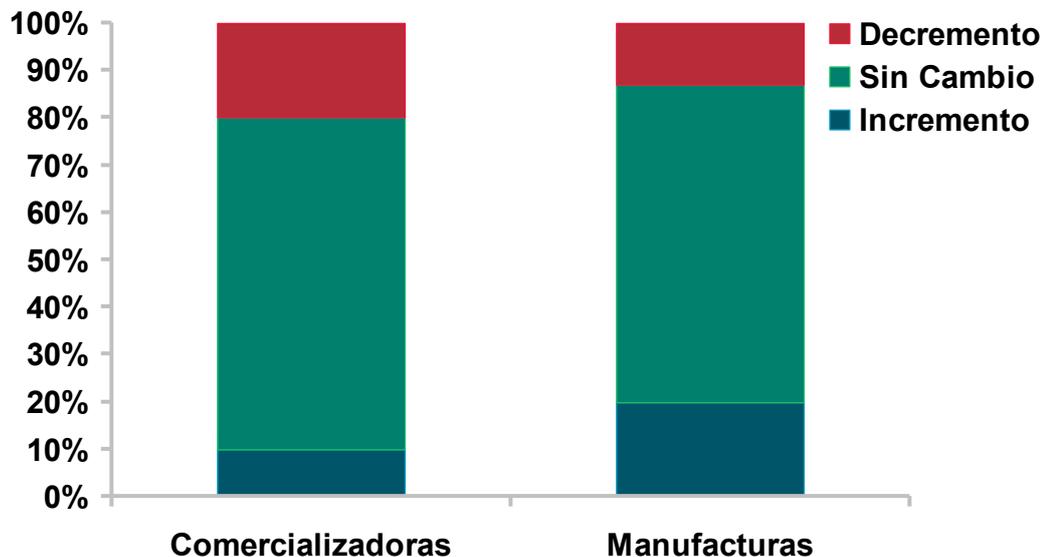


Figura 1.4 Aplicación de presupuestos (Fuente: Forrester Research, July 2003)

En este mismo sentido, Cutter Consortium (ver figura 1.5), realizó en julio del 2003 una encuesta donde a las empresas de manufactura se les preguntó su perspectiva sobre la criticidad o importancia de establecer estrategias de administración de clientes y

negocio utilizando los sistemas de información e Internet. Las respuestas determinaron que su interés coincidía en más de un 85% sobre el tema.

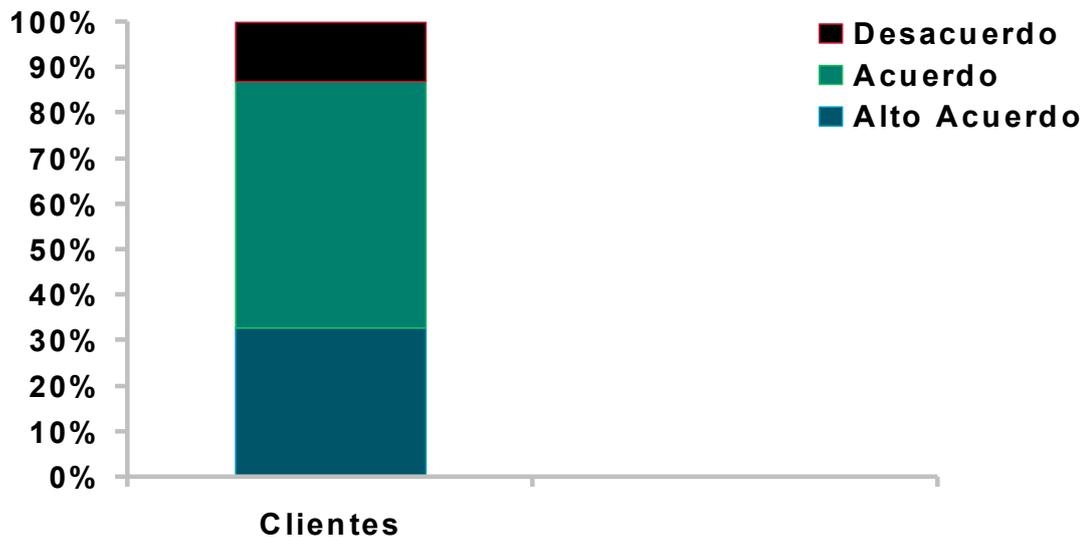


Figura 1.5 Relevancia para el Desarrollo de Estrategias de Negocio con uso de Tecnología (Fuente: Cutter Consortium, July 2003)

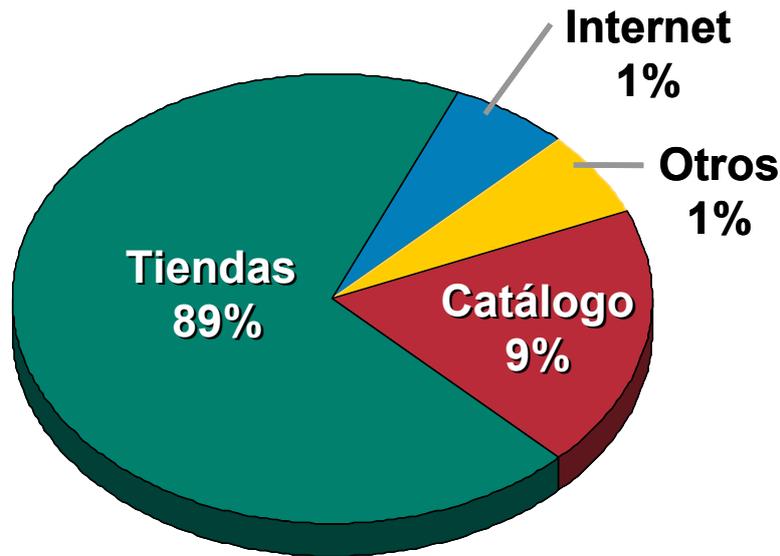
Con este fundamento, Cutter Consortium afirmó que las empresas que se enfocaran bajo esta visión serán las que determinen el rumbo de la industria, ya que las diferencias entre estas empresas y las que decidan no invertir en estos modelos, se encontrarán en el mediano plazo con un mercado mucho más demandante al que no podrán responder con la velocidad adecuada.

Las tiendas son el Activo más importante de las comercializadoras.

Para las empresas de comercialización, las tiendas siguen constituyendo la mayor parte de sus ingresos, y constituyen la fuente más importante de donde provienen las mejoras a la eficiencia de su negocio. Esta aseveración se soporta en un estudio realizado por Kurt Balmon Associates en el año 2003, donde presenta que en promedio en la industria del comercio, las tiendas siguen representando el 89% de los ingresos de las empresas comercializadoras, los catálogos un 9% y el otro 25 lo constituyen las operaciones realizadas entre otros medios por Internet.

Estos son ingresos globales de los mercados de comercialización, más no excluye que existen empresas que realizan su comercialización a través de medios electrónicos como el Internet, pero comparativamente con el resto de los mercados su participación aún no es representativa.

En la figura 1.6 se presenta una composición de ventas en el mercado a través de los medios existentes.



Ventas Comercios

Figura 1.6 Composición de medios de venta (Fuente: Kurt Salmon Associates)

Transición de un modelo con enfoque a suministro de producto a otro basado en la demanda del consumidor

Tradicionalmente, el modelo de comercialización de productos se generaba a partir de una producción en serie, se alineaban los procesos de negocio para este modelo hacia el interior de las organizaciones únicamente para su eficiencia interna, y no hacia el mercado.

Este modelo ha cambiado sustancialmente a enfocarse a una nueva relación de negocio con un carácter corporativo, es decir hacia el mercado. En este sentido, el esquema que hoy está predominando es el orientado hacia los procesos de negocio, donde cada entidad está relacionada en función de los clientes y sus requerimientos.

La siguiente figura 1.7 ejemplifica la forma en que este cambio se ha llevado a cabo, considerando los elementos que integran el proceso.

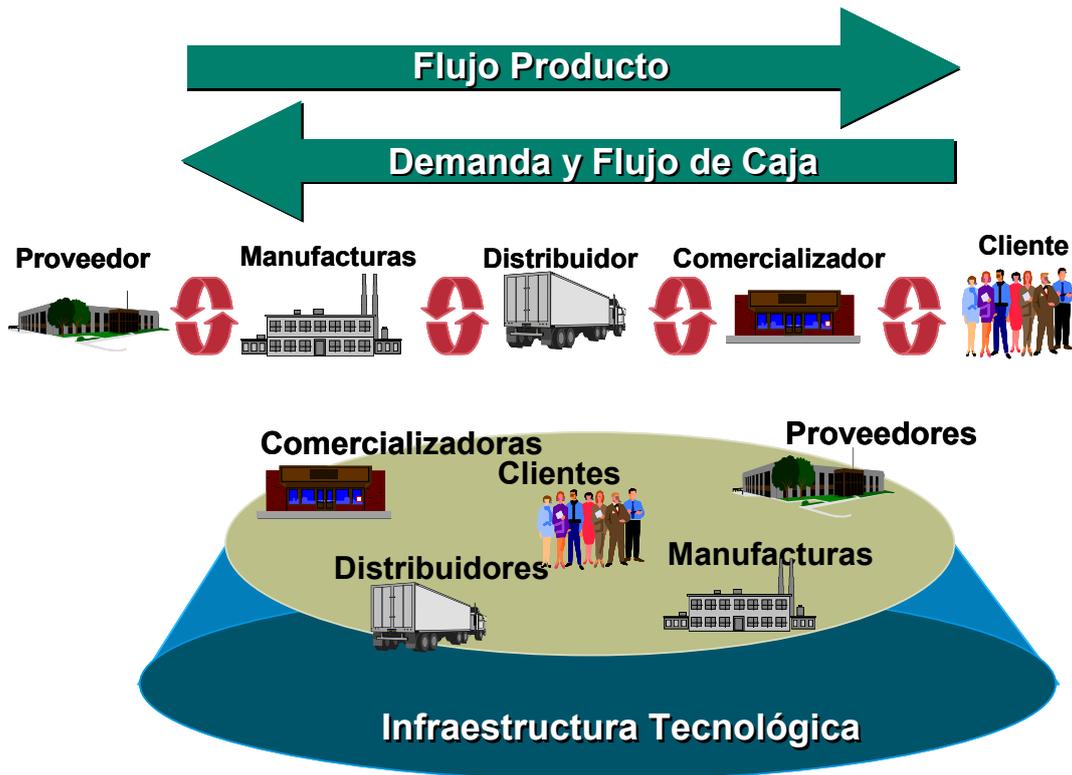


Figura 1.7 Proceso de Comercialización Actual

Principios financieros del modelo integrado: Manufactura – Comercialización – Distribución - Transporte

A nivel de negocio, las empresas de estos segmentos están basando sus indicadores de éxito en los criterios que se muestran en la siguiente tabla. Estos elementos nos muestran cómo las empresas de estos segmentos se transforman de una visión de únicamente cobertura de mercado a la rentabilidad de los negocios, cuyo valor se centra en el conocimiento de la organización.

Los objetivos son medidos a nivel organizacional y su estrategia de negocio está orientada a los clientes. Estas empresas están tomando las ventajas que la tecnología les provee para hacer más eficientes su operación y reducir sus costos.

De igual manera, la operación del negocio cambia de decisiones basadas en políticas y procedimientos a la estructuración de reglas de negocio.

Perspectiva Transformadora de Negocio en las Empresas de Manufactura

Ayer	Hoy
Tamaño de mercado	Rentabilidad
Datos	Conocimiento
Objetivos departamentales	<i>Balanced Scorecard</i> Medición organizacional
Enfocado producto/canal	Enfocado al Cliente
Sistemas en Tienda	Web-Internet
Entrenamiento en Administración y Operaciones	Entrenamiento sobre Producto/Habilidades
Políticas y procedimientos	Reglas de Negocio

Figura 1.8 Perspectiva transformadora de negocio en las empresas de manufactura

Bajo esta perspectiva, los objetivos de negocio de las empresas de estos sectores, están orientándose a los elementos que se mencionan a continuación en esta tabla:

- Incrementar productividad / menos empleados
- Incrementar lealtad clientes
- Promover comunicación y cultura
- Generar nuevos ingresos
- Producto adecuado, tienda adecuada

Figura 1.9 Factores de cambio buscados en las empresas de manufactura

Este enfoque está promoviendo tres aspectos fundamentales que se mencionaron anteriormente, que es donde las empresas están dedicando sus recursos: cadena de aprovisionamiento, servicio a clientes y tecnología para soporte al negocio.

Las empresas hoy en día requieren obtener mejores resultados con menos empleados o más eficientes, que apoyen mediante una cultura de trabajo de enfoque al cliente. Todo esto llevado a cabo a través de una comunicación efectiva, que le de el factor de oportunidad en la toma de decisiones.

De esta manera, podrán las empresas de estos sectores, y muy particularmente el de la industria manufacturera el incrementar sus ingresos actuales y generar otros, así como el mejorar en forma general su cadena de aprovisionamiento y comercialización de productos.

Estrategias de transformación para una nueva cadena de valor en manufactura

Los aspectos esenciales para lograr un cambio estructural en las empresas de manufactura están dirigidos desde la manera en que se captan, mantienen y desarrollan los clientes. De igual manera, es importante el contar con un mecanismo eficiente para operar y gestionar las órdenes de producto generadas.

En este caso, el sistema de automatización de la fuerza de ventas impacta directamente en la eficiencia comercial, la planeación de la producción, y sobre todo la administración de los costos a través de una eficiente función comercial que determina dónde hay que vender, cuándo, cuánto y sobre todo cómo producir los bienes que se estén comercializando en cantidad, calidad y oportunidad.

Una vez captada la venta, dependemos de cómo esté estructurada la cadena de aprovisionamiento y los controles internos. De ello dependerá la calidad en el servicio global que se ofrezca a los clientes.

Al mantener una mayor cercanía con los clientes, podremos lograr 2 objetivos adicionales: 1) un desarrollo de productos orientado al mercado, 2) una respuesta inmediata o con oportunidad que se traduzca en una agilidad organizacional tanto en las competencias personales, en los equipos que las integran, así como en la eficacia de sus procesos.

Con esto, se podrá realizar una oferta al mercado mucho más enfocada, con conocimiento de cada sector, se incrementarán las posibilidades de generar alianzas con mayor sentido de negocio, y sobre todo se fortalecerá el desarrollo de soluciones a los clientes bajo una perspectiva de alta calidad.

Por lo tanto, dentro de esta nueva cadena de valor se requiere la incorporación de un sistema que facilite el intercambio de información de ventas con los clientes, agilice la comunicación con las áreas responsables de la producción y los servicios, y sobre todo permita estar desempeñando su función comercial sin importar si se encuentre o no dentro de las instalaciones del cliente.

Uno de los elementos más representativos de esta iniciativa de automatización de la fuerza de ventas, es su contribución al flujo de información sobre los clientes, lo que nos apoyará a cubrir la necesidad de información oportuna para la planeación de la producción a través del conocimiento de la demanda real.

Seguido de esto, los sistemas de distribución y almacenamiento podrán conocer con detalle los requerimientos de inventario para proyectos, la forma en que deberán cubrir la demanda, y el comercializador podrá entonces administrar en forma flexible sus espacios de exhibición, contar con disponibilidad de producto, y sobre todo enfocarse a los mercados que más les interese cubrir.

Bajo todo esto se encuentra un proceso clave de todo el modelo, que es la administración de órdenes e inventarios.

En este escenario, el sistema de automatización de la fuerza de ventas propuesto, podrá contribuir con información oportuna de la demanda, las órdenes de compra procesarlas bajo un proceso flexible, y sobre todo ofrecer un valor tangible a los clientes.

I.4 Descripción general de los sistemas de automatización de la fuerza de ventas

Enfoque a la Administración de la Información

Sin duda, en un sector como es el de manufactura, su fin último es la colocación eficiente y eficaz de sus productos en el mercado. Para ello, la administración de inventarios facilitará la planeación de la producción a través de un conocimiento de la demanda. Acto seguido, la administración de dicha producción debe en una manera productiva a través de modelos de control, el facilitar el acceso a la información en centros de logística y distribución para inventariar los productos que el área comercial esté moviendo.

De esta forma, uno de los procesos clave en este sector manufactura lo constituirá la administración de órdenes proveniente de la función comercial. Por lo tanto, el enfoque a Administración de la Información se sustentará en forma significativa de lo que un sistema de Administración de Ventas genere a través de la demanda que los vendedores alimenten al mismo.

En la figura 1.10 se explica gráficamente este proceso que va desde la demanda y la forma en que se ataca al mercado, hasta el impacto en la producción y la administración de los inventarios.

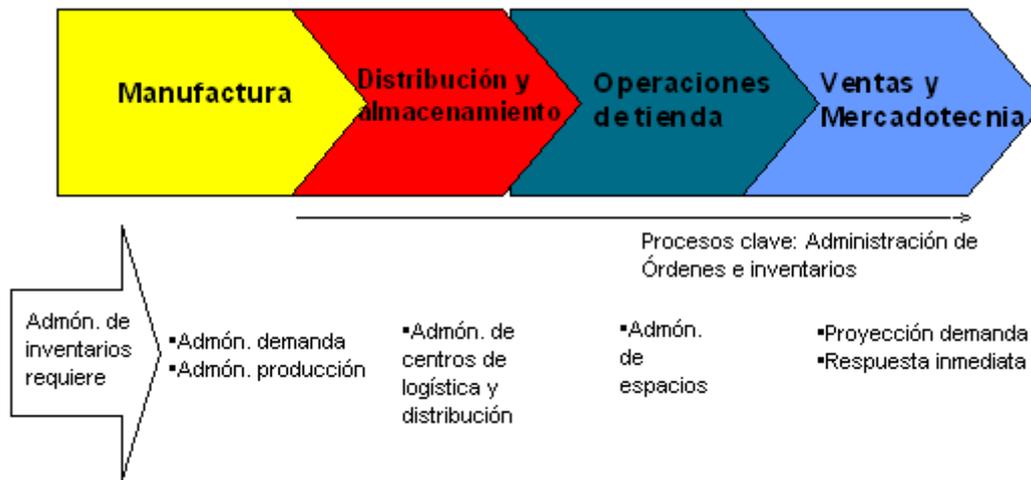


Figura 1.10 Procesos en las empresas de manufactura

Administración de las Relaciones con Clientes (CRM - Customer Relationship Management)

El concepto CRM o Administración de las Relaciones con Clientes, ha tenido un impacto significativo durante los últimos años, ya que en su primera acepción es un sinónimo de mayor Servicio a Clientes. CRM es una evolución de las ventas y la mercadotecnia, donde el esquema de SFA o Automatización de la Fuerza de Ventas es el que se encarga de administrar los contactos, cuentas y oportunidades.

De esta forma, CRM agrega el elemento de Servicio y Soporte a Clientes a través de la integración de los recursos humanos, los procesos y la tecnología. Por lo tanto, CRM establece una relación de mutuo beneficio entre las empresas y sus clientes.

CRM establece 10 objetivos principales:

1. Incrementar ventas y acelerar el proceso comercial
2. Generar mayores ventas cruzadas (servicios ligados)
3. Disminuir la cantidad de errores que se generan
4. Incrementar el volumen de ventas y transacciones
5. Disminuir costos de adquisición de clientes
6. Incrementar el porcentaje de retención de clientes
7. Incrementar la comunicación con clientes y hacer más eficientes los procesos de atención
8. Enfocarse a clientes rentables
9. Reducir costos de operación y tiempos
10. Definir métricas más efectivas de productividad y servicio a clientes

El concepto de SFA o Automatización de la Fuerza de Ventas (Sales Force Automation) está referido a la manera en que la organización está ligada internamente hacia los clientes. Los elementos a nivel conceptual que componen la automatización de la fuerza de ventas son:

- Bitácora de actividades diarias
- Registro de transacciones de negocio con clientes
- Base de información de clientes
- Colocación y administración de órdenes
- Administración de la actividad comercial con clientes
- Pronóstico de ventas basado en oportunidades
- Documentación interna

I.5 Respuesta del sistema propuesto a la problemática del sector

El sistema propuesto a desarrollar deberá permitir la automatización del área de ventas a través de un esquema remoto y central, en que cada representante de la fuerza de ventas se constituya como un asesor de negocios, todo esto en un ambiente integral en el que se podrán practicar esquemas de colaboración y flujos de trabajo, además de consolidar una base de datos centralizada para administrar el conocimiento del negocio.

Permitirá a los ejecutivos de ventas establecer y mejorar las relaciones empresariales con sus clientes, además del intercambio de información de primera mano, entre las diferentes áreas que conforman el departamento de ventas, agilizando la captura de información del negocio desde puntos de venta remotos para consolidarla en un sistema de información central para la toma de decisiones y la integración con los sistemas empresariales (ERP, Legacy Systems), así como brindar información actualizada resultado de la operación diaria en cada uno de los puntos de venta remotos.

Se espera que este sistema apoye en los siguientes aspectos:

- Una adecuada administración de la información desde oficinas centrales, para otorgar la flexibilidad y adaptabilidad que los negocios requieren.
- Una integración entre la computadora, la telefonía y la administración de relaciones con los clientes.
- Responder a las tendencias del mercado, desarrollar ventas y estrategias acordes a las necesidades de los clientes de las organizaciones del sector manufactura.
- Brindar una gama de herramientas a los ejecutivos de ventas para vender, entregar servicios, responder e incluso adelantarse a las necesidades de los clientes, haciendo el negocio más predecible.

Ventajas esperadas

- Permitir el manejo de un gran número de usuarios y de altos volúmenes de información bajo la utilización de un sistema central (Middle Ware).
- Conectividad a los sistemas de información existentes.

Los objetivos que definirán la efectividad del sistema objeto de esta tesis a nivel cualitativo y cuantitativo para cubrir las necesidades que las empresas del sector manufactura están requiriendo, se especifican en el capítulo siguiente en la etapa de Planeación.

Es importante mencionar que el cumplimiento de estos objetivos no sería posible en forma aislada por el sistema, sino es a través de la cultura, información y procesos que se determinan al interior de las organizaciones al implementar un sistema de Administración de la Fuerza de Ventas.

Conclusiones del capítulo:

A través de este capítulo se ha descrito la relevancia de enfocarse al sector manufactura del país para hacer más eficientes sus operaciones, la forma en que se enfocan al mercado, así como incrementar su efectividad comercial a través de la Automatización de la Fuerza de Ventas.

Consideramos que este sistema favorecerá al segmento industrial, cuyo enfoque ha sido tradicionalmente a la producción, con resultados poco efectivos en su posicionamiento estratégico y oportuno con los clientes en cuanto a conocimiento de mercado se refiere.

Se buscará enfocarse en forma adaptable a los cambios tecnológicos de los sistemas de información y las tendencias de negocio en su industria.

Se espera que estas iniciativas favorezcan a las empresas de este segmento a través de un manejo eficaz de los recursos comerciales, e intensifiquen la relación de negocio con sus clientes.

Capítulo II. Metodología de desarrollo

Objetivo del capítulo:

Describir la metodología empleada para el desarrollo del sistema, así como las herramientas que sirvieron de apoyo para facilitar el uso de la misma.

II.1 Problemática

Existen varias metodologías para el desarrollo de sistemas cada una tiene sus propias fortalezas y debilidades. Los desarrolladores experimentados frecuentemente usan una metodología como base y toman buenas ideas y soluciones de otras. En la práctica las diferencias entre las metodologías no fueron realmente tan significativas, con el paso del tiempo y el desarrollo de las mismas creció el parecido entre ellas, por lo que se han buscado maneras de cooperar.

El hecho de que no sólo muchos proyectos de software fallan para producir sistemas que cumplan los requerimientos del cliente y sus necesidades, sino también terminan excediendo presupuestos y los tiempos planeados, por eso las técnicas nuevas como la programación orientada a objetos, la programación visual y los ambientes avanzados de desarrollo han ayudado para incrementar la productividad.

Uno de los principales problemas con el desarrollo de software hoy en día es que muchos proyectos comienzan programando demasiado pronto y concentran mucho esfuerzo en escribir código. Esto se debe en parte a la falta de entendimiento del proceso de desarrollo de software por algunos administradores de proyectos que están ansiosos cuando su equipo de trabajo no está produciendo código.

Ciertas metodologías han intentado prevenir la urgencia de ver el desarrollo de sistemas como “un asunto pequeño de programación”, estas diferentes metodologías todas ellas con su propia notación única y herramientas, han dejado a muchos desarrolladores confundidos.

Para el diseño de este sistema se utilizó solamente la metodología de MSF (Microsoft Solution Framework) y se apoyó del lenguaje de modelado unificado UML (Unified Modeling Language).

- MSF (Microsoft Solution Framework)

Esta metodología se utilizó para la administración del proyecto, en donde es importante destacar que una de las razones fundamentales para llevar a la práctica esta metodología es tener un proceso de comunicación claro, fluido y asertivo para el desarrollo del proyecto en colaboración con el área de sistemas del cliente, quien solicitó estar involucrado en todas las fases del ciclo de vida del sistema.

- UML (Unified Modeling Language)

La decisión para utilizar este lenguaje se fundamentó en que se consolida como uno de los más utilizados para el diseño y desarrollo de componentes COM, sobre el que se apoyan herramientas como MS Visio, para llevar a cabo el diseño del sistema basado en una arquitectura de tres capas, lo que es un requerimiento técnico importante para permitir la escalabilidad del sistema en el proyecto actual y siguientes versiones de acuerdo a la visión del cliente.

II.2 MSF

Microsoft Solutions Framework (MSF) es un conjunto de modelos, conceptos y guías para la implementación de sistemas de información empresariales en un ambiente distribuido.

Como columna vertebral de esta metodología se tienen los siguientes principios:

- ¿Cómo dividir un proyecto en etapas?
- ¿Qué tareas se llevan a cabo en cada etapa?
- ¿Qué restricciones deben aplicarse?
- ¿Qué técnicas y herramientas se emplean?
- ¿Cómo se controla y gestiona un proyecto?

De ahí se desprenden las etapas que constituyen los capítulos III, IV y V de esta tesis, en donde a continuación se describe el objetivo de cada una de ellas y los entregables en las mismas.

Microsoft Solutions Framework está basado en un conjunto de modelos, los que incorporan tres factores fundamentales de éxito:

- Un punto de visión, para proveer la guía requerida para tomar decisiones técnicas.
- Un conjunto de puntos de referencia, para realizar un seguimiento efectivo de la marcha de los procesos o proyectos, con énfasis en el manejo de los riesgos durante todo el ciclo de vida.
- Capacidad de reutilización, para tomar ventaja del conocimiento previo en forma estructurada y consistente en un ambiente tecnológico flexible.

MSF ayuda a las organizaciones a obtener los beneficios de las nuevas tecnologías mediante la aplicación de cinco modelos fundamentales. Estos modelos se aplican a diferentes categorías de problemas que están enmarcadas en la planeación, construcción o implementación y administración de los sistemas.

Los cinco modelos de MSF son:

- Modelo de arquitectura empresarial de MSF (Enterprise Architecture Model)
- Modelo de aplicaciones de MSF (Application Model)
- Modelo de equipos de trabajo de MSF (Team Model)
- Modelo de procesos de MSF (Process Model)

- Proceso de diseño de soluciones con componentes (Designing Component Solutions – DCS)

El modelo a seguir para este proyecto en la fase de desarrollo está orientado al modelo de procesos que se describe a continuación.

II.2.1 Modelo de procesos de MSF (Process Model)

Este modelo permite planificar y controlar proyectos orientados a resultados, balanceando los factores fundamentales que intervienen en dichos proyectos: alcance, tiempo y recursos. Es un modelo iterativo y basado en puntos de revisión (*milestones*).

El modelo de Procesos de MSF hace énfasis en:

- Una visión para el proyecto o sistema.
- Puntos de revisión que sincronizan el trabajo del equipo con las expectativas de los usuarios a lo largo de todo el proyecto.
- Asignación de prioridades mediante análisis de riesgos.
- Implementación incremental con puntos de revisión frecuentes.

El modelo de procesos de desarrollo de MSF consiste de cuatro fases interrelacionadas, como se muestra en la figura 2.1, cada una de estas fases representa un entregable para la cual debe ser establecida una línea base, antes de que el proceso de desarrollo pueda moverse a la siguiente fase del proyecto. Las cuatro fases y sus tareas primarias son:

- Planeación (Envisioning), la que debe producir una visión compartida. Sus entregables consisten en definir el equipo de trabajo, crear un documento de visión y alcance del proyecto, incluyendo un documento de riesgos.
- Análisis y diseño (Planning), la que debe producir un plan de proyecto detallado y la arquitectura de la aplicación. Sus entregables consisten en crear el documento de especificación funcional, el plan y calendario maestro del proyecto, además de crear un diseño conceptual, lógico y físico.
- Desarrollo (Developing), la que debe producir un producto completo y bien construido. Se crea el código fuente de la aplicación, se compila para crear los ejecutables y se definen los elementos de prueba.
- Aseguramiento de calidad (Stabilizing), la que debe producir un producto instalado y estable.

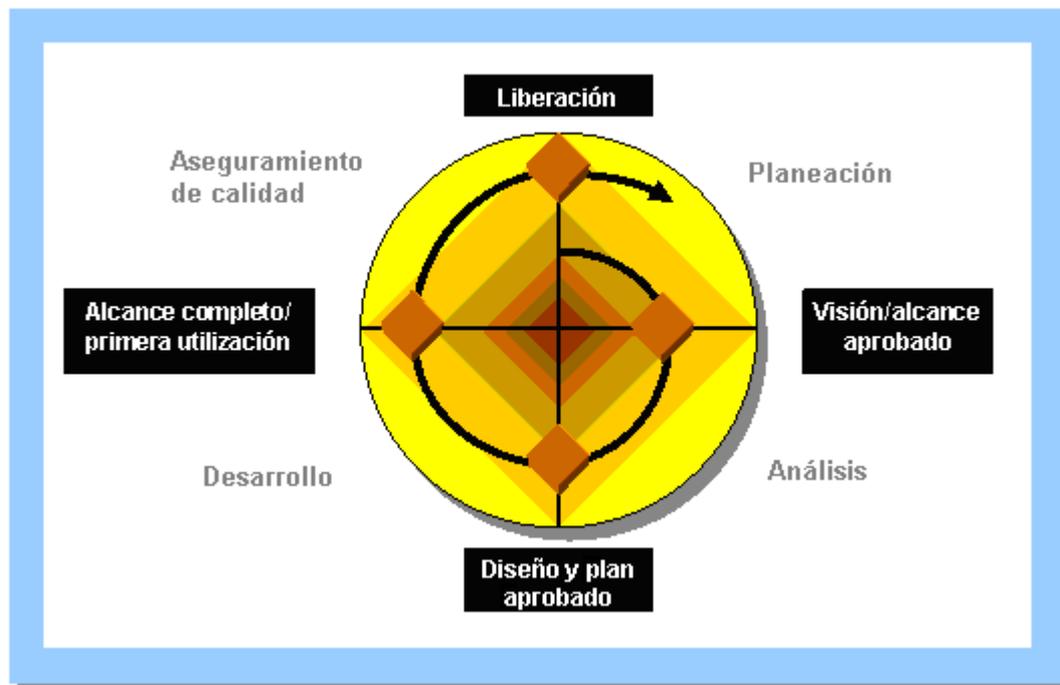


Figura 2.1 Las cuatro fases del modelo de procesos.

II.2.1.1 Fase de planeación

El objetivo de la fase de planeación es compartir entre los miembros del equipo y el cliente, una visión clara de los objetivos y metas del proyecto. Aunque la definición de la visión es un ciclo interactivo de refinamiento, al término del proceso de planeación, los responsables del proyecto deben tener aceptada la visión y deben ser capaces de comunicarla.

MSF emplea un documento de visión donde se detallan los siguientes elementos: visión, alcance del proyecto, antecedentes, requerimientos de negocio, entregables y riesgos.

Para que la visión esté completa, el equipo necesita hacer algunas investigaciones, enfocando sus esfuerzos en el cliente y otros productos similares siendo desarrollados o usados por la organización.

Una visión sólida provee lo siguiente:

- Claridad, se detalla no sólo lo que será hecho sino también lo que no se hará.
- Prioridades, se proveen los criterios necesarios para guiar al equipo en la toma de decisiones y entender no sólo qué se hará y porqué, sino también en que orden.
- Integración, el producto debe complementar y soportar la visión y funcionalidad de otros productos en la organización.
- Inversión Futura, debe formar las bases de productos futuros para ahorrar tiempo en la versión siguiente.

En cada paso de la investigación lo que ha sido aprendido se documenta, todavía el trabajo del equipo no es analizar la información sino simplemente registrar lo que se ha encontrado.

En la visión se incluyen las metas de largo alcance para el proyecto y las validaciones para moverse en la dirección correcta, en este punto se conoce suficiente información para determinar como está restringido, optimizado y aceptado, el triángulo de intercambio del producto, en recursos, características y fecha de liberación.

Un factor clave en el éxito del desarrollo de una aplicación, es la identificación de los riesgos que el proyecto pudiera encontrar, la administración de riesgos comienza en la fase de planeación; sin embargo, el proceso continúa a lo largo del proyecto, esta administración de riesgo se basa en el modelo de administración de riesgo de MSF.

Riesgo es la posibilidad de sufrir una pérdida. Para un proyecto esta pérdida puede ser de alguna de las siguientes formas:

- Disminución en la calidad del producto
- Incremento en los costos
- Incumplimiento en las fechas límite de entrega
- Falla completa en alcanzar las metas del proyecto.

La fase de planeación no consiste sólo en crear documentos, sino en comunicar la visión del proyecto dentro del equipo, los prototipos son un excelente método de comunicación.

El nivel de detalle en un prototipo de una aplicación es típicamente muy superficial, ya que puede consistir sólo de ejemplos de pantallas representativas, sin esfuerzo de programación, sin embargo el flujo de la aplicación puede ser descrito a través de un prototipo que siga los casos de uso definidos por el equipo.

En el paso de validación de esta fase, el equipo debe revisar y puede modificar el documento de visión, este es el punto en donde los miembros del equipo necesitan mirar su progreso y los entregables para asegurar que están adecuadamente preparados para aprobar la visión y avanzar a la siguiente etapa.

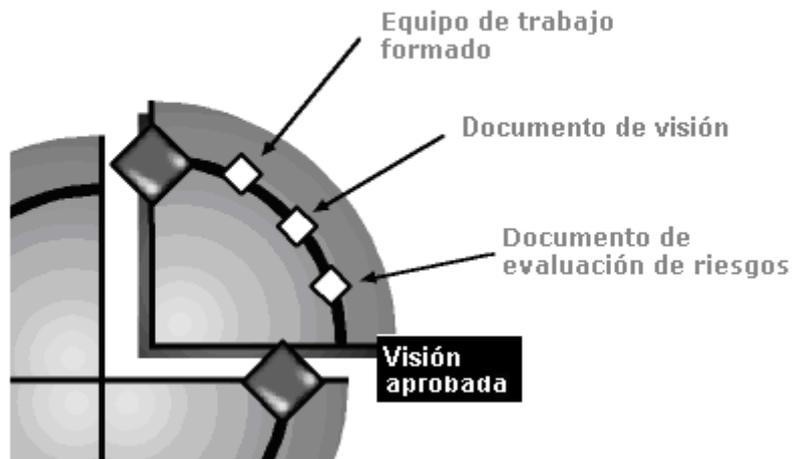


Figura 2.2 Las tres fases de la planeación

II.2.1.2 Fase de análisis

El objetivo de la fase de análisis es transcribir lo investigado del negocio actual y los requerimientos de usuario en documentos detallados para este propósito, así como generar la lista de las características deseables para que en el paso final el equipo asigne las prioridades y determine que características serán implementadas para la versión actual del producto.

La segunda de las cuatro fases del modelo de proceso de desarrollo de MSF es la fase de análisis, mostrada en la figura 2.3, la que continúa al término de la fase de planeación con el documento de visión aprobado. De todas las fases de este modelo, el análisis puede ser la fase más difícil de cumplir, la fase de planeación explora posibilidades, está llena de energía y entusiasmo, la fase de desarrollo es el tiempo de actividad de ponerse a construir con el sentimiento de que se está haciendo algo; a la mitad, está la fase de análisis donde los sueños de la fase de planeación son puestos a prueba contra la realidad.

Aquí se definen ¿qué funcionalidades son programadas hasta la versión siguiente?, ¿qué tecnología es la adecuada?, ¿qué costos no se pueden justificar? por lo que algunos equipos la evitan o la tratan de acelerar.

El trabajo hecho en esta etapa determinará el éxito o fracaso del proyecto, es mejor enfrentar la verdad ahora, en papel, que enfrentarla más tarde en una implementación fallida.

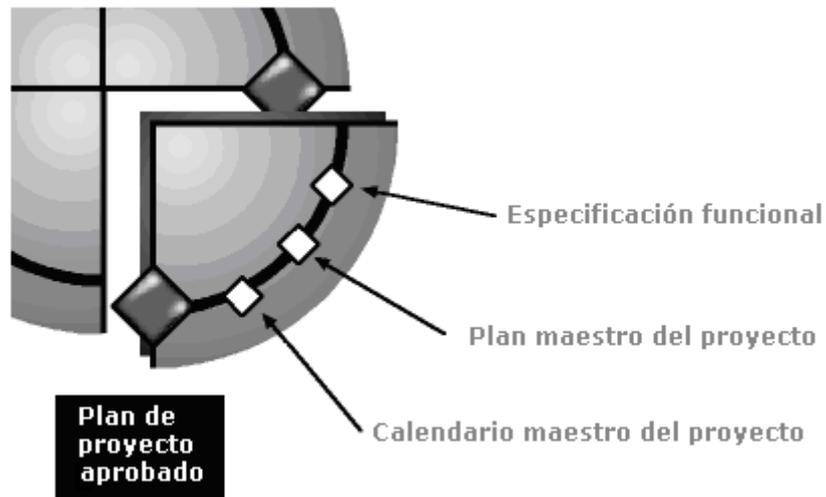


Figura 2.3 Los entregables de la fase de análisis.

Tres entregables son requeridos para alcanzar la aprobación del plan del proyecto, éstos son:

- Una especificación funcional, la que esboza el producto siendo desarrollado y las necesidades que cubrirá.
- Un plan de proyecto maestro, el que muestra como planea el equipo ejecutar el proyecto e incluye cualquier entregable subsiguiente que se intenta cubrir.
- Un calendario maestro del proyecto, el que esboza el tiempo requerido para cada entregable subsiguiente.

El objetivo de cualquier entregable es servir como una herramienta eficiente de comunicación, en este contexto, un entregable no necesariamente resulta en un artefacto de papel, un entregable puede tomar cualquier forma apropiada (un documento, un diagrama, una aplicación, una impresión de pantalla, un correo electrónico, etc.) mientras está facilite la comunicación.

“Haz un plan de trabajo y trabaja el plan”, es una frase que se oye con frecuencia, sin embargo las presiones diarias y las fechas límite de entrega evitan un análisis adecuado.

¿Por qué es tan importante el análisis?; porque un buen análisis identifica claramente la problemática a ser resuelta, ayudando a minimizar las omisiones de los aspectos funcionales, de actividades y entregables para lograr terminar de manera óptima un proyecto.

La figura 2.4 ilustra el costo relativo de corregir los defectos de diseño durante las cuatro etapas del modelo de proceso de desarrollo de MSF. El diagrama ilustra que el análisis reduce el costo en tiempo y recursos para corregir defectos causados por una falta de buen análisis.

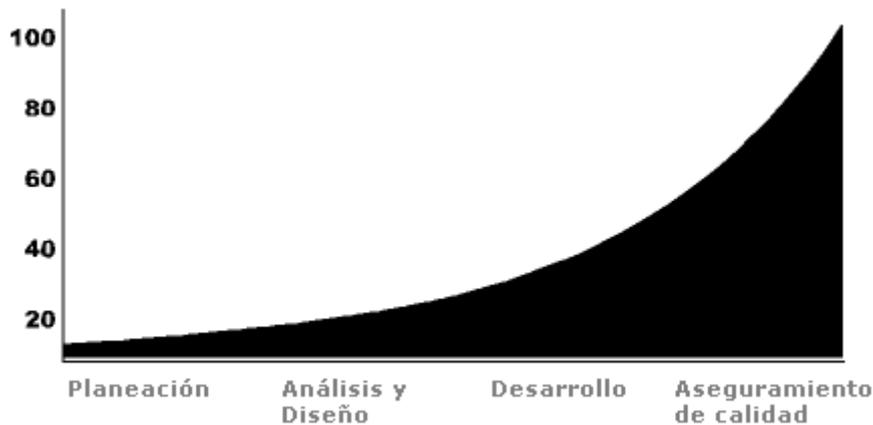


Figura 2.4 Costo de reparación por defectos de diseño en cada fase.

La meta de la fase de análisis es alcanzar la aprobación del plan del proyecto, que es la culminación del trabajo del equipo durante esta fase y significa un acuerdo entre el cliente y el equipo en asuntos críticos del proyecto.

Tabla 2.1 Acuerdos alcanzados por el plan de proyecto aprobado

Punto de Acuerdo	Documentado en
¿Qué construir para cubrir las necesidades del negocio?	Especificación funcional
¿Qué funcionalidad tiene prioridad?	Especificación funcional
¿Cuánto tiempo tomará completar el proyecto?	Calendario maestro del proyecto
¿Cómo construir el producto y quién lo construirá?	Plan de proyecto maestro
Fechas y entregables	Plan de proyecto maestro

II.2.2.3 Fase de diseño

El objetivo de la fase de diseño es transcribir las necesidades y los requerimientos del negocio obtenidos en las etapas anteriores, en una serie de documentos técnicos que permitan al equipo de programadores entender lo que tienen que construir, para escribir el código que en conjunto formará la solución.

Para esta fase la metodología de MSF provee el modelo de proceso de Diseño (MSF Design Process Model), que consiste de tres fases que permiten un enfoque interactivo de gran flexibilidad. Cada fase genera un modelo del mismo nombre.

- Modelo de diseño conceptual
- Modelo de diseño lógico
- Modelo de diseño físico

Cada una de ellas con su perspectiva ya que están orientadas a audiencias diferentes, los usuarios, el equipo del proyecto y los desarrolladores. Usar este modelo ayuda a asegurar que las aplicaciones son creadas no sólo por usar tecnología, sino para ajustarse a los requerimientos de negocio y de los usuarios, como se muestra en la tabla 2.2.

Tabla 2.2 Tareas de diseño enfocadas a las tres fases del proceso de diseño

Fase	Perspectiva	Tarea
Conceptual	Ve el problema desde la perspectiva del usuario y del negocio	Define el problema y la solución en términos de escenarios
Lógica	Ve la solución desde la perspectiva del equipo del proyecto	Define la solución como un conjunto de servicios de cooperación
Física	Ve la solución desde la perspectiva de los desarrolladores	Define los servicios y la tecnología de la solución

El modelo de proceso se mueve de requerimientos tangibles (casos de uso) a un diseño abstracto de aplicación (diseño conceptual), de aquí a un diseño de aplicación racionalizado (diseño lógico), posteriormente a un diseño de aplicación concreto (diseño físico) y finalmente a un producto tangible.

Estas tareas tienen tres características:

- No son secuenciales, donde el resultado de cada parte es usado como entrada para la parte siguiente, como se muestra en la Figura 2.5.
- Interactivas, las actividades tienen su propio ciclo de validación, diseño prueba y rediseño.
- Espirales, cada interacción de las tres actividades representa progreso hacia la aprobación del plan maestro.

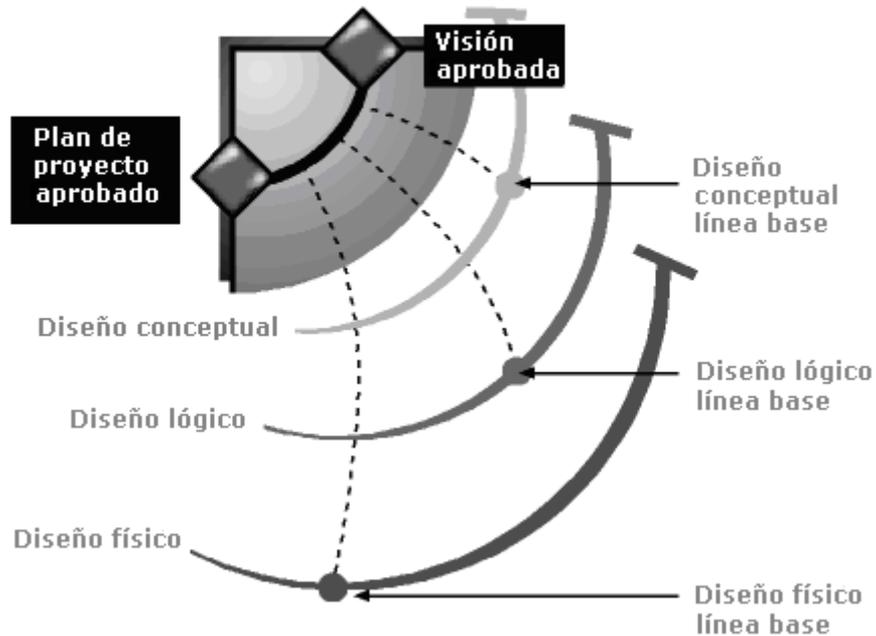


Figura 2.5 Las tres fases del modelo de proceso de diseño

Obviamente, un mínimo grado de secuencia debe existir como una cuestión práctica. Todo este proceso es requerido para completar la especificación funcional.

Diseño conceptual

El diseño conceptual no toma en cuenta el enfoque o las tecnologías necesarias para construir la solución. Es la traducción de los requerimientos de negocio y del usuario del lenguaje de los usuarios al lenguaje de los desarrolladores.

La actividad del diseño conceptual comienza cuando el equipo acuerda una visión, lo que pudiera ocurrir antes de que el equipo obtenga una visión aprobada formalmente.

Las metas del diseño conceptual son:

- Identificar las necesidades de negocio
- Entender lo que hacen los usuarios y lo que ellos requieren.
- Generar escenarios que reflejan los requerimientos de manera completa y precisa, involucrando al cliente, al usuario y a otros interesados.

Consiste de tres pasos: investigación, análisis e implementación.

Investigación involucra:

- Obtener respuestas a preguntas claves.
- Identificar procesos claves de negocio y actividades.
- Dar prioridades a procesos y actividades.
- Identificar usuarios y crear perfiles.

Análisis involucra:

- Revisar a fondo la investigación de usuarios y negocio.
- Crear escenarios para describir contexto, flujos de trabajo, secuencia de tareas y relaciones.

Implementación involucra:

- Mejorar el flujo de trabajo y la solución que la soporta.
- Validar y evaluar el trabajo, rediseñar.

Diseño lógico

El diseño lógico comienza a involucrarse en los detalles de la aplicación que el equipo construirá para satisfacer las necesidades de negocio y los requerimientos del usuario.

Las metas del diseño lógico son:

- Organizar la solución y la comunicación entre sus elementos.
- Toma el problema de negocio identificado en los escenarios del diseño conceptual y formula un modelo abstracto de la solución.
- Siguiendo el flujo de la figura 2.5, el diseño lógico toma los escenarios del diseño conceptual para crear objetos y servicios.
- Prototipos de interfaz de usuario.
- Un diseño lógico de la base de datos.

El diseño lógico tiene dos pasos: análisis y racionalización. (No existe el paso de investigación, porque el resultado del diseño conceptual es la entrada del diseño lógico)

Análisis involucra:

- Identificar los objetos de negocio y de servicios.
- Identificar atributos y relaciones.

Racionalización involucra:

- Verificar los objetos de negocio
- Identificar objetos de negocio y escenarios implícitos.

Diseño físico

El diseño físico selecciona la tecnología que será usada, este diseño agrega detalle a la arquitectura y refleja las restricciones de implementación.

Las metas del diseño físico son:

- Aplicar restricciones de tecnología, incluyendo implementación y consideraciones de desempeño a los resultados del diseño lógico especificando los detalles de la solución.
- Los resultados del diseño lógico son usados para producir componentes, especificaciones a las interfaces de usuario.
- El diseño físico de la base de datos.

El diseño físico representa la última oportunidad para validar el diseño antes de escribir código. Una vez que el equipo ha comenzado el desarrollo (a escribir código), los cambios en el diseño pueden aún ser hechos, pero el costo para el proyecto en tiempo y recursos llega a ser mucho más elevado que en la fase de planeación.

El diseño físico consiste de cuatro pasos: investigación, análisis, racionalización e implementación.

(El paso de investigación es requerido para identificar tecnologías posibles a ser usadas en la aplicación).

Investigación involucra:

- Determinar restricciones físicas de la infraestructura.
- Determinar requerimientos físicos de la solución.
- Administrar riesgos de los conflictos entre restricciones físicas y requerimientos.

Análisis involucra:

- Seleccionar tecnologías candidatas para la implementación.
- Hacer un borrador de un modelo preliminar de instalación compuesto de la red, datos y topologías de componentes.

Racionalización involucra:

- Determinar una estrategia de empaquetar y distribuir.
- Descomponer objetos en componentes basados en servicios.
- Distribuir componentes entre topologías.
- Redefinir empaquetar y distribución.

Implantar involucra:

- Determinar el modelo de programación.
- Especificar la interfaz de los componentes.
- Especificar los componentes en el lenguaje de desarrollo.

- Entender las consideraciones en la estructura de los componentes.

II.2.2.4 Fase de desarrollo

El objetivo de la fase de desarrollo o programación, es convertir las clases identificadas en la fase de diseño a código, usando algún lenguaje de programación, ya que muchas aplicaciones sufren de una pobre implementación, la transición entre las fases anteriores debe enfatizar la importancia de escribir un código sólido para la aplicación.

Una aplicación toma vida durante la fase de desarrollo, aunque también se realizan pruebas, seguimiento de errores y todas las características del producto y el código son incorporados a la aplicación

Como se ilustra en la Figura 2.6, la implementación del desarrollo guía a la codificación de la aplicación a estar terminada e inicialmente liberada al equipo del proyecto, el final de esta fase tiene las siguientes características:

- Todas las características del producto están implementadas, aunque todavía no estén completamente optimizadas.
- El producto ha pasado una etapa de pruebas básicas y la lista de errores ha sido identificada, aunque no necesariamente corregidos.
- Los miembros del equipo y usuarios claves acuerdan que las características incluidas se ajustan a la visión y diseño del producto y han sido implementadas exitosamente.
- Los materiales de rendimiento del usuario son delineados y se alistan para la etapa de pruebas y estabilización (aseguramiento de calidad).

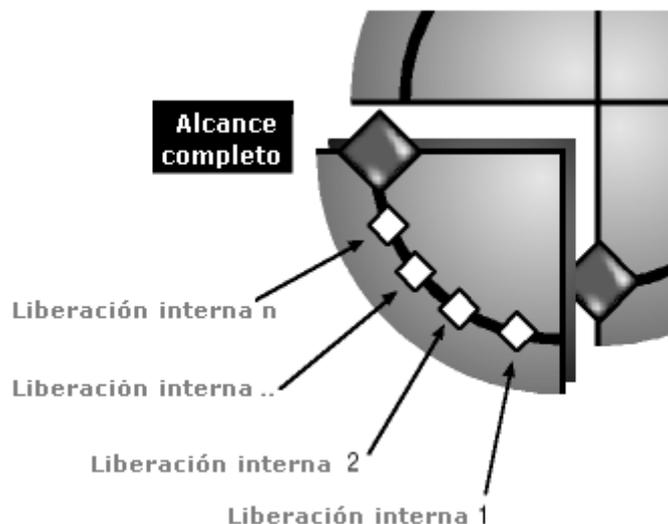


Figura 2.6 Las etapas de la fase de desarrollo

Los entregables para completar la fase de desarrollo incluirán versiones revisadas de los entregables de la fase de planeación, al igual que los siguientes:

- Código fuente y ejecutables
- Materiales de desempeño del usuario
- Elementos de pruebas

Los materiales de apoyo y desempeño del usuario documentan como trabaja la aplicación, tanto para los usuarios finales como equipos de soporte. Los elementos que soportan el desempeño del usuario van desde asistentes del producto (wizards), documentación de usuario y materiales para entrenamiento en salón de clases. Adicionalmente los elementos para los equipos de operaciones que soportarán la aplicación en producción describirán como está instalada, configurada, administrada y usada la aplicación.

En esta etapa los desarrolladores proveen pruebas funcionales iniciales de código y características.

La validación es un esfuerzo de todo el equipo, sin embargo, a lo largo de la fase de desarrollo la validación es particularmente responsabilidad de los equipos de desarrollo y pruebas, las pruebas de usabilidad, validaciones de desempeño, seguimiento de errores y mentalidad de calidad cero defectos, todas ellas se incluyen en este paso de validación.

Debido a que la liberación de múltiples productos ocurren a lo largo de la fase de desarrollo, existen muchas instancias en las que al menos una porción del equipo se enfocará en validar los entregables, por eficiencia para la validación del código de la aplicación, los equipos de desarrollo y pruebas deberán automatizar tanto como sea posible el proceso de pruebas.

Las liberaciones internas involucran agregar funcionalidad incrementalmente a una base conocida. Estas liberaciones internas también ayudan al equipo a estabilizar el producto, alcanzar metas de calidad y practicar el envío del producto. Cuando un equipo usa liberaciones internas durante el ciclo de desarrollo, es esencial establecer una línea base conocida, la que provee una herramienta de medida, el equipo puede entonces comparar la implementación de la especificación funcional con las necesidades de negocio y adicionalmente identificar puntos de diseño no descubiertos durante la fase de análisis.

Las liberaciones internas suceden a través de la fase entera de desarrollo, cada liberación está diseñada para expandir las liberaciones previas, esta independencia da un beneficio adicional de ver cada liberación como un producto. Revisar los resultados de una liberación interna es una parte necesaria de ajuste del equipo para convertirse en una organización de aprendizaje que repite las mejores prácticas y aprenden de los errores.

El código fuente y los ejecutables representan el entregable del equipo, de código y funcionalidad terminada. Tan pronto como el código producido por los programadores comienza a ajustarse a los requerimientos funcionales, la calidad del código deberá también ser considerada.

La eficiencia en desarrollo de código fuente y ejecutables es crítica para entregar un producto en tiempo. Desafortunadamente, muchos equipos sacrifican la calidad del código cuando están presionados para terminar un programa en una fecha límite en particular.

II.2.2.5 Fase de aseguramiento de calidad (Estabilización)

Los buenos equipos de desarrollo han sabido por años que las pruebas deben ser una mayor parte de un proyecto. Desafortunadamente muchos desarrolladores no prueban adecuadamente sus soluciones, ellos pudieran creer que su código es demasiado bueno que no necesita probarse o saben que su trabajo estuvo lleno de prisas y mal concebido que están temerosos de probarlo. Los desarrolladores experimentados saben que el software nunca está libre de error. Ellos también saben que sólo buenos procedimientos de prueba pueden minimizar errores de software, los desarrolladores experimentados no sólo esperan las pruebas, ellos las solicitan.

La funcionalidad del código es probada durante la fase de desarrollo, sin embargo, pruebas significativas en un ambiente como el de producción y de desempeño ocurren durante la fase de estabilización. Durante esta fase, todas las cuestiones conocidas son resueltas antes de la entrega y las tareas necesarias para soportar el mantenimiento del producto son completadas, esta fase está formada por documentación, notas de liberación y la implementación del producto.

La fase de estabilización comienza cuando el equipo cambia su foco del desarrollo de código a estabilizar y empaclar el producto, y termina cuando el cliente acepta el producto como terminado. Un aspecto significativo de esta fase es que el cliente y el usuario inician pruebas significativas del producto.

Durante la fase de estabilización, el equipo puede empaclar múltiples liberaciones de productos externos mientras se conduce hacia la liberación final del producto. Durante estas liberaciones intermedias, el equipo se enfocará primariamente en las pruebas para identificar errores y corregirlos.

Cuando el equipo se está acercando a la liberación del producto se conoce como liberación cero errores (ZBR, zero-bug release). ZBR es el primer producto en el que todos los errores activos han sido resueltos de alguna manera, ya sea corregidos, pospuestos o considerados sin importancia, una vez que la liberación intermedia es empaquetada, pruebas extensivas deben ocurrir para determinar cuando está listo el producto, en que punto el equipo puede declarar la liberación como el producto final.

Cuando el equipo piensa que el producto está potencialmente listo para la liberación final, se construye una liberación candidata, cada una de estas liberaciones incluye todos los elementos requeridos en un producto empaquetado y no tiene errores activos. El equipo ejecutará pruebas altamente intensivas contra esta liberación para determinar si llegará a ser la liberación producto final o si el equipo debe generar una nueva versión candidata con las correcciones apropiadas. No es común que la primera liberación candidata sea aquella que se empaque por los errores que se llegan a encontrar.

La liberación de producto final es la liberación candidata que todos los usuarios claves, miembros del equipo y clientes han acordado como la versión que será empaquetada, es decir ya incluida en la caja y sobre la que ya no se desarrollará más ni se harán más pruebas.

Determinar que liberación es la liberación para producto final es una decisión difícil de tomar, al final la respuesta consiste en empaquetar el producto correcto en el tiempo justo. ¿Este producto cumple las necesidades del cliente?, otras consideraciones claves para la liberación son el reporte de errores, los resultados de las pruebas de la liberación candidata, y el soporte del producto, como la mayoría de las grandes decisiones, existen muchos riesgos inherente pero la decisión será hecha como un esfuerzo del equipo.

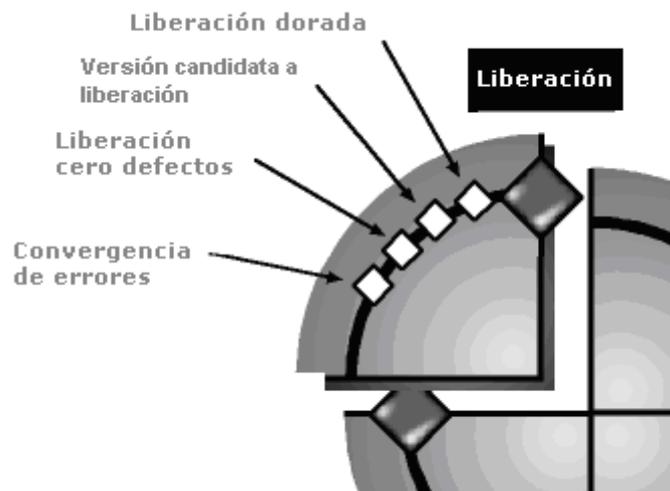


Figura 2.7 Las etapas de la fase de aseguramiento de calidad.

Al nivel del proyecto, la liberación significa que el equipo ha cumplido con la visión a nivel proyecto del producto, Esta meta es también un momento para celebrar (la meta ha sido alcanzada).

Cuando la fase de estabilización está terminada, el equipo del proyecto alcanza la meta de la liberación, los entregables de esta etapa proveen información valiosa para la implementación y uso de la aplicación en el ambiente de producción.

Los seis entregables requeridos para alcanzar la meta de la liberación son:

- Liberación de producto final, el código fuente y ejecutables, como se describió anteriormente.
- Notas de liberación del producto, documentos que contienen la información de la liberación y los cambios de última hora.

- Materiales de usuario y soporte, la información de soporte de la liberación final, estos materiales pueden tomar muchas formas tales como archivos de ayuda, asistentes, guías de cómo-hacer y materiales de entrenamiento, esto tiene dos finalidades, la de preparar a la comunidad de usuarios para la liberación del producto y la otra asistir a la comunidad de usuarios después de la liberación.
- Resultados de las pruebas, el estado de errores y la base de datos para proyectos futuros. Estos resultados pueden ser usados para determinar nuevas funcionalidades para nuevas versiones del producto, al igual que para proporcionar ayuda a los equipos de mantenimiento y soporte.
- Archivos del proyecto, toda la información relevante del producto creada durante el proceso de desarrollo, sin importar si no fue empaquetada con el producto.
- Documentos del proyecto, la documentación de todas las etapas del proyecto, incluyendo los entregables.

Administración de riesgos bajo el modelo de procesos de MSF

La naturaleza interactiva del Modelo de Procesos de MSF fomenta la evaluación y manejo de los riesgos en forma, continua durante todo el proyecto. Esto se traduce a su vez en un nivel de calidad alto y consistente a lo largo de la ejecución del proyecto.

Como riesgo se define cualquier situación que amenaza el éxito del proyecto en general o afecta la capacidad del equipo de trabajo para implantar la visión y el alcance del mismo. Los riesgos no solamente incluyen la imposibilidad de ejecución de cualquier fase del proyecto, sino también la imposibilidad de ejecutar con relación al nivel de servicio esperado o deseado.

Ya que los factores de riesgo pueden no conocerse cuando se realiza la planificación del proyecto, la identificación de éstos se convierte en un factor crítico de éxito para el equipo del proyecto.

El riesgo se maneja en el Modelo de Procesos de MSF mediante programación de tareas basada en prioridades de acuerdo a los riesgos, en la que aquellos componentes de más alto riesgo se implementan o desarrollan primero

El análisis del impacto de los riesgos tiene como objetivo balancear el costo de mitigación o contención del riesgo contra el costo de asumir el riesgo si éste se produce, dentro de un esquema que tenga el menor impacto posible sobre la ejecución misma del proyecto.

Todas las metodologías de manejo de proyectos proveen alguna aproximación a la administración de los riesgos.

El marco de trabajo de Microsoft Solutions Framework, considera el adecuado manejo de riesgos como una de las condiciones más importantes para el éxito de un proyecto.

Entendemos por riesgo, a toda aquella situación que pueda poner en peligro la terminación exitosa de un proyecto en términos de entregables, costos o tiempos.

Debe tenerse en cuenta que un riesgo no es algo “malo” o “bueno” intrínsecamente, es una situación que puede darse en el marco de cualquier proyecto. Asimismo, la asignación de un responsable no implica responsabilidad sobre la ocurrencia de dicho riesgo, sino sobre la administración de los planes de mitigación y/o contingencia que se definan.

Las acciones básicas que denotan una administración proactiva de los riesgos se pueden ejemplificar a través de la siguiente tabla:

Anticipar los problemas	Vs.	Solucionarlos cuando ocurren
Atacar las causas de origen	Vs.	Atacar los síntomas
Prevenir y minimizar el riesgo	Vs.	Reaccionar ante las consecuencias
Prepararse para las consecuencias	Vs.	Reaccionar a la crisis para minimizar el impacto
Utilizar un proceso comunicado y estructurado	Vs.	Utilizar un proceso ad-hoc

Todo riesgo debe poder ser definido en dos aspectos:

- Condición
- Consecuencia

La imposibilidad de definir cualquiera de los dos aspectos determina que el equipo de trabajo debe reconocer el análisis de ese riesgo en particular, ya que el mismo podría no ser tal o bien estar enmascarado por otros riesgos que de no ser identificados adecuadamente podrían presentarse sin que el equipo tenga planes para responder ante ellos. La adecuada definición de condición y consecuencia permite también detectar situaciones atribuibles a problemas de raíz o estructurales.

Una vez que un riesgo está detectado y definido adecuadamente se recomienda definir dos planes de acción:

Plan de mitigación:

El plan de mitigación apunta a disminuir la probabilidad de ocurrencia de un riesgo atacando las condiciones que lo generan.

Plan de contingencia:

El plan de contingencia consiste en la serie de acciones a tomar en el caso que la consecuencia del riesgo se presente. A efectos de definir la aplicación del plan de contingencia, se debe definir un “trigger” que determinará la efectiva aparición de la consecuencia de manera unívoca.

Durante el transcurso de un proyecto pueden detectarse gran cantidad de riesgos y no resulta eficiente en cuanto a los tiempos, desarrollar planes de administración para todos los riesgos detectados. Para poder calificar y cuantificar adecuadamente los riesgos, MSF propone determinar:

- Probabilidad de ocurrencia del riesgo
- Impacto generado de efectivizarse la consecuencia del riesgo

Suele ser una situación común que se ve en los análisis de riesgos, la priorización de aquellos riesgos de mayor impacto o probabilidad; y no necesariamente esta escala determina el orden de prioridad más adecuado.

Para priorizar adecuadamente los riesgos, se apela a la variable “Exposición”, este valor se obtiene por el producto entre la probabilidad y el impacto de un riesgo dado.

Las escalas en las que se definen probabilidad e impacto son arbitrarias, siendo único requisito mantenerlas constantes en todos los riesgos.

Al obtener un cuantificador normalizado, se puede priorizar adecuadamente y seleccionar los cinco o diez riesgos de mayor exposición para trabajar con ellos. Cada riesgo definido en la lista debe tener asociado también a un responsable y una fecha de revisión.

El proceso de revisión de riesgos se realizará a efectos de determinar si los planes de mitigación fueron exitosos y de esa forma es posible que se retire un riesgo de la lista dado que podría bajar su probabilidad, ingresando un nuevo riesgo a la lista.

Según los puntos expuestos, las propiedades de cada riesgo a administrar serán:

- Condición
- Consecuencia
- Probabilidad
- Impacto
- Exposición
- Responsable
- Plan de mitigación
- Plan de contingencia
- Fecha de revisión

Herramientas

II.3 UML (Unified Modeling Language)

Uno de los conceptos iniciales detrás de UML fue poner fin a la “Guerra de las metodologías” dentro de la comunidad orientada a objetos, la orientación a objetos se popularizó hasta finales de los años 80’s con los lenguajes de programación C++ y Smalltalk.

Cuando la programación orientada a objetos tuvo éxito, la necesidad de metodologías que soportaran el desarrollo de software continuó, algunas de las metodologías orientadas a objetos que fueron populares a principios de los años 90's son:

- Booch, definió la notación donde un sistema es analizado por medio de varias vistas, donde cada vista es descrita por medio de diagramas de modelo.
- OMT (Object Modeling Technique), es una metodología desarrollada en General Electric, donde previamente trabajo James Rumbaugh y se refiere a un proceso para pruebas basado en una especificación de requerimientos. El sistema es descrito por medio de modelos: el de objetos, el dinámico, el funcional y el de casos de uso, los que se complementan unos a otros para dar una descripción completa del sistema.
- OOSE/Objectory, ambas metodologías construidas bajo el mismo punto de vista básico por Ivar Jacobson. La metodología OOSE es la visión de Jacobson de una metodología orientada a objetos, la metodología Objectory es usada para la ingeniería de negocios, donde las ideas son usadas para modelar y mejorar procesos de negocio. Ambas metodologías se basan en casos de uso que son empleados en todas las etapas del desarrollo para verificar el sistema.
- Coad/Yourdon, es también conocida como OOA/OOD, fue una de las primeras metodologías usadas para el análisis y diseño orientado a objetos, esta metodología a diferencia de las demás era más simple y fácil de aprender. Sin embargo, la notación y la metodología no podían escalarse a cualquier cosa sino a sistemas muy limitados.

Cada una de estas metodologías tiene su propia notación, procesos y herramientas, esto hizo la selección de una metodología una decisión muy importante y frecuentemente genera discusiones acaloradas y debates acerca de que metodología fue "la mejor", "más avanzada", y "la correcta" para usarse en un proyecto específico, aunque raramente hay una buena respuesta, porque todas las metodologías tienen sus propias fortalezas y debilidades.

Usar un lenguaje de modelado tan complicado y extenso como UML requiere del apoyo de herramientas. Aún si los primeros borradores de un modelo, son hechos usando un pizarrón (dibujando los modelos manualmente), el trabajo de mantenimiento, sincronización y consistente en un número importante de diagramas es casi imposible sin una herramienta.

El uso de MS Visio, nos permitió contar con una herramienta para dibujar los diagramas, almacenar la información en general en un repositorio común, generar los esqueletos de código a partir de los modelos y documentación.

De UML utilizaremos sólo una parte mínima de su capacidad real, ya que MS Visio es la herramienta que se utilizará para generar los esqueletos de código. Esta parte básicamente está dirigida a los componentes.

La vista de componentes

Es una descripción de la implementación de los módulos y sus dependencias. Esta vista es principalmente para los desarrolladores y consiste en el diagrama de componentes.

Los componentes son diferentes tipos de módulos de código que son mostrados con su estructura y dependencias.

Diagrama de componentes

Un diagrama de componentes muestra la estructura física del código en términos de componentes de código. Un componente puede ser de código fuente, binario, o un ejecutable. Un componente contiene información acerca de la clase o clases que están implementadas, creando un mapeo de la vista lógica a la vista de componentes. Las dependencias entre los componentes están indicadas, haciendo fácil analizar como son afectados otros componentes por el cambio en alguno de ellos.

Los componentes son la implementación en la arquitectura física de los conceptos y la funcionalidad definida en la arquitectura lógica (clases, objetos, sus relaciones y colaboración).

Un componente de código fuente es relevante en tiempo de compilación, es un archivo de código fuente que implanta una o más clases.

Un componente binario es código objeto que es el resultado de compilar un componente fuente. Este puede ser un archivo de código objeto, un archivo de librerías estáticas, o un archivo de librerías dinámicas. Un componente binario es relevante en tiempo de ligado, en el caso de una librería dinámica, en tiempo de ejecución (una librería dinámica es cargada por el componente ejecutable en tiempo de ejecución).

Un componente ejecutable es un archivo de programa ejecutable que es el resultado de ligar todos los componentes binarios (ya sean estáticos en tiempo de ligado o dinámicos en tiempo de ejecución). Un componente ejecutable representa la unidad ejecutable que es corrida por un procesador.

Un componente es representado en UML como un rectángulo con una elipse y dos rectángulos más pequeños ubicados a la izquierda (éste es el símbolo usado en el metodología de Booch para representar un módulo). El nombre del componente es escrito abajo del símbolo o dentro del rectángulo más grande.

Una conexión de dependencia entre componentes es indicada por una línea punteada con una flecha abierta, significa que un componente necesita de otro para ser capaz de tener una definición completa. Una dependencia entre un componente de código fuente A con otro componente B significa que existe dependencia específica de lenguaje de A a B.

En un lenguaje compilado, pudiera significar que un cambio en B requerirá una recompilación de A, porque las definiciones del componente B son usadas cuando se compila A.

Si los componentes son ejecutables, las conexiones de dependencia pueden ser usadas para identificar cuáles librerías dinámicas son necesarias para que un programa ejecutable sea capaz de correr.

Un componente puede definir interfaces que son visibles a otros componentes, una interfaz es representada como una línea desde el componente con un círculo al final. El nombre de la interfaz es colocado cerca del círculo. Dependencias entre componentes pueden entonces apuntar a la interfaz siendo usada.

Conclusiones

En todo proyecto es muy importante seguir una metodología, la cual no siempre se puede seguir de inicio a fin pero lo más importante es seguir una serie de pasos que permitan controlar y finalizar un proyecto.

Todas las metodologías son buenas, lo más importante es documentar aquellas etapas y actividades que permiten identificar cuál es la necesidad y requerimientos del negocio para no tener dificultades con las expectativas generadas, ya que éstas pueden hacer que un proyecto fracase, que no se logre terminar o que los costos provoquen una pérdida para cualquiera de las partes involucradas.

El uso de herramientas para seguir alguna metodología no siempre garantiza que se logre una mejor calidad o mayor aceptación por parte del cliente, en algunos casos puede complicar generar la documentación del proyecto, que termina siendo inoperable además de consumir tiempo y recursos valiosos del mismo.

Capítulo III. Planeación y análisis

Objetivo del capítulo:

Describir el proceso de planeación y análisis a realizar para la construcción del sistema motivo de esta tesis, tomando como guía la metodología seleccionada y descrita en el capítulo anterior.

- **Planeación:** El objetivo de esta fase es compartir entre los miembros del equipo y el cliente, una visión clara de los objetivos y metas del proyecto a través de un documento donde se detallarán los siguientes elementos: Visión y objetivos, alcance del proyecto, riesgos y estructura del equipo, véase figura 3.1.
- **Análisis:** El objetivo de esta fase es transcribir lo investigado del proceso de negocio actual a automatizar y los requerimientos de usuario en documentos descriptivos de la funcionalidad que cubrirá el sistema, así como la selección de la plataforma tecnológica que permita cumplir con la visión y objetivos del sistema.

Con la finalidad de ilustrar el proceso a ejecutar en esta fase y en general de todas las fases descritas es esta tesis se presenta el caso de uno de los módulos más representativos del sistema siendo éste el de “Administración de Oportunidades”.

III.1 Planeación

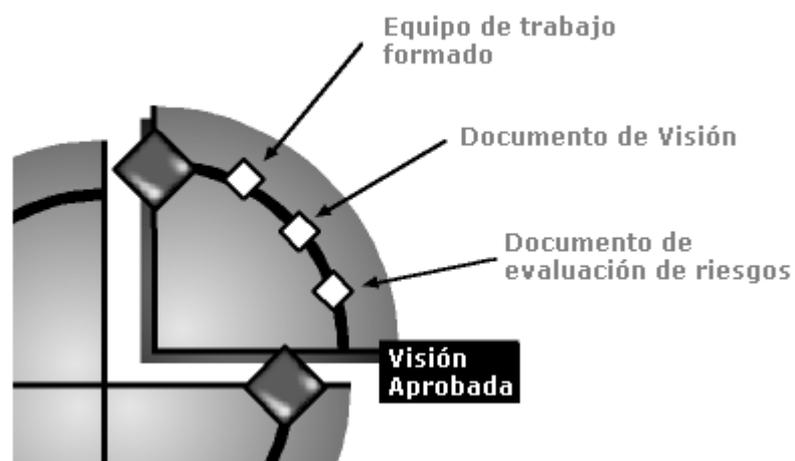


Figura 3.1 Etapa de Planeación

III.1.1 Estructura del equipo de trabajo

Esta actividad permitirá identificar las características y roles de cada uno de los individuos que participarán en la construcción del proyecto. El número de recursos dependerá del tiempo planeado para la terminación del sistema, el que deberá establecerse en forma definitiva en la etapa de diseño.

La metodología plantea un modelo en donde las actividades de cada uno de los roles tienen una secuencia definida para cada fase, a través de flujos de trabajo circulares que permiten la liberación parcial y gradual de módulos terminados.

La intención es optimizar los recursos cuidando el costo del proyecto, y evitando círculos viciosos que alarguen la duración del proyecto a causa de errores al perder la visión y propósito de cada uno de los componentes del sistema.

Para la elaboración del sistema se contemplan los roles y números de recursos para cada uno de ellos, respecto a un planteamiento inicial de 9.5 meses para la terminación del proyecto, mostrados en la figura 3.2.

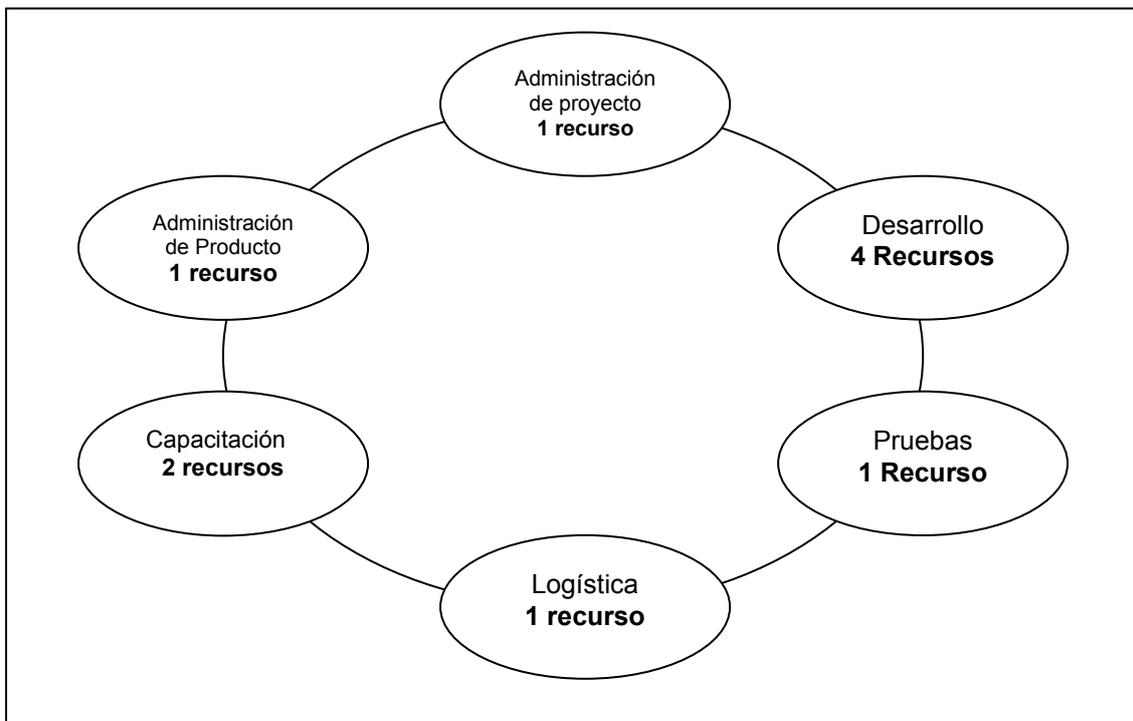


Figura 3.2 Roles y número de recursos

Administración de Producto

Este rol asegurará de que la visión del producto se cumpla buscando en cada fase comunicarla a cada miembro del equipo, evitando, detectando y corrigiendo desviaciones de la misma.

Administración de Proyecto

Este rol es el dueño del proceso de desarrollo y es el responsable de asegurar de que el producto cumpla con los requerimientos especificados y cumplir en tiempo y forma del plan de trabajo. Coordinando y sincronizando los esfuerzos de los desarrolladores optimizando los recursos en la medida de lo posible.

Desarrollo

Este rol se encarga de la implementación del proyecto cubriendo las capacidades técnicas requeridas, en donde debe combinar sus habilidades para encontrar soluciones a los procesos de negocio especificados en la etapa de análisis.

Pruebas

Este rol es responsable de detectar cualquier desviación entre los requerimientos funcionales y el sistema implementado, por lo que debe conocer tanto los procesos de negocio como el funcionamiento del sistema. Una función muy importante es retroalimentar con claridad los cambios a realizar por parte de los desarrolladores, así como la validación con el administrador de producto y proyecto.

Capacitación

Este rol es responsable de asegurar el proceso de adopción del producto siempre buscando que sea amigable, intuitivo y fácil de usar. Deberá conocer perfectamente el funcionamiento del sistema para poder crear el plan de capacitación tomando en cuenta las habilidades de cada grupo de usuarios.

III.1.2 Visión del Proyecto

El sistema a construir permitirá automatizar las áreas de ventas de una empresa dedicada a la comercialización de productos a través de ejecutivos o representantes de ventas que se encuentran atendiendo clientes en diferentes zonas geográficas, así como permitir que los usuarios en oficinas centrales consoliden la información, se procese en el sistema de gestión (ERP) y también la generen para su transmisión a dichos ejecutivos.

Por lo anterior el sistema deberá operar a través de un esquema remoto y central, en que cada empleado de la fuerza de ventas (Cliente Remoto) se constituya como un asesor de negocios, todo esto a través de un modelo de colaboración y flujos de trabajo, además de consolidar e integrar la información generada en una base de datos central (Oficinas Centrales) y unificada para su adecuada administración, con la filosofía de ser una base de conocimiento del cliente.

Permitirá a los ejecutivos de ventas administrar eficientemente las relaciones con sus clientes, intercambiar información en ambos sentidos y entre las diferentes áreas que conforman una área de ventas, agilizando la captura de información del negocio desde los clientes remotos (punto de venta) para consolidarla en un sistema de información central y su integración al sistema empresarial (ERP), así como brindar información actualizada y en tiempo, resultado de la actividad diaria realizada con el cliente en los puntos remotos.

Será un sistema integral que le ayudará en la administración de las relaciones con los clientes de la empresa donde también se podrá utilizar, para cubrir las necesidades de ventas, servicio al cliente, así como el seguimiento de estas actividades.

Dichos sistema deberá construirse como un sistema modular debido a los continuos cambios en reglas y políticas del negocio que se presentan en un área tan dinámica como ventas, siendo ésta una de las principales características del sistema a implementar.

Por lo anterior el sistema deberá permitir la transmisión de la información de las dos formas siguientes.

Red Local o Wan Privada (VPN)

Este esquema permitirá la transmisión de información en la red local de oficinas centrales, así como también para cualquier nodo comunicado a través de una red WAN sea física o virtual (Virtual Private Network). Los clientes conectados en este esquema realizarán las transacciones en línea con la Base de Datos Central.

Dial- Up

Este esquema permitirá la transmisión de información entre los Ejecutivos de Ventas (Cliente Remoto) y Oficinas Centrales (Base de Datos Central), utilizando líneas telefónicas públicas o privadas (VPN).

Los clientes conectados en este esquema realizarán las transacciones fuera de línea dado que depositarán y obtendrán la información en un arreglo de directorios, los que residen también en un servidor central destinado para este propósito.

Como requerimiento del sistema se tiene la necesidad de practicar un esquema Web como visión futura del sistema, el cual es un esquema orientado a los clientes finales, para brindar las herramientas necesarias para que puedan hacer sus pedidos en forma directa y revisar la situación de los mismos en tiempo real.

Con base a visión del sistema y a los requerimientos planteados se requiere una plataforma tecnológica que permita soportar los procesos remotos y transaccionales, así como una alta flexibilidad para realizar los cambios en las reglas de negocio propias de los procesos que soportará, así como la robustez necesaria, para centralizar la información de forma estructurada y segura, la que será descrita más adelante.

Objetivos a cubrir por el sistema:

La definición de los objetivos es muy importante para la realización de las siguientes actividades de esta fase, ya que cada uno de ellos será la guía para definir las características de cada uno de los módulos que contendrá el sistema.

Objetivos Cualitativos
Fortalecer el control y seguimiento de actividades
Incrementar los niveles de productividad reduciendo tiempos muertos por instancia y transporte de los ejecutivos o representantes de ventas a las oficinas regionales o centrales.
Fortalecer la lealtad de los clientes cautivos otorgando una asesoría especializada fundamentada en el conocimiento histórico de su negocio.
Aumentar la predictibilidad del negocio a través estandarizar del proceso de ventas a través de los porcentajes de probabilidad.
Incrementar el nivel de servicio al cliente, a través de la entrega inmediata del estatus de sus facturas y pedidos.
Mejorar la comunicación y colaboración entre los ejecutivos de ventas y el área de administración de ventas (colaboración).
Estandarizar los procesos, procedimientos y proyecciones de venta.
Generar continuamente una base de conocimientos. (Ej. Procesos de venta, información detallada de los clientes, contactos influenciadores, etc.)
Automatizar y hacer más eficiente la toma de decisiones a través de visores de información consolidada entre los diferentes niveles de la organización.

Objetivos Cuantitativos

Evitando duplicidad de información por captura manual. Eliminando costos directos e indirectos.

Aumentar las ventas a través del seguimiento efectivo del proceso de ventas y la interacción oportuna de los diferentes niveles de la organización y creación de promociones y ofertas mediante el conocimiento del cliente.

Reducir los gastos de paquetería en el envío de papelería, fax, etc.

Reducir los costos de sobreproducción por mala proyección provocada por pedidos duplicados.

Reducir el gasto en llamadas telefónicas a las oficinas centrales y centros de distribución por concepto de consultas.

Fomentar el incremento en ventas al permitir agilizar la transmisión de pedidos varias veces al día, acortando los ciclos de procesamiento.

Reducir los costos como resultado de captura manual de pedidos.

III.1.3 Evaluación de Riesgos

Esta actividad permite identificar las características que deberá cubrir el sistema para evitar el fracaso del mismo. Con base en la identificación de los riesgos deberán tomarse decisiones importantes que dictarán el camino de las siguientes fases, como la selección de la plataforma tecnológica, el modelo de diseño y programación, así como los entregables de cada etapa.

Seguridad

- 1) La pérdida de información confidencial almacenada en el dispositivo móvil, por factores externos e internos.
- 2) Robo de información debido a las conexiones remotas realizadas por la fuerza de ventas.
- 3) Filtrado de información confidencial entre empleados que genere problemas en el clima laboral.
- 4) La pérdida de información durante la transmisión provoca insatisfacción y perder a los clientes.

Adopción

- 1) Resistencia al cambio evitando el uso del sistema para la ejecución de la actividad propia del vendedor debido a la cultura tecnológica de la fuerza de ventas.
- 2) Falta de confianza en la calidad de la información ingresada al ERP (Sistema de Gestión) para disparar procesos de producción relacionados a las órdenes de venta.

Desempeño

- 1) Gasto excesivo por larga duración de las llamadas para transmitir información desde hoteles o lugares públicos.
- 2) Una larga duración del proceso de consolidación de información y carga al sistema empresarial (ERP) puede interferir con otros procesos realizados durante la noche o con la operación diaria afectando la entrega de otros servicios.

Plan de proyecto

Acorde a la visión del proyecto, objetivos cuantitativos y cualitativos, evaluación de riesgos y equipo de trabajo seleccionado es posible plantear la duración del proyecto y de cada etapa. Para el planteamiento y definición de tiempos es importante contemplar la fecha en que la empresa seleccionada requiere el sistema en operación en donde entran factores como el cierre de año fiscal, evento anual en que reúne a su fuerza de ventas, etc.

La etapa de planeación es una inversión conjunta de tiempo y recursos, en la que se define el costo del proyecto, tiempos de entrega y se establecen compromisos de ambas partes contractualmente. Como paso siguiente se establece la selección de los recursos para los roles necesarios y se plantea el plan de proyecto para definir los entregables de cada etapa, véase figura 3.3.

Figura 3.3 Plan de Proyecto

Plan de Trabajo:			
Actividad	Duración	Responsable	Responsable del cliente
Planeación	2 semanas	Administrador de Producto, Administrador de Proyecto	Líder de proyecto, usuarios finales (directores y gerentes)
Análisis	7 semanas		
Levantamiento de requerimientos	3 semanas	Administrador de Producto, Administrador de Proyecto	Líder de proyecto, directores, gerentes, usuarios expertos
Definición de casos de uso	3 semanas	Administrador de Producto, Administrador de Proyectos	
Revisión final y aceptación formal	1 semana	Administrador de Producto, Administrador de Proyecto	Líder de proyecto, usuarios finales (directores y gerentes)
Diseño	7 semanas	Administrador de Producto, Administrador de Proyecto, Desarrollador	Líder de proyecto
Desarrollo	16 semanas	Administrador de Producto, Administrador de Proyecto, Equipo de desarrollo	
Aseguramiento de calidad	4 semanas	Administrador de Producto, Administración de proyecto, Equipo de pruebas y capacitación	Líder de proyecto
Implantación y liberación a producción	4 semanas	Administrador de Producto, Administrador de Proyecto, Desarrollador, Equipo de capacitación y pruebas,	Líder de proyecto, usuarios finales (directores y gerentes), 1 usuario expertos por rol definido
Semanas Totales	40	Meses Totales	10

III.2 Análisis

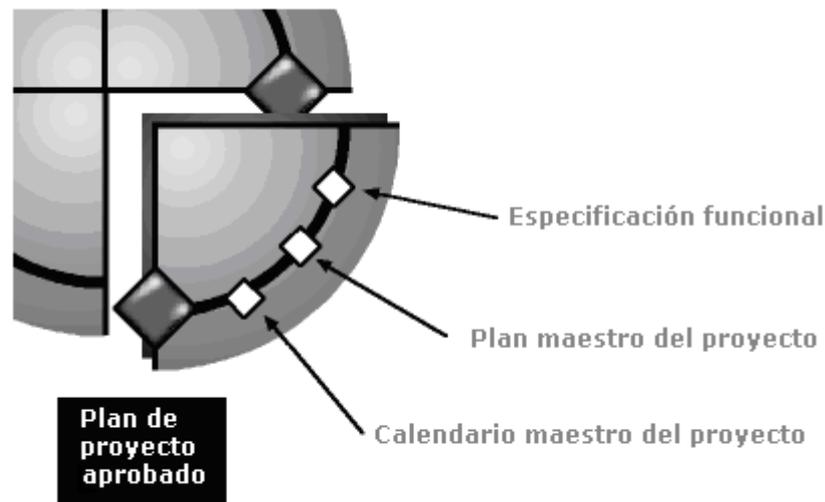


Figura 3.4 Etapa de Análisis

III.2.1 Definición de Requerimientos

Esta etapa tiene gran relevancia para el éxito del proyecto ya que plasma en un documento los procesos de negocio a automatizar y debe establecer la estructura y composición del mismo.

En esta etapa el trabajo en conjunto con cada uno de los usuarios para generar la definición de requerimientos es intenso, ya que se deben adquirir compromisos de ambas partes que serán formalizados a través de un documento final aprobado por ambas partes.

En esta actividad se realizaron reuniones con los usuarios responsables de la definición de cada uno de los procesos a automatizar por el sistema y de la definición de requerimientos, para lo que se programaron diferentes sesiones, tomando como guía la visión y objetivos planteados inicialmente.

A continuación se muestra como ejemplo un fragmento de la documentación firmada por el cliente para el módulo de "Administración de Oportunidades", el que será el utilizado para ilustrar cada una de las etapas de la metodología y documentos entregables en esta tesis.

Módulo de Administración de Oportunidades

Este módulo, otorgará la funcionalidad mencionada a continuación:

- Permitirá al Usuario (Ejecutivo de Ventas) negociar con el Cliente con base en la información presentada en el histórico de oportunidades (información consolidada y obtenida del ERP) y de las oportunidades abiertas, que serán utilizadas para generar Pedidos.
- Permitirá al Usuario (Ejecutivo de Ventas) capturar las oportunidades de venta efectuadas con cada Cliente. Los Pedidos pueden ser generados a partir de las oportunidades capturadas en el proceso de ventas.
- Permitirá al Usuario (Ejecutivo de Ventas) indicar los productos en promoción (sin costo) otorgadas al Cliente, estos productos de obsequio deben ser reflejadas en la oportunidad de venta.
- Permitirá al Usuario (Ejecutivo de Ventas) capturar notas referentes a la oportunidad de venta, estas notas son almacenadas para el registro de la actividad realizada en la Base de Datos Central.
- A partir de las oportunidades que conforman la proyección de venta se podrá tener visibilidad del estimado de producción de aquellas oportunidades con un porcentaje de probabilidad del 60%.

Procedimiento de Captura de un Pedido:

1. Seleccionar la opción de “Administración de Oportunidades” del menú principal de Visitas al Cliente.
2. Se presentará una pantalla con la siguiente información:
 - Datos del Cliente
 - Nombre del Cliente
 - Empresas relacionadas (sucursales
 - Contactos relacionados)
 - Datos de la Oportunidad
 - No. de la Oportunidad
 - Nombre de la Oportunidad
 - Porcentaje de probabilidad de cierre
 - Fecha de Cierre
 - Fecha Prometida
 - Detalle del Partida
 - Código del Producto
 - Descripción
 - Cantidad
 - Promoción (Si, No)
 - Precio del producto

- Importes
 - Subtotal
 - IVA
 - Total
 - El Sistema mostrará el catálogo de productos para su búsqueda y selección para la oportunidad.
3. Capturar la fecha de cierre de la oportunidad y el porcentaje de avance, con lo que el sistema brindará nuevas opciones en los casos siguientes:
- Porcentaje de probabilidad 80%: El sistema presentará en forma automática la opción de impresión de pedido, guardando la información con los campos necesarios para el registro de un pedido.
 - Porcentaje de probabilidad 100%: El sistema lo cambia en forma automática una vez que el pedido fue transmitido al sistema central, operado en el ERP y replicado nuevamente al sistema remoto, para ser desplegado en la sección de “Oportunidades de Venta Cerradas”
4. Seleccionar Guardar y Salir.

Consideraciones Generales:

- La información capturada en este módulo será enviada a Oficinas Centrales para su consolidación en la Base de Datos Central del sistema
- La información será integrada en el Sistema de Gestión (ERP) utilizando la interfaz generada para dicho propósito.
- El procesamiento de los Pedidos Generados (Oportunidad al 80%) se realizará en el Sistema de Gestión (ERP) en donde se generarán las facturas que posteriormente serán transmitidas al equipo portátil y que se mostrarán en la vista de “Oportunidades Cerradas”.

Actores:

- **Ejecutivo de Ventas (Vendedor de campo)**

Funciones:

- Realizar la preparación de la cita con el cliente analizando el comportamiento de compra en la vista de oportunidades cerradas.
- Capturar la información de nuevas oportunidades.
- Realizar el registro del seguimiento de cada visita en la sección de notas.
- Generar pedido impreso (Oportunidad al 80%)
- Transmitir la información a Oficinas Centrales, para actualizar información en su equipo portátil y para enviar las operaciones realizadas.

- **Gerente de Zona/Territorio**
 - Realizar consultas de la actividad de cada ejecutivo de ventas de su zona/territorio a cargo.
 - Agregar comentarios e instrucciones para el ejecutivo de ventas a través del ingreso de notas.
 - Aprobar pedidos generados que contengan producto en promoción.
 - Trasmirir la información a Oficinas Centrales, para actualizar información en su equipo portátil y para enviar las operaciones realizadas.

Selección de la plataforma tecnológica

El objetivo de esta actividad es seleccionar la plataforma tecnológica para la implementación del sistema objeto de esta tesis. Incluyendo herramientas de diseño y desarrollo, arquitectura de red y comunicaciones con base a las necesidades de negocio planteadas en la visión y objetivos en la fase de planeación.

III.2.2 Arquitectura de Desarrollo

La arquitectura de desarrollo del sistema se planea como una arquitectura de tres capas, fundamentada en una programación orientada al objeto haciendo uso del modelo COM (Component Object Model), la que se describe a continuación:

Arquitectura de tres capas

Como modelo predecesor a este tipo de arquitecturas se tiene el modelo Cliente – Servidor el que corresponde a una arquitectura de dos capas en donde se maneja que el cliente procese, presente y maneje los datos y procesos funcionales de la aplicación conocida como clientes pesados (fat client).

En este tipo de aplicaciones se presentan rupturas al momento de tener un número grande de usuarios dado que cada cliente se conecta a los servidores de datos ocasionando perdida de escalabilidad dada la limitante en el número de conexiones disponibles, así como grandes limitantes en la reutilización de la funcionalidad dado que la aplicación residente en los clientes está ligada íntimamente a los formatos de la base de datos, siendo poco flexibles.

Por las razones citadas anteriormente y los requerimientos planteados para la implementación del sistema solicitado se seleccionó una arquitectura de tres capas, la que nos dará la capacidad de construir un sistema abierto a cambios dinámicos, constituyendo la mejor solución.

Este tipo de arquitectura se plantea como un marco de trabajo que permite construir sistemas escalables, flexibles, seguros y distribuidos que pueden funcionar en diversos modelos de red en esquemas remotos o locales. También se ha planteado como un modelo de programación que integra el mundo de los sistemas Cliente – Servidor y los basados en Internet (Windows DNA).

En este modelo también conocido como Multi – capas (N-Tier), la escalabilidad y la reutilización son características altamente significativas, lo que es posible dado que la capa de presentación (usuario), reglas de negocio (negocio) y datos están lógicamente separados como se ilustra en la figura 3.5:

El siguiente diagrama muestra la arquitectura tres capas:

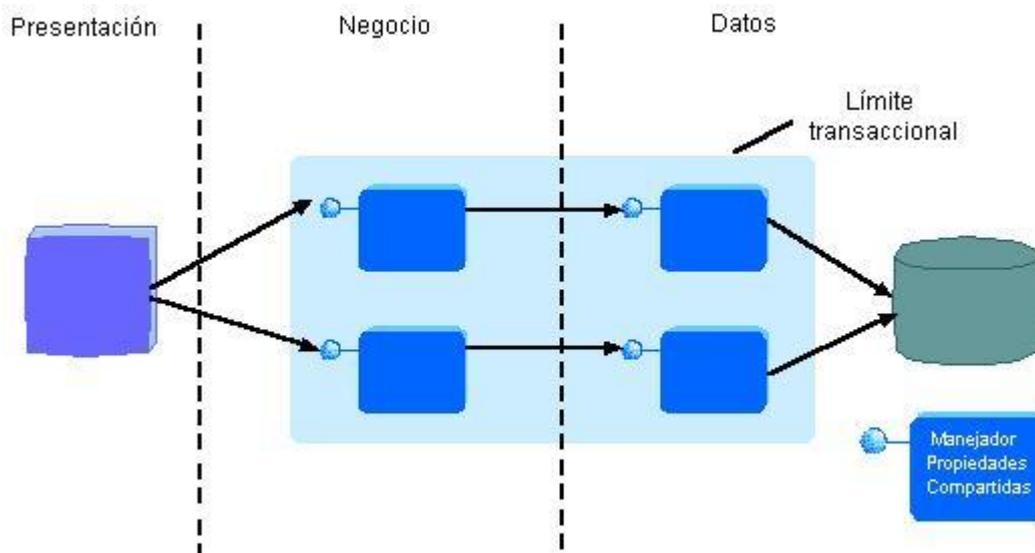


Figura 3.5 Arquitectura tres capas

A continuación se describen las tres capas a utilizar para la implementación del sistema:

➤ **Capa de Presentación:**

Esta capa representa el medio a través del cual el Usuario interactúa con el sistema, presentando los datos al usuario y permitiendo la edición de los mismos.

La implementación de esta capa depende de la plataforma utilizada, dado que se podrían clasificar en dos tipos principalmente:

- Interfaces nativas (PC): Son instaladas del lado del cliente y hacen uso de servicios del sistema operativo por ejemplo en el caso del sistema objeto de esta tesis se utilizarán controles de Windows, lo cuales hacen uso del Win32 API.
- Interfaces Web: Están fundamentadas en HTML y actualmente en XML (Extensible Markup Language), las que son expuestas a través de un Navegador de Web y son portables a cualquier plataforma.

Para el caso del sistema se plantean los siguientes requerimientos, respecto a esta capa:

- Interfaz gráfica poderosa, en cuanto al manejo y desplegado de la información, para el uso de matrices de información para la presentación y captura ágil de la información.
- Intuitiva y de fácil manejo para el Usuario, el cual no tiene experiencia en el manejo de sistemas.
- Manejo de altos volúmenes de información tanto para consulta como para edición.
- Permitir la operación del sistema en forma remota y sin dependencia a una conexión al servidor central.

Con base a las necesidades expuestas como requerimiento se propuso que el sistema no sea una aplicación orientada al Web, si no una aplicación de escritorio (Desktop), la que permitirá que los Usuarios remotos (Ejecutivos de Ventas) puedan trabajar en forma local y tengan la funcionalidad completa, teniendo una copia de su base de datos correspondiente, la que actualizarán en un proceso de replicación, misma que se explicará en los siguientes capítulos.

➤ **Capa de Negocios:**

En esta capa se implementarán las reglas de negocio las cuales actúan sobre los datos llevando a cabo los procesos de negocio. Las reglas de negocio pueden ser algoritmos, políticas, validaciones de negocio, etc. Por ejemplo el descuento aplicado a clientes con ventas mayores a 100,000.00 pesos.

Las reglas de negocio ayudarán a asegurar la validación y en ocasiones la integridad de los datos.

Una consideración importante es que las reglas de negocio no deben estar ligadas ni presentar dependencia a un tipo de cliente específico, es decir no deben ser parte ni depender de la capa de presentación citada anteriormente.

Las reglas de negocio deberán estar implementadas como módulos independientes que pueden ser utilizados por diferentes aplicaciones las cuales pueden corresponder a distintos clientes (interfaces gráficas) sean tipo Web o de escritorio. En su mayoría estos módulos son almacenados en forma centralizada para su uso a través de múltiples aplicaciones como se ha comentado.

La característica citada anteriormente respecto a la implementación de las reglas de negocio como código independiente representa uno de los principios del desarrollo basado en componentes COM (Component Object Model), el que se describirá más adelante.

➤ **Capa de Datos:**

En esta capa es donde reside la función obtener e ingresar lo datos. Se constituye de los servicios de acceso a datos y también deben implementarse como módulos de código independiente encapsulando el conocimiento de cómo manejar los procedimientos de almacenamiento de la información.

Con base a lo anterior, si el lugar donde se almacenan los datos se mueve o cambia de formato, sólo los servicios de acceso a datos que componen esta capa deberán ser actualizados.

Es importante mencionar que dichos servicios son responsables de la integridad de los datos, sea cual fuere el método de acceso a datos seleccionado, por ejemplo ADO (Active Data Objects) de Microsoft para base de datos relacionales o tecnología OLE DB.

En la mayoría de los casos el almacenaje de datos los constituyen en forma común los DBMS (Database Management Systems) de donde los sistemas introducen los datos en tablas, así como optimizan la obtención de los mismos a través de índices.

La figura 3.6 muestra un esquema del modelo de tres capas.

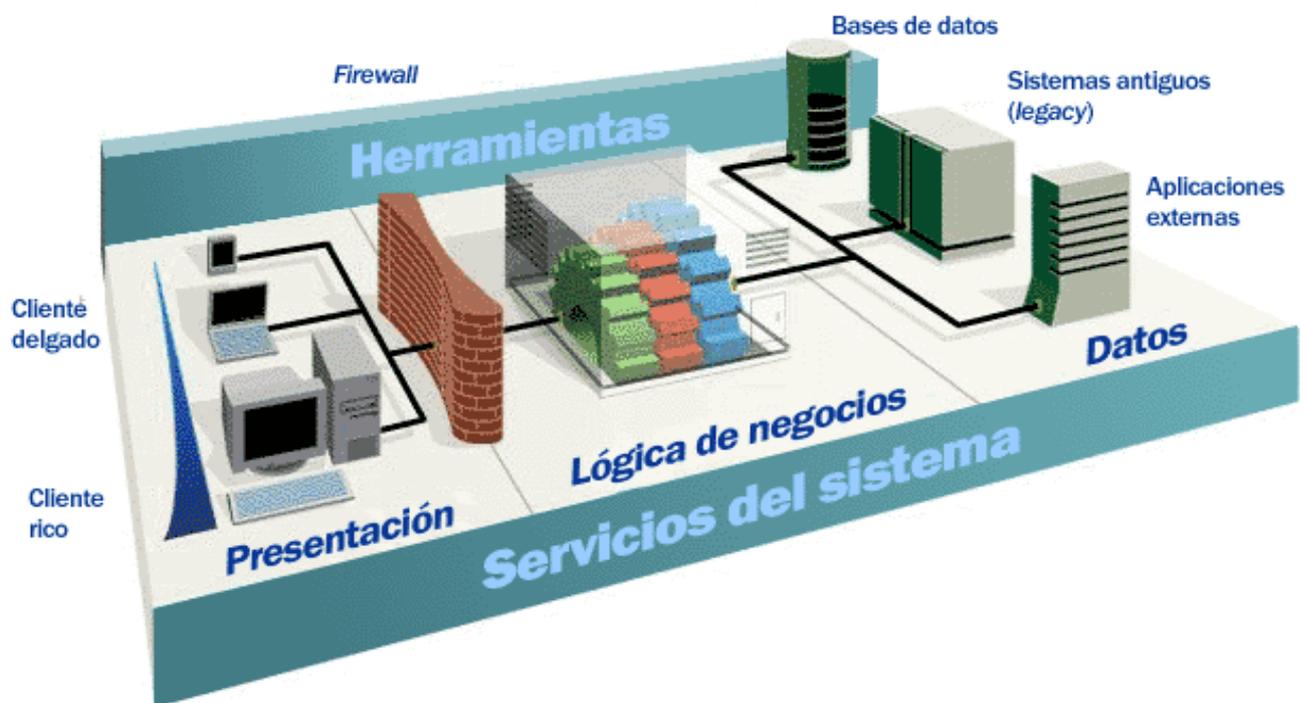


Figura 3.6 Modelo de tres capas

Como parte fundamental para la implementación de la arquitectura propuesta, se seleccionó el modelo COM (Component Object Model) el que se describe a continuación:

COM (Component Object Model)

Es un estándar de arquitectura que permite el desarrollo de componentes de software a través de objetos contenidos en archivos ejecutables que cumplen con un estándar binario, los cuales brindan servicios para aplicaciones, sistemas operativos y otros componentes.

COM brinda una definición estándar de cómo los objetos pueden comunicarse y compartir datos. Los objetos pueden comunicarse a través de un estándar o interfaces personalizadas.

COM es una arquitectura de desarrollo de objetos, la que es independiente del lenguaje de desarrollo, del sistema operativo y de la ubicación física en donde el objeto o componente se encuentre.

Sus características más sobresalientes son:

- Permite la creación de interfaces de usuario de manera gráfica.
- Desarrollo rápido de aplicaciones de alto rendimiento.
- Desarrollo de aplicaciones cliente/servidor o multicapas.
- Creación e integración de componentes reutilizables (componentes COM) y controles ActiveX.
- Soporte para intercambio de datos para otras aplicaciones.
- Soporte para establecer enlaces con Windows y con otras rutinas escritas en otros lenguajes (DLL's).
- Ofrece conectividad a una amplia gama de bases de datos que soporten SQL.
- Es posible optimizar la velocidad y el tamaño de las aplicaciones, así como mejorar el rendimiento de las mismas.

Este modelo se seleccionó sobre otros como CORBA (Common Object Request Broker Architecture), el modelo utilizado en plataforma UNIX. Las herramientas seleccionadas se detallarán en la siguiente sección, así como la plataforma tecnológica con la que contaba el cliente en el que se basó originalmente este desarrollo.

Herramientas seleccionadas

La selección de las herramientas a utilizar se realizó con base a los siguientes factores:

- La experiencia de los desarrolladores con los que se cuenta, en cuanto a herramientas de programación y conocimiento de la plataforma tecnológica.
- El sistema operativo instalado en los equipos de cómputo en donde se utilizaría el sistema, es decir, la planta instalada con la que ya cuenta el cliente.

- El conocimiento de la interfaz gráfica por parte del Usuario final, dado el requerimiento de una rápida adopción de la tecnología y una mejor administración del cambio a través del uso del sistema.
- Que las herramientas de programación permitieran la implementación del modelo de tres capas descrito en el punto anterior.
- El soporte con el que se cuenta por parte del fabricante a nivel del sistema operativo, herramientas de programación y software de terceros entre otros.
- La perspectiva de permanencia de la plataforma seleccionada, así como la evolución de la misma.
- La planta instalada de la plataforma tecnológica (sistema operativo) actualmente en México, lo que facilitaría la adopción del sistema.

Con base a los factores expuestos se listan a continuación las herramientas seleccionadas en donde se describe en forma breve la función de cada una de ellas. Siendo importante comentar que algunas se mencionarán con mayor detalle en los siguientes capítulos de esta tesis.

Herramientas para la administración del proyecto.

Microsoft Solution Framework (MSF).

MSF Es una metodología que incluye modelos que ayudan a guiar a los desarrolladores de software a través de la planeación, construcción y manejo de ciclos incluyendo el manejo de infraestructura de tecnologías distribuidas.

MSF puede ayudar a dirigir los siguientes puntos:

- Cómo manejar los recursos de computación.
- Cómo mejorar el desarrollo, reducir el mantenimiento e incrementar la reutilización de tecnología.
- Cómo modificar usuarios o requerimientos de negocios sin desarrollar un sistema completo.

MSF es una serie de modelos flexibles e interrelacionados que permiten guiar a una organización en el proceso de integración de los recursos, gente y técnicas necesarias para asegurar que la infraestructura tecnológica y soluciones reúnan sus objetivos de negocios.

MSF puede ser implementado solo o en conjunto con otras herramientas y técnicas.

Project.

Es una herramienta que ayuda a la administración de proyectos de desarrollo de software, permite organizar, calendarizar y controlar los avances de las actividades en las diferentes etapas de los proyectos.

Project ofrece características muy avanzadas con las que resulta mucho más sencillo diseñar, administrar y comunicar las tareas a realizar a los diferentes miembros de un equipo, independientemente de cuál sea su experiencia en diseño de proyectos.

Herramientas para el análisis y diseño.

MS Visio.

Es una herramienta de modelado visual de objetos; ayuda a los analistas y diseñadores a plantear sistemas y modelos utilizando notaciones de metodologías orientadas a objetos tales como: Booch, OMT y UML.

Permite la representación de clases, objetos y componentes para el análisis y diseño de sistemas de software complejos.

Utiliza la filosofía de las 4+1 vistas para categorizar y organizar al sistema modelado.

Permite el trabajo en equipo y el desarrollo iterativo incremental, así como también la integración con otras herramientas de ingeniería de requerimientos, control de versiones y aseguramiento de la calidad.

Unified Modeling Language (UML).

Es un lenguaje visual de modelado, compuesto por una serie de símbolos e íconos que permiten modelar sistemas complejos basándose en los conceptos soportados por las metodologías de análisis y diseño orientadas a objetos más populares:

Casos de uso, Actores, Clases, Objetos, Mensajes, Diagramas de transición de estado, Diagramas de secuencia, paquetes estereotipos, etc.

UML es un estándar notacional abierto que no depende de una metodología de análisis específica, teniendo sus orígenes en Rational con los tres metodólogos más importantes (Booch, Jacobson y Rumbaugh).

Herramientas para la programación.

Visual Basic (.Net)

Es un sistema de desarrollo diseñado especialmente para la creación de aplicaciones gráficas en ambiente Windows.

Para soportar este tipo de desarrollos, Visual Basic utiliza fundamentalmente dos herramientas, una que permite desarrollar la interfaz gráfica que será utilizada para

interactuar con los usuarios y la otra es un lenguaje de alto nivel que permite la programación y desarrollo que sustentan la lógica y operación de los módulos de proceso de negocio.

Es un lenguaje de programación orientado a eventos, sin embargo cuenta con la capacidad de desarrollar objetos bajo los estándares de COM.

Las transacciones son llevadas a cabo por objetos o componentes registrados en éste el servicio de componentes de Windows.

Dicho servicio permite que las operaciones realizadas por los componentes sean expuestas a las aplicaciones como interfaces (servicios) que son invocados en forma síncrona o asíncrona, encapsulando en un componente específico cada una de las operaciones a realizar, permitiendo la modularidad del software generado, la fácil modificación o sustitución sin tener la necesidad de efectuar cambios a otras partes de la aplicación.

Esta arquitectura brinda independencia al cliente y al servidor, otorgando alta escalabilidad de los sistemas o aplicaciones que la utilizan.

SQL Server (Versión 2000)

RDBMS de alto rendimiento el cual se integra dentro de la familia de aplicaciones Microsoft BackOffice, bajo una arquitectura Cliente-Servidor, apegado al modelo relacional de Bases de Datos.

Dentro de sus características principales podemos citar:

- Permite el manejo transaccional de los datos mediante SQL (Standard Query Language).
- Interacción con servicios y aplicaciones que hagan uso de Internet.
- Manejo de grandes volúmenes de información, contando con herramientas y comandos para la adecuada administración de la información en sistemas de misión crítica.
- Utiliza modelos de seguridad propios o integrados con Windows NT Server.
- Diseñado con una arquitectura que permite la recuperación gradual de la información
- El establecimiento de estrategias de respaldo acorde a las características de la información.
- Alta escalabilidad, rendimiento y flexibilidad.
- Permite el establecer diferentes esquemas de replicación de Bases de Datos, adecuados a la arquitectura empresarial propuesta.

Infraestructura de cómputo y comunicaciones

La infraestructura de cómputo requerida para la implantación se describe en la figura 3.7 acerca de la que se describirá fundamentalmente el modelo de comunicaciones y el modelo de replicación de datos.

Modelo de Infraestructura

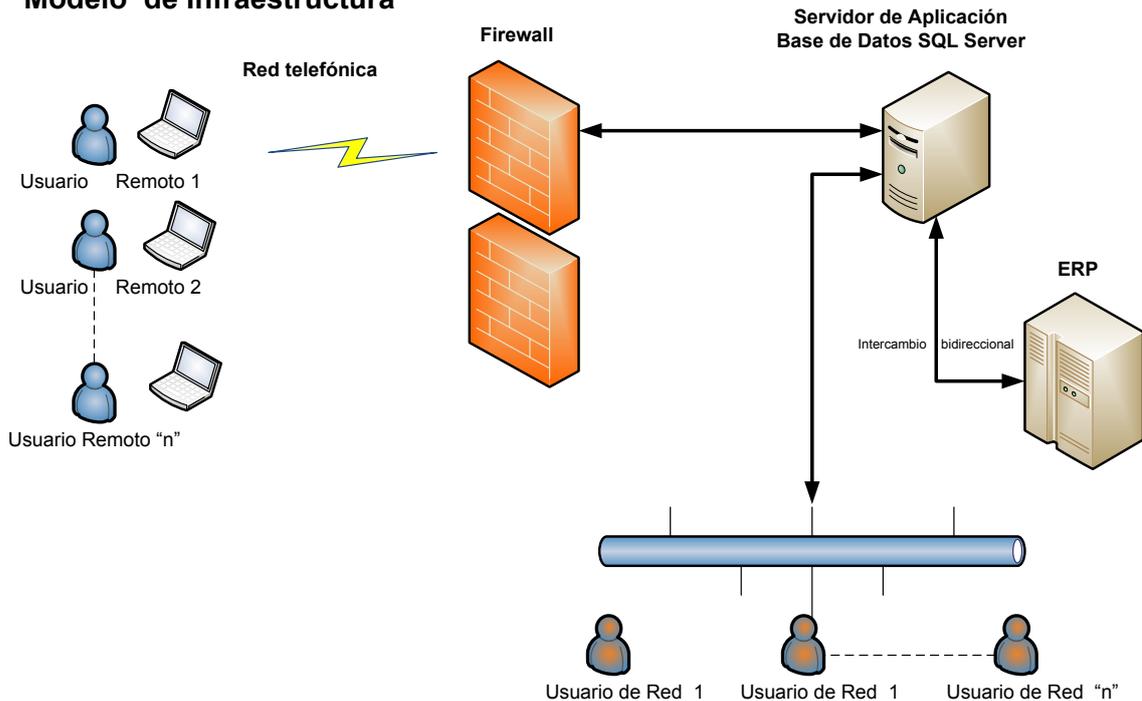


Figura 3.7 Esquema de comunicaciones

Al utilizar esta arquitectura, se pretende que el usuario móvil (vendedor), acceda a los servicios de información de ventas en forma remota. Esto funciona en forma equivalente cuando los usuarios se encuentran en la red local del cliente o en su propia red corporativa.

En este proceso, lo que se busca es que la información se transmita en forma eficiente cuidando los aspectos de seguridad en transferencia de datos. Para ello, lo que se habilita es una transmisión segura a través de una red privada virtual (VPN) que se inicia en la computadora del usuario y se valida en la red local de su corporativo en un servidor de autenticación y validación de usuarios.

El acceso remoto puede ser a través de una llamada por módem o a través de la red local del usuario que tenga salida a Internet. Para los accesos remotos por línea telefónica se ha habilitado un servidor de acceso que contiene un grupo de modems que recibirán las llamadas de los usuarios. Una vez recibida la llamada, se valida el usuario, y se le permite acceso a la red

De esa manera, el usuario puede acceder a los servicios de ventas en forma remota desde cualquier lugar.

Una vez que accedieron a la red corporativa, los usuarios pueden tener acceso a sus buzones individuales, donde es depositada la información que se genera de las bases de información.

De igual manera, los usuarios pueden realizar cambios, modificaciones y actualizaciones en su computadora, de forma tal que al conectarse a la red, pueden actualizar la base de datos con los cambios que realizaron fuera de línea (sin estar en red).

La segunda validación del usuario se dará a nivel de aplicación al registrarse al sistema.

Para empresas que tengan sus comunicaciones subcontratadas con un proveedor de servicios o empresa telefónica, ésta última se encargará de la fase de comunicaciones en cuanto al acceso de usuarios en la red, dejando a la empresa la validación de acceso a sus aplicaciones de cómputo. De esta forma, la empresa de comunicaciones, deberá establecer un acceso seguro (VPN) hacia la red local donde los servidores de aplicaciones residen.

Bajo este modelo de comunicaciones, un sistema de este tipo incrementara su éxito al facilitar las posibilidades de acceso desde cualquier lugar de acceso en el mundo. Esto significa que un vendedor pudiera permanecer la mayor parte del tiempo en su función de ventas, y disminuir el tiempo de gestión administrativa.

En las siguientes etapas de este sistema, se podrán habilitar accesos vía Web (Intranet) para la gestión de procesos operativos en forma remota hacia otros sistemas.

Conclusiones:

En este capítulo se establecen las bases del sistema objeto de esta tesis. Así como las expectativas, visión y definición de los alcances, tiempos de entrega y costo del proyecto, describiendo el propósito y los resultados esperados.

Las actividades realizadas definen las guías a seguir en las siguientes fases, alineando los objetivos de usuarios finales con la visión del sistema.

Esta etapa es crucial para definir los compromisos de ambas partes, definiendo los responsables de cada etapa y que rol jugaran para la adecuada evolución del proyecto. Siendo el momento adecuado para la toma de decisión definitiva si se lleva a cabo el proyecto con la expectativa inicial, se modifica o se desiste del mismo.

Capítulo IV. Diseño y desarrollo

Objetivo del capítulo:

El objetivo de este capítulo es describir las actividades realizadas en la fase de diseño y desarrollo, basadas en la metodología empleada para el desarrollo del sistema, así como el uso de las herramientas que sirvieron de apoyo.

Aquí describiremos a manera de ejemplo algunos de los documentos técnicos que formarán la solución.

IV.1 Fase de diseño

El diseño es una expansión técnica y la adaptación de los resultados del análisis, teniendo como objetivo generar los documentos que serán usados por los programadores en la fase de desarrollo.

Empezaremos con el diseño conceptual de la aplicación, aquí deben quedar ya identificadas las necesidades del negocio, todos los procesos deben estar entendidos, la funcionalidad acordada, lo ideal es tener esta documentación validada y firmada involucrando al cliente, al usuario y a todos los interesados.

IV.1.1 Diseño conceptual

El diseño conceptual no toma en cuenta el enfoque o las tecnologías necesarias para construir la solución. Es la traducción de los requerimientos de negocio y del usuario del lenguaje de los usuarios al lenguaje de los desarrolladores.

Las metas del diseño conceptual son:

- Identificar las necesidades de negocio
- Entender lo que hacen los usuarios y lo que ellos requieren.
- Generar escenarios que reflejan los requerimientos de manera completa y precisa, involucrando al cliente, al usuario y a otros interesados.

Siempre es necesario escribir todos los detalles que se mencionen por parte del cliente y del usuario final, sobre todo de este último ya que es el que ejecuta el proceso de manera regular y puede ayudar al equipo de analistas en aquellos casos raros y excepciones del proceso. Un buen equipo de analistas se llega a involucrar tanto en los procesos de negocio que se convierten en expertos y son piezas claves en las etapas posteriores.

Todos esos detalles se describen como escenarios y pueden usarse documentos que describan los procesos como el extracto que se muestra en el capítulo anterior, diagramas de flujo o casos de uso, esto va a depender del tipo y nivel de conocimientos técnicos del usuario final del cliente. Una meta de esta etapa es contar con la validación y firma de estos documentos generados, por lo que es muy importante que sean fáciles de entender y que agilicen el proceso de validación, documentos demasiado técnicos llegan a complicar esta firma.

IV.1.2 Diseño lógico

El diseño lógico comienza a involucrarse en los detalles de la aplicación que el equipo construirá para satisfacer las necesidades de negocio y los requerimientos del usuario.

Las metas del diseño lógico son:

- Prototipos de interfaz de usuario.
- Un diseño lógico de la base de datos.

Es muy importante tener en la mente del equipo del proyecto la creación de un prototipo el que mostrará las pantallas que usará el usuario final, no es necesario escribir el código de la aplicación para este fin, el prototipo servirá para mostrar la navegación y se trata sólo de un cascaron para validar y tener como resultado final un documento firmado que avala el diseño gráfico, esta actividad es valiosa ya que el tiempo invertido aquí será empleado para generar los materiales de apoyo para soporte a los usuarios finales.

El no considerar el diseño gráfico final puede producir muchos cambios en la programación, el efecto puede sentirse en la etapa final del proyecto cuando se está entregando y el tiempo se vuelve un factor en la liberación a producción, además de ocasionar fricciones entre el equipo de desarrollo y el cliente.

En la figura 4.1 podemos observar la pantalla de “Consulta General y Proyección de Ventas” que nos servirá para indicar la distribución de los campos de captura, así como los procesos que serán cubiertos en esta pantalla.

Información del Cliente

Nombre del Cliente: Cliente DEMO

Dirección: Calle Domicilio # 11

Domicilio dos: Domicilio dos

Ciudad: México CP: 03100

Empresas Relacionadas

- Empresa 1
- Empresa 2
- Empresa 3
- Empresa 4

Contactos Relacionados

Nombre	Teléfono	Correo elec
Rebeca Sánchez	56701202	usu@mail.com
Marco Alfaro	55801102	usu2@mail.com

Fecha Inicio: 26/Nov/2004 Fecha Final: 24/Dic/2004

Oportunidades de Venta Cerradas

No Factura	Monto	Vencimiento Factu
FAC-03933	\$ 400,000.00	27/Nov/2004
FAC-03945	\$ 1,500,000.00	01/Dic/2004

Monto Total: \$ 1,900,000.00

Oportunidades de Venta Abiertas

No Oportunidad	Monto	Fecha Cierre	% Prob
OP49330	\$ 600,000.00	06/Dic/2004	60 %
OP49400	\$ 850,000.00	12/Dic/2004	40%

Monto Total: \$ 1,450,000.00 % Prob: [dropdown]

Nuevo Limpiar Guardar Salir

Figura 4.1 Pantalla de Captura de Pedidos del prototipo.

Empezar a trabajar en el diseño de la base de datos es muy importante para mapear la información de los requerimientos con las estructuras de datos, a continuación se muestra como ejemplo la descripción de algunas tablas de pedidos que forman parte del diccionario de datos, el documento completo con el diccionario de datos se maneja de manera independiente a este documento

Tabla de Encabezado de Oportunidades Cerradas / Facturadas

Nombre de la Tabla: SFA_Oportunidades

Campo	Tipo de Dato	Tamaño	Descripción del Campo	Observación
1. OportunityIDSFA (Campo Llave)	Char	5	Prefijo de oportunidad (OP) y año (00), concatenado con consecutivo	Código identificador de la oportunidad que indica el tipo de documento, año y consecutivo
2. FacturaSFA (Campo Llave)	Char	5	No. Factura (ERP)	Código identificador de la factura proveniente de sistema ERP con el que se integre el sistema
3. PedidoSFA (Campo Llave)	Char	3	Prefijo de oportunidad (PO) y año (00), concatenado con consecutivo	Código identificador de la oportunidad que indica el tipo de documento, año y consecutivo
4. PorcenProbSFA (Campo Llave)	Numérico	2	Estatus de la oportunidad	0.- prospecto 10.- Calificación de oportunidad 20.- Oportunidad Calificada 40.- Demostración de la solución 60.- Prueba de la solución (asociado a producto promocional) 80.- Cierre de la oportunidad (Calificación de Pedido que se tramite y consolida en ERP) 100.- Calidad de Factura
5. ClienteSFA	Char	5	Número de cliente (ERP)	Código del cliente proveniente de sistema ERP con el que se integre el sistema
6. AfiliadaSFA	Char	5	Número de afiliada (ERP)	Código de la empresa relacionada puede ser una afiliada, sucursal, etc.
7. FCreacionOpSFA	Fecha	8	Fecha Creación	Fecha de captura del documento
8. FCierreOpSFA	Fecha	8	Fecha de Cierre de la oportunidad	Fecha de consolidación de la oportunidad como pedido.
9. FImpresiónPOSFA	Fecha	8	Fecha impresión y entrega Pedido	Fecha original del documento impreso como Pedido
10. FechaVencFacSFA	Fecha	8	Fecha vencimiento factura (ERP)	Fecha de vencimiento de la factura
11. DescuentoSFA	Numérico	9	Descuento aplicado (ERP)	Descuento en porcentaje a nivel oportunidad
12. TransporteSFA	Numérico	5	Número de Transporte	Código del transporte seleccionado

Comentarios:

En esta tabla se almacena la información concerniente de las Oportunidades Creadas y Cerradas, en donde se actualiza la información correspondiente al estado de la oportunidad (0-60% Oportunidad, 80% - Pedido, 100% Factura) agregando la información del ERP con el que se integra, que se obtiene en el proceso de transmisión-recepción (número de facturas, número de pedido).

Tabla de Detalle Partidas

Nombre de la Tabla: SFA_PartidasOportunidad

Campo	Tipo de Dato	Tamaño	Descripción del Campo	Observación
1. OportunityIDSFA (Campo Llave)	Char	5	Prefijo de oportunidad (OP) y año (00), concatenado con consecutivo	Código identificador de la oportunidad que indica el tipo de documento, año y consecutivo
2. FacturaSFA (Campo Llave)	Char	5	Consecutivo	Código identificador de la factura proveniente de sistema ERP con el que se integre el sistema
3. PedidoSFA (Campo Llave)	Char	3	Prefijo de oportunidad (PO) y año (00), concatenado con consecutivo	Código identificador de la oportunidad que indica el tipo de documento, año y consecutivo, todas las oportunidades con porcentaje de probabilidad de un 80% son consideradas pedido
4. PorcenProbSFA (Campo Llave)	Numérico	2	Estatus de la oportunidad	0.- prospecto 10.- Calificación de oportunidad 20.- Oportunidad Calificada 40.- Demostración de la solución 60.- Prueba de la solución (asociado a producto promocional) 80.- Cierre de la oportunidad (Calificación de Pedido que se tramite y consolida en ERP) 100.- Calidad de Factura
5. NumPartidaSFA	Numérico	9	Número de partida de la oportunidad/pedido/factura	Número de Partida para referenciar fecha de ingreso y actualización de la oportunidad antes de generarse como pedido o factura.
6. NumParteSFA	Numérico	9	Número de parte	Número de Parte del producto tomado del catalogo de productos vigente el ERP con el que se integre el sistema.
7. CantidaSFA	Numérico	9	Cantidad de Producto	Es la cantidad de producto de la partida
8. PrecioProdSFA	Numérico	9	Precio Unitario	Precio unitario del producto
9. CantEntregaSFA	Numérico	9	Cantidad de Producto enviada	Es la cantidad de producto de la partida que fue entregado
10. UnidadSFA	Char	12	Unidad de Medida	Código de unidad de medida tomado del catálogo de productos vigente el ERP con el que se integre el sistema.
11.FechaUpdPartSFA	Fecha	8	Fecha de modificación de de la partida	Fecha de modificación de la partida
12.FechaEntregaSFA	Fecha	8	Fecha de entrega	Fecha de entrega del producto
13.FechaFacturacionSFA	Fecha	8	Fecha de Facturación	Fecha de la factura
14.SubtotalPartidaSFA	Numérico	9	Subtotal por partida	El valor almacenado se obtiene de la multiplicación de precio unitario del producto por la cantidad de producto
15.DescuentoCltProdSFA	Numérico	9	Descuento cliente / producto	Descuento predefinido en el ERP como parámetro controlado en las reglas de negocio del Sistema de Gestión.

Comentarios:

Esta tabla almacenará las partidas de las oportunidades, pedidos o facturas dependiendo del estado de la oportunidad como se especificó en la tabla anterior.

IV.1.3 Diseño físico

El diseño físico selecciona la tecnología que será usada, este diseño agrega detalle a la arquitectura y refleja las restricciones de implementación.

- Los resultados del diseño lógico son usados para producir componentes, especificaciones a las interfaces de usuario.
- El diseño físico de la base de datos.

El uso de MS Visio nos permitió contar con una herramienta para dibujar los diagramas, almacenar la información en general en un repositorio común, generar los esqueletos de código a partir de los modelos y documentación.

Así mismo podemos identificar las dependencias entre los componentes, esta vista es de gran utilidad para los desarrolladores ya que sirve como base para la compilación y mantenimiento, si un desarrollador no contempla estas relaciones y dependencias puede provocar que una aplicación deje de funcionar como haría habitualmente.

En la siguiente figura 4.2 se puede observar los componentes de la capa de datos en su relación con los de la capa de negocios, modelado en MS Visio.

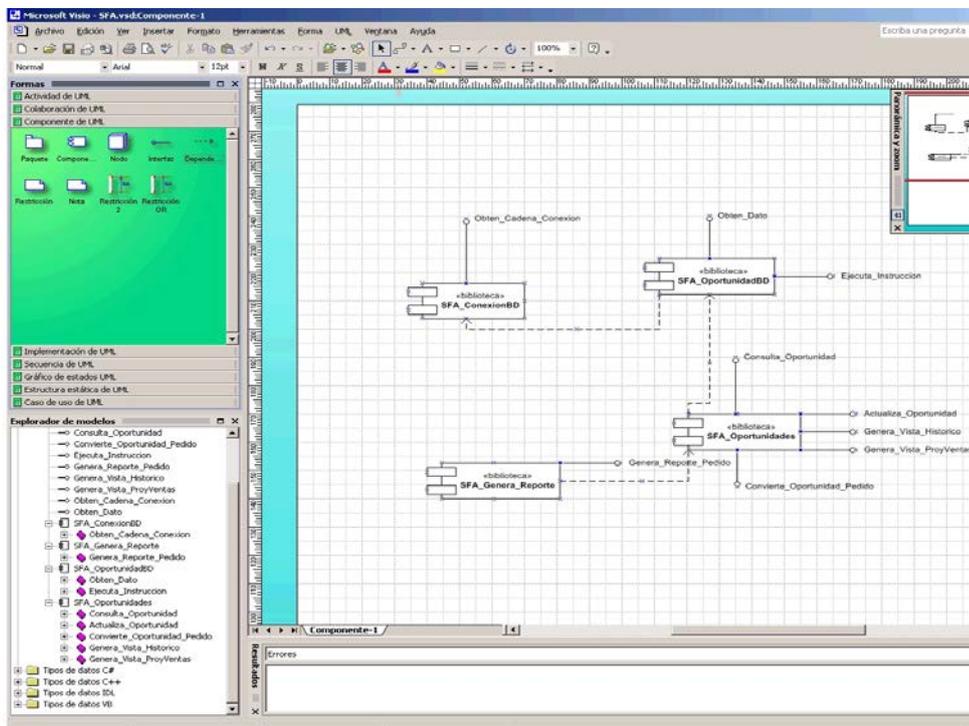


Figura 4.2 Relación entre componentes

Una vez generado el modelado de los componentes y establecidas las relaciones entre ellos es posible generar las clases que forman parte del componente, todo esto se verá reflejado en el cascarón que será utilizado para que los programadores terminen, incluyendo el código de programación y con esto los componentes estén terminados con la funcionalidad requerida para poder ser usados por otros componentes y por la capa de presentación.

Para completar el diseño físico ya con las entidades identificadas, la base de datos puede ser generada físicamente, utilizando la herramienta de modelado MS Visio, que puede generar los comandos y sentencias necesarias para crear la estructura de las tablas en el RDBMS.

IV.2.1 Fase de desarrollo

El desarrollo es la actividad de escritura del código, si el diseño ha sido hecho correctamente y con suficiente detalle, la codificación debería ser una tarea simple.

Esta etapa involucra tomar las decisiones finales, traducir los diagramas de diseño y las especificaciones en la sintaxis del lenguaje de programación seleccionado. Esto además involucra el proceso práctico de desarrollo, escribir, compilar y probar componentes.

Las herramientas usadas en este desarrollo se listan a continuación.

- Visual Studio .Net, donde se usará Visual Basic .Net para la programación de los componentes de la aplicación, así como para la interfaz gráfica.
- Visual Source Safe para el manejo de versiones.

Visual Basic .Net es un lenguaje conocido y aceptado por un alto porcentaje de programadores ya que es fácil de entender y utilizar.

Es muy importante cuando se desarrolla un proyecto poder controlar las versiones durante la etapa de desarrollo, ya que sin importar el número de programadores involucrados, el control a las modificaciones de código puede ahorrar muchas horas de esfuerzo, retrabajos y dolores de cabeza.

En este proyecto es todavía más importante el uso de estas dos herramientas, ya que se tienen 4 programadores y no se tiene tiempo para aprender un lenguaje, adicionalmente la distribución del trabajo, el controlar la ubicación y las modificaciones del código hacen un escenario ideal para las mismas.

La arquitectura seleccionada en el análisis fue la de tres capas, por lo que construiremos componentes para acceso a los datos, otros componentes que controlen las reglas de negocio y para la capa de presentación, así como todas las pantallas que usará el usuario final.

Es importante citar que para la construcción de los componentes de cualquiera de las capas (datos, negocio, presentación) se sugiere seguir los siguientes lineamientos:

- Especificar el bloque de documentación en el se hace la declaración de variables a ser usadas, fecha de creación, versión del componente, función a desempeñar, operación realizada por cada método, tipo de componente.
- Posteriormente existe bloque de manejo de errores.

Si existe un error regresamos el número de error para que el programador decida la mejor manera de informar que ha sucedido algo que no permitió la ejecución esperada, es importante considerar que estos mensajes se pueden mandar al visor de eventos de Windows, para una mejor operación del sistema.

Por convención devolvemos un valor conocido estándar numérico si no existió ningún error o en caso de existir establecer ese estándar que permita identificar los tipos de errores posibles (sistema, control de operación, etc.).

- Es importante destruir las instancias de los objetos creados, para evitar caídas del sistema por sobrecarga de memoria y decremento en el desempeño del mismo.

IV.2.1.1 Capa de datos

Componentes

Se construyó un componente especializado en acceder a la base de datos y que servirá para que el resto de los componentes lo utilice y evitar con esto la escritura repetitiva de líneas de código en el resto de los componentes.

Es importante considerar que el uso de MS Visio generará los esqueletos, con los nombres de los métodos a ser creados, es importante utilizar estándares de nombres y codificación para que todos los integrantes generen piezas fáciles de mantener y reutilizar en el futuro.

Es importante documentar al mismo tiempo de crear líneas de código ya que por lo regular la documentación se queda como una tarea que se hará al final y que en muchos casos ya nunca se hace.

Se debe considerar que estos componentes deben de ser lo más simples y especializados en su función, se deben pensar como pequeñas piezas que al combinarlas formarán una pieza con mayor funcionalidad y reutilizable.

A continuación hablaremos de los componentes de la capa de datos.

SFA_ConexionBD.dll (Nivel 0)

Este componente genérico es utilizado por todos los componentes, ya que es el encargado de establecer las cadenas de conexión, con lo que se puede utilizar para diferentes manejadores de Base de Datos.

Dicho componente contiene los siguientes métodos:

Método Obten_cadena_conexion

SFA_OportunidadBD.dll (Nivel 1)

Este componente permite realizar las funciones de consulta, alta, baja y actualización de las tablas SFA_Oportunidades, SFA_PartidaOportunidad

Método Obten_Dato

Descripción: Esta función realiza la obtención de los datos almacenados en las tablas citadas, integrando la información en una sola vista.

Método Ejecuta_Instruccion

Descripción: Ejecuta un INSERT, INSERT INTO, UPDATE, DELETE. Todas las operaciones que modifican la información de las tablas citadas.

IV.2.1.2 Capa de Negocios

Esta capa es responsable de tener la lógica del negocio, estos componentes usan a otros para poder incrementar su capacidad de proceso, es decir los componentes creados anteriormente en la capa de servicio de datos serán usados aquí.

Como ejemplo tomaremos al componente SFA_Oportunidades.dll, el cual internamente para conectarse a la base de datos hace uso de algunos componentes, uno de ellos el SFA_ConexionBD.dll y al componente transaccional de SFA_OportunidadDB.dll

SFA_Oportunidades.dll

Este componente específico es utilizado para las operaciones de negocio definidas en el documento de requerimientos funcionales para el manejo de las oportunidades de ventas, aplicando las reglas de negocio definidas para la operación.

Este componente utiliza los componentes de validación de privilegios, que definen las operaciones que puede realizar.

Dicho componente contiene los siguientes métodos:

Método Consulta_Oportunidad

Este método permite obtener toda la información relativa a una oportunidad, utilizando los componentes de la capa de datos descritos anteriormente para su despliegue en la capa de presentación (Formas frm).

Método Actualiza_Oportunidad

Este método permite realizar las operaciones de creación, actualización y borrado de una oportunidad, utilizando los componentes de la capa de datos descritos anteriormente para realizar la transacción en la Base de Datos.

Método Convierte_Oportunidad_Pedido

Este método permite convertir a pedido una oportunidad, generando el código de pedido e información relacionada a esta operación. Aplica la regla de negocio que reside en este método del componente, en la cual cuando el porcentaje de probabilidad sea de 80% y se guarde la oportunidad se ejecuta esta conversión, utilizando los componentes de la capa de datos descritos anteriormente para realizar la transacción en la Base de Datos.

Método Genera_Vista_Historico

Este método crea la vista histórica de oportunidades que tienen la categoría de pedido o factura, utilizando la regla de negocio que reside en este método del componente, en la cual cuando el porcentaje de probabilidad es del 100% se desplegará en esta vista como factura tomado la información relativa a este tipo de documento. Este método utiliza también los componentes de la capa de datos descritos anteriormente para obtener información de la Base de Datos para su vaciado en la capa de presentación (FRM).

Método Genera_Vista_Proyventas

Este método crea la vista de oportunidades vivas, utilizando la regla de negocio que reside en este método del componente, en la cual presentará sólo aquellas oportunidades con porcentaje de probabilidad menor a 80%. Este método utiliza también los componentes de la capa de datos descritos anteriormente para obtener información de la Base de Datos para su utilización en la capa de presentación (FRMGUIConsulta_Proyeccion.vb).

IV.2.1.3 Capa de presentación**Pantallas (Formas .vb)**

Esta capa interactúa directamente con el usuario de la aplicación, aquí estaremos creando las pantallas que servirán para generar una oportunidad, imprimir un pedido (reporte) y que harán uso de los componentes de regla de negocio, así como también de los de acceso a datos. La pantalla de “consulta general y proyección de ventas”, permitirá tener una vista de la actividad comercial del ejecutivo de ventas.

Esta Forma (.vb) invocará un componente de negocio para determinar los privilegios del usuario firmado en el sistema, con lo cual habilita o deshabilita los controles relacionados a cada privilegio, haciendo uso de una función propia de la capa de presentación, que encapsula este comportamiento.

A continuación se describe uno de los componentes de esta capa.

SFA_Genera_Reporte**Método Genera_Reporte_Pedido**

Este método crea el reporte formateado para la impresión de pedido, utilizando la regla de negocio que determina si el usuario puede imprimir un pedido. Este método utiliza también los componentes de la capa de datos descritos anteriormente para obtener información de la Base de Datos para su utilización en el formato predefinido.

A continuación se muestra como ejemplo en la figura 4.3, la pantalla citada anteriormente, la cual fue aprobada en el prototipo presentado en la etapa de diseño.

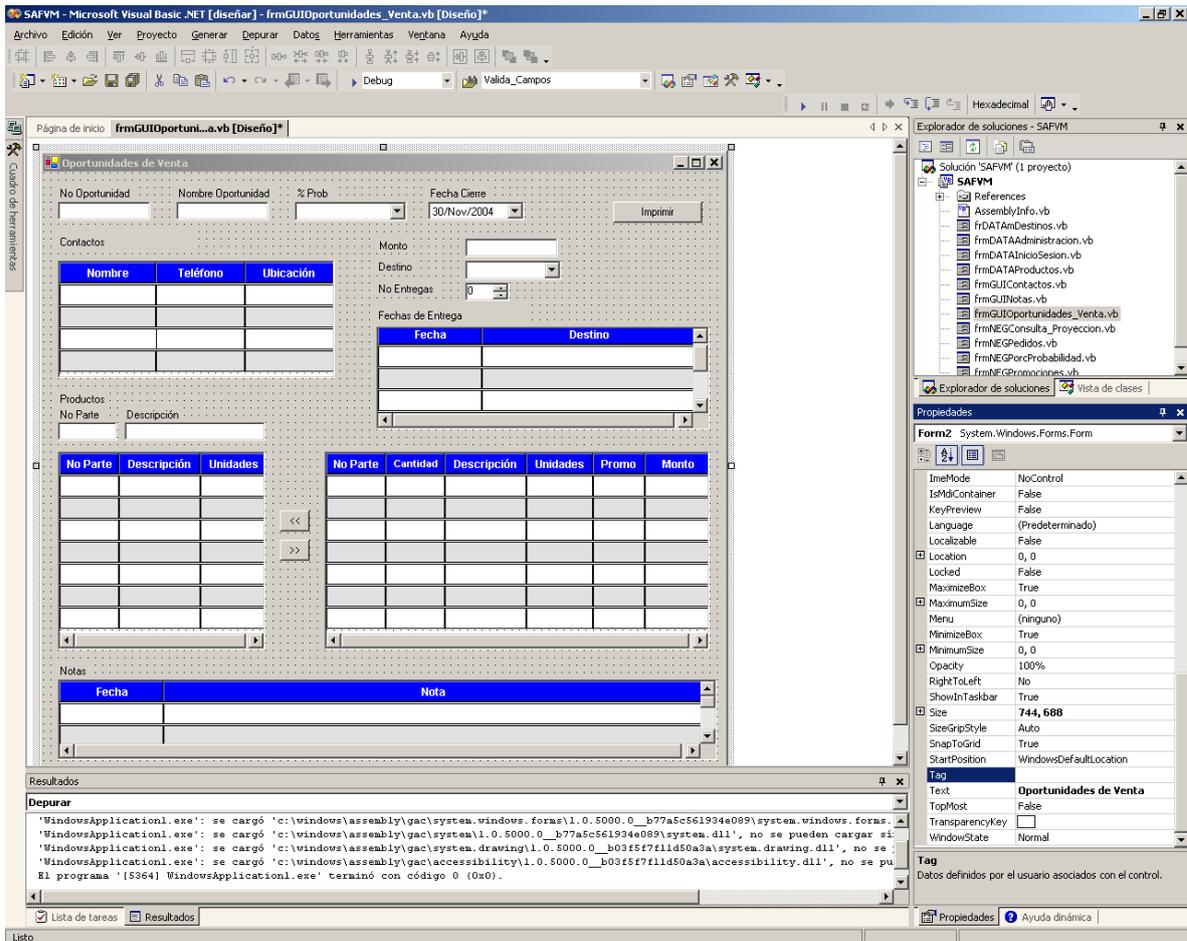


Figura 4.3 Pantalla de captura de pedidos.

Internamente esta pantalla hace uso de los componentes de negocios de Pedidos llamado SFA_Oportunidades.dll, teniendo siempre la práctica de destruirlo para liberar los recursos del cliente, como es la memoria y las conexiones a la base de datos.

Conclusiones

El desarrollo es una actividad muy popular entre los programadores y los administradores de proyectos, los programadores sienten que es una área familiar para ellos y para los administradores de proyectos piensan que hasta que se codifica se empieza realmente a trabajar. En muchos casos, la prisa por comenzar a codificar es un error, se dice que se debe conducir el proceso de desarrollo en una manera iterativa, analiza un poco, diseña un poco, codifica un poco y entonces regresa al análisis. Si alguna de estas actividades demora demasiado puede provocar mucha administración y documentación, un error muy común es querer tener una aplicación perfecta que no requiera ninguna modificación con solamente un ciclo de desarrollo, es decir en la versión 1.0.

Desafortunadamente, muchos equipos sacrifican la calidad del código cuando están presionados para terminar un programa en una fecha límite en particular, a la larga esto es más costoso ya que compromete la calidad de todo el proyecto y nunca se termina adecuadamente.

Capítulo V. Aseguramiento de la calidad

Objetivo del capítulo:

El objetivo de este capítulo es proporcionar una visión general de los procesos realizados para poder liberar el sistema de fuerza de ventas al usuario final, desarrollado por el equipo de programadores, aquí se examina la forma en que se asegura la calidad y prueba de los sistemas y software.

El objetivo del aseguramiento de la calidad no sólo abarca las pruebas para descubrir los errores de programación, sino también de usabilidad y reglas de negocio. Puede ser que el equipo de desarrollo programó muy pocos errores pero resulta que la aplicación es muy difícil de usar o poco operable para el usuario, o quizás no tiene todas las validaciones que se requieren o un proceso se programó de manera muy diferente a como se realiza en la operación real.

En los negocios, organizaciones y en las actividades de todos los días se automatizan más funciones críticas y se confía más en lo que es automatizado. Esta realidad es una carga creciente para los analistas de sistemas al asegurar que los sistemas que desarrollen sean de una calidad adecuada.

V.1 Detección y corrección de fallas**Confiabilidad de los sistemas**

Un sistema confiable es aquel que no produce peligro o fallas costosas cuando se utiliza en forma razonable, es decir, de la manera que un usuario común espera que sea normal, esta definición reconoce que los sistemas no siempre pueden ser utilizados en las formas en las que esperan los que diseñan. Los usuarios y las operaciones de negocios cambian; sin embargo, existen pasos que los analistas pueden llevar a cabo para asegurarse que el sistema es confiable cuando sea instalado y que la confiabilidad se mantenga después de la puesta en marcha.

Confiabilidad

Existen dos niveles de confiabilidad; el primero es que el sistema cubra los requerimientos correctos. Si, por ejemplo, se espera que tenga características específicas de seguridad o controles incorporados por los usuarios, pero el diseño falla al especificarlos y permite la pérdida de información por un periodo largo antes que alguien detecte el problema, el sistema no es confiable.

La confiabilidad en el nivel de diseño es posible sólo si se realizó una completa y efectiva determinación de los requerimientos de sistemas por parte del analista.

El segundo nivel de confiabilidad de sistemas es el trabajo efectivo de éstos proporcionado al usuario.

Un error ocurre cuando el sistema no produce la salida esperada. Es cierto que ningún programa está totalmente o completamente depurado y probado, o puede ser demostrado como correcto. Los errores no se limitan sólo al uso correcto de la sintaxis de la programación. Una falla es cuando ocurre un error de software, considerado por su nivel de gravedad.

Aseguramiento de la calidad

Asegurar la calidad es la revisión de los productos de software y la documentación relacionada para que sean completos, correctos, confiables y por supuesto incluye asegurar que los sistemas cumplan las especificaciones y requerimientos que se proyectaron para su uso y desempeño.

Niveles de seguridad

Los analistas utilizan cuatro niveles para asegurar la calidad; prueba, verificación, validación y certificación.

Prueba

La prueba de sistemas es un proceso caro pero crítico que puede utilizar hasta 50% del presupuesto del desarrollo de programas. En general, la prueba de programa se lleva a cabo para probar que no existen errores en un programa. El enfoque más útil y práctico es entender que la prueba es el proceso de ejecutar un programa con la intención específica de encontrar errores; es decir hacer que el programa falle. Una prueba exitosa, por lo tanto, es aquella en la que se encuentra un error. Los desarrolladores no pueden probarla.

Verificación y validación

La verificación también se utiliza con objeto de encontrar errores. Se desarrolla al ejecutar un programa en un ambiente simulado. La validación se refiere al proceso de utilizar el software en un ambiente real para encontrar errores.

Certificación

La certificación del software es la confirmación de que el programa es correcto. Esto es un aspecto que está creciendo en importancia para las aplicaciones de sistema de información. Esta certificación consiste en probar el software contra las características establecidas del mismo, si no se encuentran discrepancias serias o fallas, se certifica que el software realiza lo que la documentación menciona.

Estrategias de Prueba

Ya se ha indicado que la filosofía detrás de la prueba es encontrar errores. Los casos de prueba se preparan con este propósito en mente. Un caso de prueba es un conjunto de datos que el sistema procesará como una entrada normal; sin embargo, los datos son creados con la intención expresa de determinar si el sistema los procesará correctamente. Por ejemplo, los casos de prueba para el manejo de pedidos deben incluir situaciones en donde las cantidades que serán solicitadas exceden, igualen y sean menores que las cantidades que se tienen en existencia.

Existen dos estrategias generales para probar software, las estrategias de prueba de código y de prueba de especificación.

Pruebas de Código

La estrategia de prueba de código examina la lógica del programa. Para seguir este método de prueba el analista desarrolla casos de prueba que resultan en la ejecución de cada instrucción del programa o módulo, es decir, cada ruta dentro del programa se prueba. Una ruta es una combinación específica de condiciones; en el ejemplo del sistema de pedidos, una ruta a través del sistema es generar pedidos, donde se introducen los productos, después las cantidades correctas, los precios y los descuentos son los vigentes y se almacena la información en la base de datos.

En general la prueba de código parece ser un método ideal para probar el software. El razonamiento de que todos los errores del software pueden ser descubiertos al verificar cada ruta dentro de un programa, es falso. Esta estrategia de prueba no indica que el código cumpla con sus especificaciones ni determina que todos los aspectos estén considerados.

Prueba de especificación

Para llevar a cabo la prueba de especificaciones, el equipo de pruebas examina las especificaciones que el programa debe hacer y cómo debe desempeñarlas bajo diferentes condiciones. Entonces se desarrollan los casos de prueba para cada condición o combinación de condiciones y se someten a procesamiento. Esta estrategia trata al programa como una caja negra, es decir, el equipo de pruebas no ve dentro del programa para estudiar el código y no se preocupa si cada instrucción o ruta del programa se prueba. No es una prueba completa; sin embargo, la suposición es que si el programa cumple con las especificaciones no fallará.

Niveles de prueba

El equipo de pruebas debe realizar tanto la prueba unitaria como la del sistema total.

Prueba unitaria

En la prueba unitaria el equipo de pruebas, prueba los programas que conforman un sistema. (Por esta razón a veces se denomina prueba de programas) Las unidades de software de un sistema son los módulos y rutinas que se ensamblan e integran para desempeñar una función específica.

La prueba unitaria se enfoca primero a los módulos, independientes uno de otro, para localizar errores, lo que permite a quien está haciendo la prueba detectar errores en la codificación y lógica que están contenidos dentro de cada módulo.

Si el módulo recibe entrada o genera salida, los casos de prueba también se necesitan para probar el límite de valores esperados, incluyendo tantos valores válidos como inválidos. Si el módulo está diseñado para desempeñar iteraciones, con procesos específicos contenidos dentro de un ciclo, es recomendable ejecutar cada condición límite: cero iteraciones, una iteración a través del ciclo y el número máximo de iteraciones.

Por supuesto, siempre es importante examinar los resultados de la prueba, pero debe darse especial atención a estas condiciones, con mucha frecuencia aquí se presentan los errores por supuestos mal manejados.

La prueba unitaria puede llevarse a cabo de abajo hacia arriba, comenzando con los módulos más pequeños y de menor nivel y desarrollando uno a la vez. Para cada módulo en la prueba ascendente, un programa corto (conocido como programa conductor, porque maneja o corre el módulo) ejecuta el módulo y proporciona los datos necesarios, de manera que se le pide al módulo desempeñar en todas las formas lo que tiene que hacer dentro de un sistema mayor. Cuando los módulos del nivel inferior se prueban, la atención se revierte a aquellos del siguiente nivel que utilizan a los de menor nivel. Son probados en forma individual y después unidos con los módulos de menor nivel previamente examinados.

La prueba descendente, como su nombre lo indica, comienza con los módulos de mayor nivel; sin embargo, dado que las actividades detalladas normalmente, desempeñadas en las rutinas de menor nivel no se proporcionan, se escriben los módulos esclavos. Un módulo esclavo es que puede ser llamado por el módulo de mayor nivel y cuando se le alcanza apropiadamente regresará un mensaje al módulo que lo llamó, indicando que ocurrió la interacción en forma apropiada.

Durante esta etapa es importante documentar los errores que se presentan en la aplicación, es importante incluir una descripción de los pasos a seguir para reproducir el error, es además importante generar un formato para el documento que el equipo de pruebas se encargará de ir llenando y que se reporten los errores de manera uniforme.

No todo lo que se reporta por los usuarios que prueban son errores en la funcionalidad o de diseño, siempre existen oportunidades de mejora, aunque en la mayoría de los casos se está mejorando la funcionalidad solicitada en su momento y que fue desarrollada de acuerdo a los establecido por los analistas y arquitectos de la aplicación durante la etapa de diseño. Es importante que cuando se presentan muchos incidentes o errores, éstos se pueden priorizar para poder atender aquellos que sean de severidad y dejando al final aquellos que no son tan importantes. El ciclo de pruebas se debe repetir una vez que los desarrolladores han corregido y han liberado una nueva versión.

Es importante que el administrador del proyecto se reúna con los clientes para dar seguimiento de todos estos errores y quede de acuerdo con el cliente con la secuencia de trabajo.

V.2 Pruebas finales y liberación de versión 1.0

Prueba de Sistemas

La prueba de sistemas no prueba el software sino la integración de cada módulo en el sistema. También hace pruebas para encontrar discrepancias entre el sistema y su objetivo original, las especificaciones actuales y la documentación del sistema.

El equipo de pruebas trata de encontrar áreas en donde los módulos se han diseñado con diferentes especificaciones para la longitud de datos, tipo y nombre de los elementos dato.

La prueba de sistemas debe verificar también que el tamaño de los archivos sea adecuado y que se han construido apropiadamente los índices.

Prueba de carga

Su objetivo es determinar si el sistema manejará el volumen de actividades que ocurren cuando éste se encuentra en el punto máximo de demandas de procesamiento, pero siempre considerando acercarse a situaciones reales. Una prueba de carga que simula que un pedido es generado por un vendedor cada segundo, no es una prueba que sirva para el sistema ya que no es una situación que se pueda presentar en la vida real.

Prueba de almacenamiento

Los analistas especifican una capacidad para el sistema cuando se diseña y construye. Las capacidades se miden en términos del número de registros que un disco puede manejar o que un archivo puede contener. Estas capacidades están ligadas a un espacio y al tamaño de los índices, llaves de registro, etc., y también se deben probar.

La prueba de almacenamiento requiere de la introducción de datos reales, muchos sistemas nunca se prueban de esta manera. Los usuarios encuentran demasiado tarde que las características que se ofrecían durante la puesta en marcha no son verdad, no existe suficiente capacidad de almacenamiento para las transacciones y los registros del archivo maestro.

Prueba de desempeño

Cuando los analistas están desarrollando un diseño, sus preocupaciones se centran más sobre los reportes, entradas, archivos y secuencias de procesamiento que sobre el tiempo de desempeño. Durante las pruebas sencillas unitarias y de sistemas, se utilizan conjuntos de datos relativamente pequeños para encontrar errores u ocasionar fallas; por lo tanto los usuarios con frecuencia sólo se dan cuenta de que encuentran qué tan rápido o lento es el tiempo de respuesta del sistema hasta después de que se ha instalado y cargado con datos reales. Esto puede ser demasiado tarde.

Prueba de recuperación

El equipo de pruebas debe suponer que el sistema fallará y que los datos se dañarán o perderán. Aún cuando los planes y procedimientos se escriban para cubrir estas situaciones, esto también debe probarse, creando una situación de falla o de pérdida de datos en donde los usuarios se vean forzados a volver a cargar y recuperar una copia de respaldo, el equipo de pruebas puede realmente determinar si los procedimientos de recuperación son adecuados. Los planes mejor diseñados normalmente se ajustan o aumentan después de esta prueba.

Prueba de procedimiento

Los manuales de documentación y de trabajo que le dicen al usuario cómo desempeñar ciertas funciones se prueban con bastante facilidad pidiéndole al usuario que los siga exactamente a través de una serie de situaciones.

Por supuesto, un conjunto de manuales de procedimiento bien diseñado es insustituible. Los analistas se concentran en los detalles principales y críticos de un diseño de sistemas e incluyen esto en la documentación. También le ponen atención a los detalles pequeños, pero a menudo las descripciones de los detalles no se obtienen dentro de la documentación. Este tipo de prueba no solamente muestra dónde se necesitan sino también en dónde están mal, es decir, dónde se sugieren acciones en la documentación que no cumplan con aquellas que deben realmente llevarse a cabo para hacer que el sistema funcione.

Pruebas de factores humanos

Consiste en encontrar respuesta a las preguntas sobre cómo reaccionará la gente al sistema en formas que no se hayan anticipado. El analista debe asegurarse que se despliegue un mensaje informándole al usuario acerca de las situaciones que se están presentando en el sistema, como procesamiento de transacciones, generación de reportes o procesos de larga duración.

También el analista debe asegurarse de observar cómo la gente introduce los datos, que tan útil resulta la aplicación que usará el usuario en su vida diaria para realizar su trabajo, conocer la opinión de los usuarios, todo esto para tomar en cuenta estas consideraciones y realizar ajustes antes de liberar la aplicación en caso que se considere muy necesario.

V. 3 Elaboración de manuales

Manuales de operación

Incluso los sistemas bien diseñados y técnicamente bien hechos pueden tener éxito o fracaso debido a la forma en que se operan y se utilizan, por lo tanto, la calidad de la capacitación del personal involucrado, así como de los manuales de operación y de administración, ayuda o dificulta y puede incluso obstaculizar por entero el éxito de la puesta en marcha de un sistema de información. Las personas que trabajarán con el sistema o que se verán afectadas por éste deben conocer con detalle las funciones que desempeñarán, como utilizarán el sistema y lo que éste hará o no. Tanto los operadores como los usuarios necesitan contar con los manuales que correspondan a su nivel de usuario o rol que desempeñen en el sistema.

El equipo de soporte técnico debe tener manuales para la instalación del sistema, así como de la operación del mismo a un nivel más técnico, ya que ellos deberán poder resolver cualquier tipo de solicitud de soporte de los usuarios finales.

Ante esta situación se deben considerar manuales para los administradores del sistema y los usuarios finales.

Manuales técnicos para los operadores del sistema

Muchos sistemas dependen del personal del centro de cómputo, quien tiene la responsabilidad de mantener el equipo en buenas condiciones, así como de proporcionar el servicio de apoyo necesario. Su manual debe garantizar que están todas las operaciones posibles que se pueden realizar con el sistema, tanto las de rutina como las extraordinarias.

Si el sistema requiere la instalación de equipo nuevo, por ejemplo, de un nuevo sistema de cómputo, deben incluirse también todos los componentes necesarios ya sean de software o de hardware involucrado, así como los manuales respectivos de esas piezas.

El equipo de soporte también debe tener conocimiento de los errores comunes que pueden ocurrir, cómo detectarlos y tomar las medidas necesarias cuando se presentan.

A continuación se muestra en la figura 5.2 un ejemplo de las pantallas que se describen en el manual de administración del sistema, en este ejemplo se incluye la parte de administración de usuarios donde se configuran los datos de los usuarios, así como dependencias y privilegios.

The screenshot shows a window titled "Pantalla de Administración de Derechos" with the following content:

Ejecutivo de Ventas

No. Empleado	Nombre	Centro de Costos
000020	REBECA SANCHEZ LABASTIDA	0958

Jefe Inmediato

No Empleado	Nombre	Centro de Costos
000010	JOSE ANTONIO VELAZQUEZ VIDRIO	0958

Derechos

- Crear Oportunidades
- Actualizar Oportunidades
- Imprimir Pedidos
- Transferencia de Información
- Consultar Información Histórica

Rango de Fechas

Período Inicial: 06/Ago/2004 Período Final: 08/Sep/2004

- Consultar Información Subordinados
- Reasignar Oportunidades

Buttons: Nuevo, Limpiar, Guardar, Salir

Figura 5.2 Ejemplo del manual de administración

Manuales de usuario

El manual de usuario radica en la operación del sistema mismo, el objetivo de este manual es familiarizar al usuario con el sistema de captura de datos, procesamiento y transacciones. La documentación adecuada, aunque esencial, no sustituye a la capacitación.

En la figura 5.3 un ejemplo de las pantallas que se describen en el manual de usuarios del sistema.

Oportunidades de Venta

No Oportunidad: OP0404 Nombre Oportunidad: OP PRODUCTO 1 % Prob: 40 % Fecha Cierre: 30/Nov/2004 Imprimir

Contactos

Nombre	Teléfono	Ubicación
Rebeca Sánchez	55567809	Tulipán # 45
José A Velázquez	56690943	Dirección # 55

Monto: \$ 4,000,000.00 Destino: Destino 1 No Entregas: 1

Fechas de Entrega

Fecha	Destino
01/Dic/2004	Dirección Destino 1

Productos

No Parte	Descripción	Unidades
0101012	Producto 1	Piezas
093933	Producto 2	Piezas
872347	Producto 3	Caja
797953	Producto 4	Toneladas
878554	Producto 5	Empaques
785436	Producto 6	Piezas
778054	Producto 7	Piezas

No Parte	Cantidad	Descripción	Unidades	Promo	Monto
093933	20,000	Producto 2	Piezas	No	\$ 200.00
872347	1,000	Producto 3	Caja	Si	\$ 100.00

Notas

Fecha	Nota
30/Nov/2004	El Cliente tiene presupuesto para las cantidades de producto especificadas en la oportunidad y sólo requiere aprobac

Nuevo Limpiar Guardar Salir

Figura 5.3 Ejemplo del manual de usuario

Conclusiones:

La mejor manera de alcanzar un nivel alto de confiabilidad es probando el sistema, la mayoría de los clientes no le da importancia a las pruebas y esperan contar con un producto libre de errores y 100 % perfecto, no importa lo que se haga ni la metodología que se siga, ningún sistema puede garantizar que nunca se presentará un error, pero tener muchos errores cuando está un sistema a punto de liberarse a producción genera un ambiente de incertidumbre en los usuarios finales y los clientes cuestionan la factibilidad del proyecto.

La documentación es otra tarea que normalmente se deja al final y que en muchos proyectos nunca se hace, por eso debe considerarse que los detalles importantes sean documentados conforme se avanza en las etapas del proyecto, no esperarse al final ya que en muchos casos la relación con el cliente está muy desgastada y se concluye la relación laboral sin tener ningún tipo de documentación complicando con esto la operación y mantenimiento para el equipo de soporte del cliente.

La documentación es un elemento importante de apoyo en la capacitación de los diferentes roles de usuario.

Capítulo VI. Implantación y liberación a producción

Objetivo del capítulo:

Describir el proceso de implantación y liberación a producción realizado en la empresa seleccionada para el uso de la versión 1.0 del sistema motivo de esta tesis, tomando como guía la metodología seleccionada y descrita en el capítulo II.

A continuación describimos dos actividades que fueron esenciales, la planeación y la prueba piloto, ya que en ellas se establecieron los procedimientos que aseguraron el éxito de esta fase.

VI.1 Planeación:

Las actividades de planeación consideraron los recursos que participaron en la implantación y liberación del sistema, así como del staff que dio soporte al usuario durante el período de liberación a producción.

Se definieron los recursos y procesos necesarios, con el fin de establecer con precisión el plan de trabajo y fechas de liberación final del sistema en ambiente de producción.

- **Escenarios de las localidades a implantar:**

Fue importante considerar las condiciones de cada uno de los sitios en donde el sistema fue implantado, definiendo puntualmente la infraestructura disponible en cada localidad, para obtener los escenarios de implantación.

Lo anterior permitió definir un procedimiento estándar para cada escenario y evitó problemas y retrasos al equipo designado para esta actividad.

Para el proceso de la implantación a realizar en la Empresa seleccionada se identificaron dos escenarios:

1. Oficinas Centrales y Sucursales Remotas: Infraestructura de Red que permite una instalación masiva a través de un servidor central en los diferentes equipos (clientes) que utilizarán el sistema.
2. Equipos móviles de la fuerza de ventas: Instalación independiente en cada equipo.

- **Migración y carga de Información inicial :**

Es importante citar que durante la fase de diseño se definieron los procesos para cargar la información al sistema tanto en la base de datos central y en las bases de datos que residirán en los equipos móviles (laptops) a utilizar por la fuerza de ventas.

Adicionalmente en la fase de desarrollo y aseguramiento de calidad se construyeron y probaron los módulos que ejecutarían esta función.

Los módulos de carga contemplan principalmente dos procesos que se citan a continuación:

- 1) Proceso de transferencia de información del ERP al Sistema.
Existe un módulo de administración en donde se establece el tipo y cantidad de información (transferencia periódica o total) en ambos sentidos.
- 2) Proceso de transferencia de información de los Equipos Móviles a la Base de Datos Central.
Existe un módulo de transferencia de información habilitado para el Usuario del equipo móvil que ejecuta dos operaciones, envío y recepción de información a oficinas centrales.

Con base en lo anterior las actividades que se realizaron para implantar y liberar el sistema en producción fueron:

- Definir el volumen y antigüedad de información histórica que sería cargada en forma inicial a la Base de Datos central y a cada una de los usuarios móviles dependiendo de su rol. Esto se definió tomando en consideración las especificaciones establecidas en la fase de diseño con base al desempeño que debe tener el sistema para usuarios móviles.
- Establecer el momento y tiempo que llevará esta actividad con base a la fecha de arranque, definida para iniciar las operaciones utilizando el sistema.

- **Definición de proceso de instalación :**

Para los dos escenarios identificados se tendrá un instalador que contemple la instalación de los adaptadores de hardware necesarios para el funcionamiento del sistema, registro de parámetros y banderas que definen el perfil de cada usuario y la instalación de los componentes propios del sistema.

El instalador generará una bitácora que registre el estatus de la instalación como elemento de control y seguimiento. Esto permitirá que los recursos asignados a la implantación pueda asegurarse que todos los usuarios inician operaciones de la misma manera, siendo que el sistema es una herramienta de colaboración en donde la información se comparte y se relaciona entre diferentes usuarios.

Se consideraron los tiempos que se llevaron cada una de las instalaciones ya sea en oficinas centrales o en usuarios móviles con el fin de programar con la mayor precisión la liberación del sistema para inicio de operaciones formales.

- **Piloto:**

Esta actividad requirió la instalación de un ambiente idéntico al de producción con base a los requerimientos especificados respecto a Hardware y Software, para cada uno de los escenarios definidos.

Se identificaron los usuarios (uno de cada perfil) para iniciar con un grupo piloto con el cuál se realizaron transacciones reales, a las cuales se les dio seguimiento puntual y una vez validada la calidad de las mismas, se replicaron gradualmente hasta trasladar a todos los usuarios.

- **Soporte a Producción:**

Esta actividad es vital en el momento de arranque de las operaciones con el sistema, por lo que el plan contempla el número de recursos necesarios para garantizar el tiempo de respuesta para los incidentes que sean reportados.

El equipo tiene como función revisar la consistencia e integridad de la información que se ingresa y se obtiene del ERP, así como vigilar los procesos de transferencia de información en tiempo y forma.

Esta etapa incluyó la transferencia de conocimiento al grupo de sistemas que se encargará del soporte y mantenimiento del sistema en forma permanente.

Esta transferencia de conocimiento incluyó el conocimiento de los procesos, bases de datos para la explotación de información y parámetros dinámicos del sistema.

- **Capacitación:**

La capacitación tuvo dos categorías para lograr la adecuada transferencia de conocimiento a los diferentes tipos de usuario del sistema.

Usuarios Final: Esta capacitación consistió en entrenar al usuario final en el uso adecuado de la funcional del sistema, lo cual se realizó por grupos formados por usuarios con el mismo perfil. La capacitación la realizó el mismo equipo que participó en la etapa de control de calidad, realizando las pruebas y elaborando el manual de usuario basado en la experiencia y conocimiento del sistema.

Usuarios Administrador (Técnico): Esta capacitación consistió en entrenar al usuario que administrará y configurará el sistema para su adecuado funcionamiento. La capacitación se realizó en conjunto con el líder de proyecto, para aprovechar su experiencia y conocimiento del sistema.

Esta capacitación incluye la explicación detallada de la documentación técnica elaborada durante la etapa de diseño y desarrollo del sistema.

La capacitación concluyó con la configuración del sistema en conjunto con el área de sistemas, para las pruebas piloto, migración al ambiente de producción y la puesta punto.

VI. 2 Estructura del equipo de trabajo de implantación

Esta actividad se apoya en las características y roles de cada uno de los individuos que participaron en la construcción del proyecto.

La metodología plantea un modelo en donde las actividades de cada uno de los roles tienen una secuencia definida para cada fase, a través de flujos de trabajo circulares que permiten la liberación parcial y gradual de módulos terminados.

Se logró el objetivo de mantener el presupuesto del proyecto asignando únicamente a los recursos necesarios para garantizar la adecuada capacitación de los usuarios finales y técnicos, así como lograr la transferencia de conocimiento efectiva y práctica. El equipo de trabajo que se empleó en esta etapa se muestra en la figura siguiente.

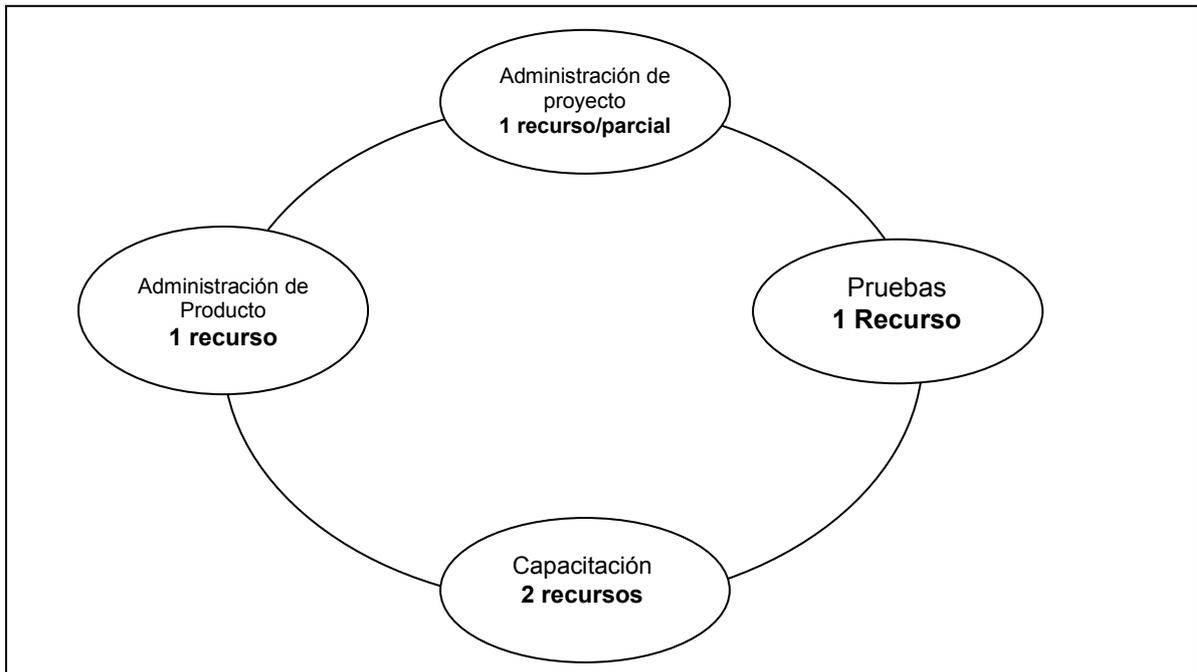


Figura 6.1 Equipo de Trabajo

Conclusiones:

Esta etapa es crucial y definitiva para muchos proyectos con resultados desastrosos, resultado de la NO utilización de una adecuada metodología, dejando proyectos inconclusos después de un desgaste intenso del cliente y proveedor.

Partiendo de una aplicación adecuada de la metodología propuesta, esta etapa permitió afinar la configuración del sistema para el manejo de los volúmenes de información reales, ajustar la programación de los procesos que ejecuta el sistema acorde a las necesidades de negocios. Por ejemplo la definición de horarios para replicación y sincronización de la información, etc.

Otro de los logros de esta etapa lo constituye una buena capacitación del sistema de lo cual depende su adopción, su éxito y la recuperación de la inversión realizada, a través de su explotación por parte de los usuarios a capacitar.

La instalación del piloto nos permitió probar y mejorar la documentación técnica elaborada en las etapas de diseño y desarrollo, lo que permitirá al área de sistemas encargada preparar su plan mantenimiento y de recuperación en caso de desastres, acorde a sus políticas y procedimientos.

Conclusiones Generales

No se puede escribir una receta conteniendo los pasos para garantizar el éxito de un proyecto, lo que se puede recomendar es seguir una metodología que brinde una guía flexible, que pueda responder rápida y efectivamente a las condiciones propias del proyecto y recursos con los que se cuente para el desarrollo del mismo.

Lo más importante es tener una buena comunicación entre los miembros del equipo del proyecto y el cliente. Todos los miembros deben tener claro lo que se está desarrollando y cuál es su alcance final para no generar falsas expectativas.

Los recursos involucrados deben tener el perfil necesario para el rol que van a desempeñar o el proyecto está en riesgo de no tener fin.

Es importante que cada etapa tenga una duración adecuada y una secuencia lógica, hay que trabajar en el análisis y diseño de la aplicación y no preocuparnos por empezar a programar desde el primer día, ya que esto provoca que se termine haciendo algo que no es lo que se espera. La calidad del producto es algo que debe importar a todo el grupo del proyecto.

La documentación y la capacitación deben estar terminadas y haber cubierto a la mayor cantidad de usuarios finales y operativos antes de la liberación a producción.

Para liberar a producción un sistema debe estar estable, con la mayor cantidad de pruebas de carga en ambiente real y un plan de mejora continua definido.

El seguimiento adecuado de la metodología planteada en este trabajo de tesis, nos permitió lograr la exitosa culminación del proyecto, con una desviación controlada de los tiempos de entrega, a través del cumplimiento de los objetivos de cada etapa, los que se resumen a continuación:

En el capítulo I se logró visualizar como este tipo de herramientas informáticas favorecerán a las empresas de este sector empresarial (manufactura) a través de un manejo eficaz de los recursos comerciales, e intensifiquen la relación de negocio con sus clientes.

En el capítulo II se llegó a la conclusión de que todas las metodologías son buenas, lo más importante es documentar aquellas etapas y actividades que permiten identificar cuál es la necesidad y requerimientos del negocio para no tener dificultades con las expectativas generadas, ya que éstas pueden hacer que un proyecto fracase, que no se logre terminar o que los costos provoquen una pérdida para cualquiera de las partes involucradas.

En el capítulo III la principal conclusión es que la etapa de Planeación y Análisis es crucial para definir los compromisos de ambas partes, definiendo los responsables de cada etapa y que rol jugaran para la adecuada evolución del proyecto. Siendo el momento adecuado para la toma de decisión definitiva si se lleva a cabo el proyecto con la expectativa inicial, se modifica o se desiste del mismo.

En el capítulo IV podemos resaltar principalmente que en muchos casos, la prisa por comenzar a codificar es un error, se dice que se debe conducir el proceso de desarrollo en una manera iterativa, analiza un poco, diseña un poco, codifica un poco y entonces regresa al análisis. Si alguna de estas actividades demora demasiado puede provocar mucha administración y documentación, un error muy común es querer tener una aplicación perfecta que no requiera ninguna modificación con solamente un ciclo de desarrollo, es decir en la versión 1.0.

En el capítulo V tuvimos como conclusión que la mejor manera de alcanzar un nivel alto de confiabilidad es probando el sistema de manera ordenada en conjunto con el usuario final. La mayoría de los clientes no le da importancia a las pruebas y esperan contar con un producto libre de errores y 100 % perfecto, no importa lo que se haga ni la metodología que se siga, olvidando que ningún sistema puede garantizar que nunca se presentará un error, pero tener muchos errores cuando está un sistema a punto de liberarse a producción genera un ambiente de incertidumbre en los usuarios finales y los clientes cuestionan la factibilidad del proyecto, lo cual debe eliminarse siguiendo los pasos recomendados en esta etapa.

La documentación es un elemento importante de apoyo en la capacitación de los diferentes roles de usuario.

En el capítulo VI se llegó a la conclusión de que esta etapa es crucial y definitiva para muchos proyectos con resultados desastrosos, resultado de la NO utilización de una adecuada metodología, dejando proyectos inconclusos después de un desgaste intenso del cliente y proveedor.

Partiendo de una aplicación adecuada de la metodología propuesta, esta etapa permitió afinar la configuración del sistema para el manejo de los volúmenes de información reales, ajustar la programación de los procesos que ejecuta el sistema acorde a las necesidades de negocios.

Es importante citar que el material presentado en esta tesis ilustra de una forma práctica la metodología empleada, con el claro objetivo de aportar únicamente un documento referencial fundamentado en nuestra experiencia que pueda servir de guía en el desarrollo de una herramienta informática que resuelva efectivamente una necesidad específica de un sector empresarial.

Bibliografía:

- Analyzing Requirements and Defining Solution Architectures
MCSD Training Kit
Microsoft Press
ISBN 0-7356-0854-7
- Análisis y Diseño de Sistemas de Información
James A. Senn
McGraw-Hill
ISBN 968-422-165-7
- UML toolkit
Hans-Erik Eriksson and Magnus Penker
Wiley Computer Publishing
Printed in the USA
ISBN 0471 19161 -2
- www.inegi.gob.mx