



UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

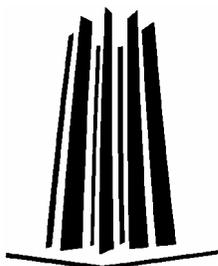
**Aplicación de la Ingeniería de Software en el
Análisis, Desarrollo, Implementación y Mantenimiento de
Sistemas Cliente-Servidor para la Administración Integral
de Agencias Automotrices en México**

**INFORME DEL
EJERCICIO PROFESIONAL**
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
PRESENTA
FELIPE SALAZAR VARGAS

Asesor: M. en C. Jesús Hernández Cabrera

Estado de México

2008





Universidad Nacional
Autónoma de México



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.

DEDICATORIAS

A Sapsa,¹ mi empresa. Por haberme dado la oportunidad de poner en práctica mis conocimientos teóricos y así alcanzar la realización tanto profesional como personal.

A la UNAM, mi escuela. Por crear un profesionista con capacidad de análisis, pensamiento propio y creatividad... por haberme dado la visión necesaria para más que formar un empleado, formar un pensador.

A Neza, mi ciudad. Por haber crecido en la adversidad y en ella cultivar mi espíritu y forjar mi carácter.

A Graciela Vargas, mi madre. Por tomar la decisión de que yo existiera y guiar mis pasos.

A Camilo Salazar, mi padre. Por enseñarme a ser un Hombre, más que con palabras con el ejemplo de toda la vida y por transmitirme todos los valores que rigen y dan sentido a mi existencia.

¹ Sistemas Automotrices Proactive, S.A. de C.V.

ÍNDICE

Resumen	4
Capítulo 1 Introducción	6
1.1 Experiencia Profesional – Administrador / Desarrollador.....	6
1.2 Problemática Enfrentada en la Experiencia Profesional	7
1.3 Solución de Problemas.....	9
1.4 Objetivos de este informe.....	10
Capítulo 2 Software para agencias Automotrices en México	11
2.1 Panorama Actual de la Industria Automotriz	11
2.2 Sistemas de Gestión Automotriz	21
Capítulo 3 Desarrollo de un caso práctico	26
3.1 Análisis	27
3.1.1 Concepto de Análisis.....	27
3.1.2 Recomendaciones Previas al Análisis.....	29
3.1.3 Determinar requerimientos (¿Qué hay que hacer?).....	30
3.1.4 Análisis Estructurado (Detalles de los requerimientos).....	41
3.2 Diseño.....	50
3.2.1 Diseño dentro del contexto de la ingeniería de software	52
3.2.2 Proceso y calidad del diseño.....	53
3.2.3 Conceptos del Diseño	54
3.2.4 El Modelo de Diseño	57
3.2.5 Diseño de Software Basado en Patrones	58
3.3 Construcción.....	60
3.2.1 Elección de herramientas	61
3.2.2 Estándares de programación	65
3.4 Implementación	72
3.4.1 Metodología de Implementación	72
3.4.2 Equipos de Implementación	73
3.4.3 Áreas de Especialización	73
3.5 Atención a Clientes.....	81
3.5.1 Objetivo del Área de Atención a Clientes.....	81
3.5.6 Calldesk	84
3.5.7 Postventa.....	86
3.6 Mantenimiento y Desarrollo de Nuevas Funcionalidades	87
3.6.1 Objetivo.....	87
3.6.2 Actividades	88
3.7 Control de Calidad.....	91
3.7.1 Objetivo del Control de Calidad.....	91
3.7.4 Actividades	91
Capítulo 4 Evaluación del Resultados Globales y Conclusiones	93
4.1 Evaluación de Resultados.....	93
4.2 Conclusiones	93
4.3 Mensaje Final	98
Referencias Bibliográficas	99

RESUMEN

Saber lo que se tiene que hacer no necesariamente implica el saber cómo hacerlo. La transición entre Teoría y Práctica dentro del ámbito de la Ingeniería consiste en adquirir un conjunto de destrezas intelectuales, junto con el conocimiento para hacer el correcto uso de las herramientas disponibles, y utilizar adecuadamente algoritmos y organización de tareas.

El presente trabajo es una de narrativa sobre la **Aplicación** de la Ingeniería de Software, donde se describen a detalle la manera en que logramos obtener todos y cada uno de los puntos mencionados en el párrafo anterior, pero sobre todo haciendo especial énfasis en describir *las Técnicas* utilizadas durante este proceso. Todo esto puede resultar altamente útil para quien inicie sus pasos en esta interesante profesión.

La **Ingeniería de Software** es la rama de la ingeniería que crea y mantiene las aplicaciones de software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos.

Podemos definirla como "*la aplicación inteligente de principios probados, técnicas, lenguajes y herramientas para la creación y mantenimiento, dentro de un coste razonable, de software que satisfaga las necesidades de los usuarios*". El criterio más común para distinguir al ingeniero de software es el conocimiento y aplicación de las técnicas y herramientas de las metodologías de la Ingeniería del Software, o dicho de otra manera, la técnica empleada en la Metodología de Desarrollo de Sistemas.

La ingeniería de software se puede considerar como la ingeniería aplicada al software, esto es en base a herramientas preestablecidas, la aplicación de las mismas de la forma más eficiente y óptima; objetivos que siempre busca la ingeniería. No es solo de la resolución de problemas, sino más bien teniendo en cuenta las diferentes soluciones, elegir la más apropiada. El objetivo es encontrar procesos o metodologías que mejoren la productividad y la calidad.

La ingeniería de software requiere llevar a cabo muchas tareas, que se agrupan a su vez en etapas del proceso que van desde el **Análisis, Desarrollo, Implementación y Mantenimiento**, es decir las Fases de la Vida útil de nuestro sistema. Dentro de las actividades más destacadas que abordaremos en cada una de las Fases de esta metodología propuesta para el desarrollo de sistemas tenemos las siguientes:

- **Análisis de requisitos.** Extraer los requisitos de un producto de software es la primera etapa. Se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado se plasma en el documento ERS, Especificación de Requerimientos del Sistema.
- **Especificación.** Es la tarea de describir detalladamente el software a ser escrito, en una forma matemáticamente rigurosa. En la realidad, la mayoría de las buenas especificaciones han sido escritas para entender y afinar aplicaciones que ya estaban desarrolladas.
- **Diseño y arquitectura.** Se refiere a determinar como funcionará de forma general sin entrar en detalles. Consiste en incorporar consideraciones de la implementación tecnológica.
- **Programación.** Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no es necesariamente la porción más larga.
- **Pruebas.** Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral.

- *Documentación.* Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema.
- *Mantenimiento.* Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software.

La Plataforma Operativa que utilizaremos para ilustrar nuestro ejemplo es la arquitectura de **Cliente-Servidor**. Esta arquitectura consiste en que un programa el Cliente informático realiza peticiones a otro programa, el servidor, que les da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita el diseño del sistema. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta sobre una sola máquina ni es necesariamente un sólo programa.

Como ventajas de esta arquitectura cliente-servidor tenemos: *Centralización del control:* los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Todos los datos se almacenan en los servidores, así que tienen capacidad mejor del control de la seguridad. El servidor puede controlar el acceso al cerciorarse que dejó solamente esos accesos de usuarios permitidos y cambia datos. *Escalabilidad:* se puede aumentar la capacidad de clientes y servidores por separado. Hay las tecnologías maduras diseñadas para el paradigma de cliente servidor que asegura seguridad, interfaz amigable para usuarios con poca pericia, y por ende la *Facilidad de empleo*. Es más *Flexible* para poner al día los datos u otros recursos.

El área aplicación de nuestro sistema será la **Administración Integral** de un negocio ejemplo. Las prioridades de nuestro programa serán las de satisfacer las necesidades de toda la gente alrededor de manera que se aumente la productividad. Por un lado para los dueños o accionistas tendrán mayor control. Los contadores o gerentes tendrán en el sistema una ayuda en la toma de decisiones. Para los empleados facilitará las labores operativas. Para los clientes dará confianza al tener sus saldos actualizados. Para los proveedores facilitará la emisión de pedidos, mientras que para el gobierno federal emitirá en automático declaraciones y pago de impuestos. De esta manera el sistema deberá convertirse en corto plazo la columna vertebral del negocio que den soporte a las áreas Operativa (Venta de Autos, Servicio y Refacciones) y Administrativa (Contabilidad, Cobranza, Cuentas por Pagar) simultáneamente.

La propuesta es la aplicación de un negocio hecho sobre las **Agencias Automotrices** debido a la experiencia que he adquirido en este sector. Desde el año 2004 he participado en la creación de software para la venta de autos en más de 20 agencias, lo garantiza que los puntos que vamos a analizar estén debidamente probados y validados bajo situaciones de trabajo con altos niveles de exigencia y presión. Se pretende establecer los principios fundamentales en el desarrollo del software automotriz así como los criterios para medir sus resultados

El trabajo consiste en obtener el conocimiento profundo de los procesos operativos que se desarrollan en la agencia automotriz: La venta de autos; las operaciones de servicio; la venta e inventario de refacciones; la contabilidad; y las cuentas por cobrar y por pagar. El análisis conceptual la agencia automotriz, así como su problemática actual, y también el futuro de los sistemas de cómputo utilizados en la administración de la agencia y el rumbo que se deberá tomar al desarrollar nuevos sistemas dentro de nuestro contexto globalizado.

1. INTRODUCCIÓN

1.1 EXPERIENCIA PROFESIONAL – ADMINISTRADOR / DESARROLLADOR

El presente informe tiene como objetivo general compartir con las actuales generaciones de estudiantes de Ingeniería en Computación el panorama real con el cual habrán de enfrentarse en su futuro desempeño profesional. Se trata pues de compartir mi experiencia profesional adquirida a través de 5 años de servicio en dos importantes empresas.

La transición de Estudiante a Profesionista puede convertirse en un camino sinuoso y lleno de obstáculos para los cuales no se nos prepara en las aulas. Así pretendo darle a este informe un enfoque de “avanzada de exploración”, de tal suerte que quien la lea pueda tener una clara visión del ámbito laboral de un Ingeniero en Computación en dos de los puestos más importantes en el medio informático: La administración y el desarrollo de Sistemas.

Una vez egresado de la FES Aragón¹ mi primera experiencia profesional fue como Administrador de Sistema en Nacional Financiera S.N.C.² del 1 de Julio del 2002 al 31 de Marzo del 2004. Nafinsa³ como agente financiero del gobierno federal es la institución encargada de administrara los créditos que México obtiene de los organismos financieros internacionales. Mi actividad principal en Nafinsa fue operar el sistema de administración de créditos. El puesto se trataba de lo siguiente:

Un **administrador** de sistema (en inglés abreviado sysadmin) es el trabajo de la persona que tiene la responsabilidad correr y asegurarse el correcto funcionamiento de un sistema informático, o algún aspecto de éste. El significado preciso varía. En las organizaciones con un sistema muy grande y complicado, generalmente dividen al personal informático según su especialidad. En este caso un administrador de sistema es aquel responsable del mantenimiento de un sistema informático existente. Deberes como administrador de sistema:

- Realizar copias de seguridad.
- Actualizar el sistema operativo, y configurar los cambios.
- Instalar y configurar el nuevo hardware y software.
- Agregar, borrar y modificar información de las cuentas de usuarios, contraseñas, etc.
- Responder consultas técnicas.
- Responsable de la seguridad.
- Responsable de documentar la configuración del sistema.
- Resolución de problemas.
- Configuración óptima del sistema.
- Implantación de DRP's⁴.

En grandes organizaciones, algunas de las tareas listadas arriba se dividen entre diferentes administradores de sistema. Por ejemplo, debe haber un individuo o un grupo responsable de probar e instalar las nuevas actualizaciones. En organizaciones más pequeñas, un administrador también puede tener responsabilidades asociadas a otros campos, son:

- Soporte técnico.
- Administrador de base de datos.
- Administrador de red.
- Analista de sistemas.
- Administrador de seguridad.

¹ Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Aragón.

² Nacional Financiera, Sociedad Nacional de Crédito.

³ Nacional Financiera

⁴ Disaster Recovery Plan, ó Plan de Recuperación de Desastres

- Programador.

Los administradores de sistema, en grandes organizaciones, tienden a no ser arquitectos de sistemas, ingenieros de sistemas, ni diseñadores de sistemas. Como sucede con muchos roles de este campo, en las pequeñas organizaciones las diferencias de roles no están muy delimitadas. Como sea, incluso en grandes organizaciones, los administradores de sistema titulares tienen conocimiento en varias de estas otras áreas, como resultado de su amplia experiencia laboral.

Convertirse en un administrador de sistema. A diferencia de muchas profesiones, no hay sólo un camino para convertirse en administrador de sistema. Algunos colegios o universidades tienen programas específicos para administradores de sistemas. Por eso muchos administradores de sistemas tienen un grado de conocimiento de campos generales: ciencias de la información, ingeniería de software, administración de sistemas informáticos, etc. En el 2004 sólo dos universidades de los Estados Unidos de Norteamérica, Tufts y la Universidad Estatal de Florida, tenían un programa de postgrado en administración de sistemas.

Además, dada la naturaleza práctica de los administradores de sistema, y de la disponibilidad de programas de servidores de código libre, muchos administradores ingresan a este campo de forma autodidacta. Generalmente, se requiere una experiencia previa con el sistema que se espera administrar.

La segunda experiencia profesional que tuve fue en Sistemas Automotrices Proactive, S.A. de C.V. con el puesto de Desarrollador de Sistemas para la industria automotriz durante el periodo del 1 de Abril del 2004 al 31 de Enero del 2007. El perfil del puesto es el siguiente:

Desarrollador. Un desarrollador es un informático que programa aplicaciones en distintos lenguajes de programación informáticos. El concepto de desarrollo incluye:

- un trabajo en equipo: los proyectos son en general una colaboración entre varios desarrolladores, que tratan cada uno una parte del programa, y también de otros colaboradores como los comerciales, que definen con el cliente la finalidad del producto, diseñadores gráficos que definen el aspecto y la ergonomía.
- la concepción (diseño): a partir de un pliego de condiciones (user requirement specifications⁵), definir las especificaciones técnicas (estructura de datos, comunicación entre los módulos)
- las pruebas, que sirven para detectar las no conformidades y los errores
- el mantenimiento: la corrección de los errores después de la salida del programa informático, y la mejora para hacer evolucionar el producto.

Así, con la visión y panorama obtenidos en estos dos puestos a los largo de estos años de ejercicio profesional, trataré de explicar a detalle todo el proceso mediante el cual se generan sistemas informáticos en México, haciendo uso de un caso práctico (software para una agencia de autos) explicaremos la manera en que se hace, así como los problemas enfrentados y las experiencias obtenidas. En lo personal considero que esto puede resultar bastante interesante para quien lo lea, ya que proporcionará una perspectiva de los problemas que puede enfrentar a fin de que esté preparado con una solución a este tipo de problemas comunes, como los que se mencionan a continuación.

⁵ Especificación de requerimientos de los usuarios del sistema

1.2 PROBLEMÁTICA ENFRENTADA EN LA EXPERIENCIA PROFESIONAL

El problema de coordinar el uso de Software Multiusuario en México.

Una empresa Española radica en Santander trajo a México su exitoso **DMS**⁶ de administración de agencias automotrices. La *seguridad* con la que ellos vendieron su producto hacia suponer que todo iba a salir perfecto. El software se usaba sin problemas en España, Portugal y Alemania, con resultados satisfactorios. La primera agencia seleccionada para implantar su software fue Acura Pedregal. Una vez pasado el furor inicial por el novedoso producto y realizados los análisis más a fondo de las particularidades del sistema, se encontraron con que el sistema español no tenía un módulo de “cuentas por cobrar”.

Los usuarios del sistema cuestionaron con cierta preocupación el porqué le hacia falta este módulo, preguntando directamente al líder de proyecto ibérico. La respuesta del líder de proyecto ibérico dejó perplejos a todos - ¿Qué es cuentas por cobrar?... Parecía una broma de pésimo gusto, sin embargo era real. El sistema no tenia módulo de Cobranza, y no solo eso, ni siquiera sabían de lo que se trataba.

Con la seguridad de quien tiene dominada la situación el desarrollador español pidió que le explicasen qué era cuentas por cobrar. Sobrepuestos al espanto inicial se le explicó que es un módulo de reportes para saber cuanto nos deben los clientes y por ende saber como cobrarles... A lo que respondió de una manera muy cándida. *“A los clientes no es necesario cobrarles, ellos solitos depositan sus cheques en las cuentas bancarias, además de que actualmente ya casi no hay ventas a crédito.”*

Este claro ejemplo nos permite ilustrar que no siempre lo que para otros funciona tenga que funcionar para nosotros, en un país desarrollado no es frecuente tener problemas con los pagos, pero en un país como el nuestro la mayoría de clientes con morosos. Así estas *diferencias de idiosincrasia deberán considerarse muy seriamente al construir productos informáticos.*

México, un país surrealista.

Y es que hablar de desarrollo de sistema para empresas mexicanas implica un profundo conocimiento del medio Nacional y el contexto en que el usuario mexicano de sistemas lleva a cabo su actividad diaria. Alguien alguna vez cito a México como un país surrealista, por la manera en que se llevan a cabo las cosas en este país, la forma en que a veces nos obstinamos en hacer las cosas al revés. Para nosotros es común ver autos estacionados en las banquetas y salas de exhibición de negocios a media avenida. Es duro admitirlo pero ese desorden que tiene la gente en su vida cotidiana, es común que lo trate de llevar a todos los ámbitos de su trabajo, incluyendo a los sistemas que opera.

Manejo de software multiusuario en el país surrealista – Una riesgosa profesión.

Es por este contexto especial que al desarrollar sistemas en México se deben de tomar en cuenta consideraciones especiales para el tipo de usuarios con los que trabajamos. Es común pues encontrar gente sin capacitación en cómputo, que tiene esos puestos porque son gente de confianza de la gerencia; o es común encontrar malas prácticas administrativas que siguen persistiendo porque la gente alega que “siempre se ha hecho así”

⁶ Document Management System - Sistema de Gestión de Documentos

Usuarios mexicanos de sistemas.

Un factor que puede echar a perder un sistema de cómputo completo si no se tiene cuidado es el usuario de sistemas, ya que a veces tienden a culpar al sistema por que no quedó el trabajo en lugar de admitir que ellos mismos no hicieron bien las cosas. El usuario es a su vez el peor enemigo del sistema. Vamos tratar de explicarlo de esta manera:

El contador sugiere que el sistema esta mal porque en las órdenes de servicio no se le puede agregar un costo de 20 pesos de estopa a cada orden de servicio, por lo cual pide que se modifique para que se puedan agregar esos 20 pesos. Una vez modificado sugiere que esos veinte pesos se apliquen a una partida contable diferente, por lo cual deberá modificar el registro contable a través de una póliza para que se incluyan en una partida especial esos 20 pesos. Una vez modificada la póliza se nota que en reporte de órdenes hace falta una columna que muestre cuánto se cobró por la estopa para cada orden, considerando que en las garantías no se cobra y en las órdenes de taller publico si se cobra. Una vez modificado este reporte, se encuentra que ahora este reporte no cuadra con la balanza contable, por la falta de 20 pesos, por lo cual se deberá modificar...

Este tipo de ejemplos ilustra de manera detallada lo que sucede en la realidad con los sistemas. Lo que al principio parece una modificación insignificante termina creciendo como una bola de nieve hasta llevarnos a incongruencias en el sistema al no cuadrar las cifras, lo cual pudimos haber omitido si no hubiéramos permitido esa modificación.

Desafortunadamente, este tipo de problemas no son los únicos a los que los ingenieros de software nos debemos enfrentar. Desde usuarios sin capacitación “recomendados”, computadoras en mal estado, redes de cómputo mal instaladas, constante rotación de personal y por ende cambio de procesos operativos, nula capacitación teórica, usuarios que no conocen los procesos, desorganización administrativa, etc. son factores con los cuales se tiene que lidiar y que no siempre se detallan en las teorías acerca del diseño de sistemas.

1.3 SOLUCIÓN DE PROBLEMAS

Todos estos problemas mencionados referentes al desempeño de los sistemas difícilmente se abordan en las aulas universitarias; y frecuentemente se nos presentan de manera imprevista. La falta de pericia para solucionar este tipo de dilemas puede traducirse en resultados desastrosos, y en el peor de los casos, puede terminar con el sistema mismo. Considerando que los sistemas actuales son dinámicos y en constante cambio, el correcto desempeño de las funciones del binomio administrador del sistema y desarrollador serán la clave del éxito en la operación diario de cualquier sistema de cómputo multiusuario. Haciendo una analogía de la operación de un sistema de cómputo, podemos verlo como un equipo de carreras de autos, al Administrador como el piloto, y el desarrollador como el mecánico. El talento de ambos se conjugará ya sea para el éxito o el fracaso del sistema.

Clave del Éxito: El administrador y el desarrollador, Analogía Piloto / mecánico

Sin embargo, a pesar de todos los problemas mencionados, podemos afirmar que es factible lograr el éxito en la implantación de un sistema de muchos usuarios, pero se deberá considerar el talento de las dos personas más importantes en la correcta operación del sistema: El administrador y el desarrollador. Si esto de operar sistemas fuera una carrera, se trataría del piloto y el mecánico, y del talento combinando de ambos dependerá que el sistema llegue con éxito al final de la carrera.

1.4 OBJETIVOS DE ESTE INFORME

a) Informar cómo funciona una empresa mexicana de IT⁷. Mostrar la manera en que está estructurada una “Fábrica de Software” y las áreas que la constituyen, así como la manera en que opera en las fases de su proceso, como lo son el Análisis, Diseño, Construcción, Implantación, Helpdesk⁸, Mantenimiento y Control de Calidad.

b) Enumerar las mejores prácticas encontradas en la industria automotriz que garanticen el óptimo funcionamiento de una agencia y por ende el éxito de nuestro sistema de información. Dentro de este objetivo particular explicaremos la manera de identificar las trampas en las que en un momento dado se pueden caer, para evitarlas, y si por la inexperiencia se cae en ellas, se explicará a la manera de salir.

c) Identificar factores externos a la metodología de desarrollo que inciden directamente sobre el éxito o fracaso del sistema. La teoría dista muchos de realidad. Así pues tenemos una enseñanza donde los métodos se aplican sin mayores contratiempos. Si para construir y mantener los sistemas solo fuera necesario seguir al pie de la letra la metodología escrita para ello, cualquiera lo haría. Sin embargo, el ingeniero mexicano debe ir más allá en la búsqueda de soluciones a problemas que aun no se han planteado. Éste trabajo de investigación más que enseñar a seguir los métodos establecidos, enseña la manera de crear métodos sistemáticos que optimicen los sistemas.

d) Analizar la manera en que la enseñanza recibida incide de manera decisiva en el desempeño profesional. Tengo esta inquietud de presentar este trabajo de esta manera, ya que pienso que conlleva la esencia que he recibido a través de toda mi formación en la UNAM. Desde mi formación en el CCH⁹, fueron inculcados en mis valores como la curiosidad, el cuestionamiento, la búsqueda de diferentes maneras para obtener resultados satisfactorios, la promoción de mi propio aprendizaje. Esta manera de visualizar mi desarrollo profesional, la universidad dejó en mi una huella indeleble, y una manera de conceptualizar los problemas que ha repercutido de manera directa en mi ejercicio profesional. El administrador y el desarrollador son los responsables de que un sistema funcione óptimamente. Y esa responsabilidad los lleva a estar constantemente en un proceso de reingeniería que hace más eficiente el uso de sistema, así como la interacción con usuarios para explotar al máximo el potencial del software.

e) Compartir la experiencia de cómo solucionar los problemas enfrentados. Este informe de trabajo es una manera de compartir con futuras generaciones de ingenieros aragoneses, las primeras experiencias obtenidas en la ejecución de mi ejercicio profesional, un compendio de toda la experiencia adquirida todos estos años en importantes empresas (Nafinsa, Ford), para encontrar la manera de cómo solucionar los problemas presentados como administrador / desarrollador de sistemas, que sirvan de guía y apoyo a los recién egresados a encontrar atajos y soluciones a problemas para los cuales no vemos en las aulas, y no imaginamos pudieran presentarse, pero que si no son resueltos de manera eficiente y simple, pueden llegar a echar por la borda sistemas en los cuales se les han invertido años de trabajo y millones de presupuesto. Siento que tengo la autoridad para hablar del tema, ya que desde 2002 he estado en el ojo del huracán de los sistemas, de manera operativa, y en el desempeño de mi labor profesional he recopilado métodos, atajos y estrategias mismas que servirán como punto de partida para establecer lineamientos en la programación y mantenimiento de sistemas.

⁷ Information Technology – Tecnologías de la Información.

⁸ Helpdesk – Atención a Clientes.

⁹ Colegio de Ciencias y Humanidades

2. SOFTWARE AUTOMOTRIZ EN MÉXICO

El Dinamismo con el que hoy en día se desenvuelven diariamente empresas de todos tipos, hace impensable de que se pueda subsistir sin un sistema de cómputo centralizado como el centro neurálgico que controle las operaciones cotidianas de cualquier empresa.

Ya se trate de una simple gasolinera, una línea aérea o de auto transportes, un almacén de compra venta de productos o un consultorio médico, la férrea competencia actual en un este mundo globalizado requiere de aplicaciones cada vez más potentes para estar a la altura de la movilidad de los procesos operativos actuales.

Una aplicación bastante interesante de la ingeniería de software se encuentra en la construcción de sistemas para agencias automotrices, dado la industria automotriz en México por su tamaño es una de las más importantes del mundo.

Por la importancia de esta industria en México y por la amplia experiencia obtenida en este sector, hemos elegido revisar a detalle el proceso de creación de software para las agencias automotrices de venta de autos, de tal manera que quien lea este trabajo encuentre en él un herramienta que le facilite la comprensión de las actividades involucradas en el Desarrollo de Software.

Es necesario precisar que aunque el trabajo se basa principalmente en la aplicación de sistemas en agencias automotrices, la visión con la que se hace este trabajo es establecer una metodología estándar que pueda usarse en la creación de cualquier tipo de sistemas cliente servidor.

La características principal de los sistemas que analizaremos en la centralización en bases de datos relaciones de la información operativa de empresas, principalmente dedicadas a la compra y venta, ya sea de productos o servicios, y que primordialmente pueden ser operadas por cientos de usuarios de manera simultanea.

2.1 PANORAMA ACTUAL DE LA INDUSTRIA AUTOMOTRIZ

Establecido el objetivo de analizar la creación de sistemas para agencias automotrices a manera de ejemplo para la creación de otros tipo de sistemas similares, vamos a comenzar con echar un vistazo a la actualidad de la industria automotriz en México, dado que el conocimientos previo que podamos obtener sobre el tipo de empresa en la que vamos a trabajar, será de vital importancia para crear aplicaciones que se adapten perfectamente a las necesidades de las empresas.

Muchas veces se comete el error de crear sistemas que se adapten a la Teoría que hay escrito sobre ellos. Actualmente es necesario trabajar al revés, es decir, hacer que la teoría se adapte a la realidad y no tratar de adaptar la realidad a la teoría. Así se nos facilitará la visión y los objetivos primordiales del sistema en base a sus necesidades actuales, lo cual será una gran ventaja tanto en la vida útil del sistema, así como cuando nuestro sistema se tenga que comparar contra los de la competencia.

De esta forma, en México tenemos una Industria Automotriz estructurada de esta manera:

2.1.1 EL AUTOMÓVIL

El automóvil es un vehículo destinado al transporte de personas, generalmente de cuatro ruedas y con capacidad entre uno y nueve asientos. Se desplaza gracias a un motor de explosión a base de una mezcla de gasolina, GLP¹ o de gasoil y aire. En algunos países la mezcla detonante se fabrica a partir de etanol (maíz), y para el ciclo Diesel de aceite (soja) de determinados vegetales. Recientemente se han comenzado a producir automóviles que funcionan con motor eléctrico, si bien la autonomía de estos vehículos es aún limitada debido al peso de las baterías. Las ruedas delanteras de los automóviles pueden moverse hacia ambos lados para realizar giros y tomar las curvas.

2.1.1.1 Historia

El automóvil, tal como lo conocemos en la actualidad, fue inventado en Alemania en 1886 por Carl Benz. Poco después otros pioneros, como Gottlieb Daimler y Wilhelm Maybach, presentaron a su vez sus modelos. El primer viaje largo en un automóvil lo realizó Bertha Benz en 1888, al ir de Mannheim a Pforzheim, ciudades separadas entre sí por unos 105 km. En 1910, Henry Ford comenzó a producir automóviles en una cadena de montaje, sistema totalmente innovador que le permitió alcanzar cifras de fabricación hasta entonces impensables. El automóvil alteró no sólo la forma de vivir, las ciudades y el paisaje, sino que se convirtió pronto en el producto industrial por excelencia. Más de 2.500 fábricas grandes y pequeñas han producido automóviles desde sus comienzos. Muchas de ellas fueron industrias metalmeccánicas, de armas y de bicicletas, y apostaron por el nuevo producto con base en sus experiencias de fabricación en aspectos técnicos útiles para producir automóviles.

2.1.1.2 Impacto Económico y Social del Automóvil

◆ CAMBIOS ECONÓMICOS

- ☞ **Reestructuración de la industria.** Se crearon enormes industrias para producir automóviles. La gasolina floreció de una manera tan importante que los gobiernos tomaron medidas para asegurar un flujo constante del aceite. La industria de acero creó enormes negocios. El producto químico, el caucho, y las industrias petroleras fueron rehechos para satisfacer las necesidades del automóvil y las industrias originaron, por ejemplo estaciones de gasolina, los moteles, y el seguro de automóvil.
- ☞ **Infraestructura.** Uno de los efectos más visibles que el automóvil ha tenido en el mundo es el aumento enorme en la cantidad de carreteras. Los caminos fueron pavimentados con asfalto, y las autopistas llegaron a ser cada vez más comunes.

◆ CAMBIOS TECNOLÓGICOS

- ☞ **Producción.** Las líneas de ensamble y otros métodos de producción en masa fueron desarrollados para buscar maneras de construir más automóviles a precios más bajos.

◆ CAMBIOS AMBIENTALES

- ☞ **Contaminación.** Los automóviles son una fuente importante del bióxido de carbono que está causando un cambio global climático. El número total de los autos en el planeta era cerca de 600 millones de 2006.

◆ CAMBIOS CULTURALES

- ☞ **Cambios en la Sociedad Urbana.** En los años 40, la mayoría de los ambientes urbanos perdió sus tranvías, calandrias, y otras formas de transporte ligero, para ser substituido por los coches o los autobuses. proporciones epidémicas.
- ☞ **Advenimiento de la sociedad suburbana.** Debido a el automóvil, se aceleró el crecimiento exterior de las ciudades, y los suburbios comenzaron a crecer rápidamente.

¹ Gas Licuado del Petróleo

- 🔊 **Cultura del Auto.** El coche tenía un efecto significativo en la cultura de la clase media. Los automóviles fueron incorporados en todas las partes de vida de la música a los libros a las películas. Los temas automotores de estas canciones reflejaron la cultura general de la industria del automóvil: aventura sexual, liberación del control social, y energía masculina.
- 🔊 **Cambios a la forma de vida individual.** Hacia el final del siglo 19, se puso énfasis en la libertad personal y la movilidad individual. El automóvil abarcó ambos ideales.
- 🔊 **Estatus Social.** El automóvil significa mucho más a muchos que simplemente un modo del transporte. el coche sirvió como marca de la distinción.
- 🔊 **Recreación.** La creación autopistas y automóviles confiables cambió la recreación y las vacaciones. Una vez que el automóvil llegó a ser abundante, Los recursos aparecieron en lugares escénicos, lejos de la vida agitada de las ciudades.
- 🔊 **Seguridad.** Los accidentes del automóvil causaron muchas muertes y lesiones. A esta fecha, los automóviles siguen siendo una causa importante de la muerte accidental y lesión, no mencionar la tensión emocional.

2.1.1.3 El futuro del automóvil

El futuro del automóvil reside en la no contaminación de éste, debido a que uno de los mayores problemas actuales esta en la contaminación. Serán vehículos que se moverán sin necesidad de gasolina, sino de otras fuentes como puede ser el hidrógeno, con el cual se está experimentando actualmente y ya se han logrado resultados. Aparte, se diseñaran o nuevas formas de moverse más rápidas o mejores carreteras por las que moverse. Incluso los coches podrán volar.

Otro posible futuro del automóvil es su desaparición o sustitución por medios de transporte colectivos más eficientes energéticamente. Esto puede suceder principalmente a causa del escaseo de petróleo barato y abundante como consecuencia del agotamiento de este combustible fósil.

Las alternativas como el biodiesel y el etanol dejan al aire muchas dudas, primero por su competencia directa con las tierras de cultivo para alimentar a las personas, y segundo por la falta de estudios serios que miren su viabilidad energética. El hidrógeno se ha de fabricar por medio de electrólisis, y si actualmente las energías renovables solo son un 2,8% de la energía mundial, no esperemos que puedan sustituir a los fósiles que son el 80%.

2.1.1.4 El auto símbolo de status

Encuesta revela que quienes manejan BMW y Audi tienen mayor actividad sexual. Hay un mito, reforzado por la publicidad, que dice que uno conquista a las mujeres según el valor del auto que maneja. ¿Es un mito o una realidad? ¿Cuán seductor puede ser un auto? Esa respuesta acaba de ser cuantificada por una revista alemana, Men's Car, que preguntó a 2,253 conductores y lectores entre 20 y 50 años de edad... *En nuestra sociedad el auto es herramienta de seducción y objeto de deseo.* Brinda la ilusión de status, clase y poder. "La gente encuentra su alma en el automóvil" (H. Marcuse, El hombre unidimensional).



Imagen 1. El auto: "símbolo de status"

2.1.2 LA INDUSTRIA AUTOMOTRIZ EN EL MUNDO

2.1.2.1 Fabricantes y Marcas de Automóviles

Por fabricantes de automóviles se entiende las compañías que diseñan, desarrollan, fabrican y prueban los automóviles bajo una propia identidad, ya sean empresas independientes o integradas en un grupo mayor.

Marcas de automóviles

Todas aquellas compañías que constituyen o que han constituido en el pasado la identidad de un fabricante o de una división o línea de modelos de automóviles.

Alfa Romeo	Aston Martin	Audi	Bentley	BMW
Bugatti	Cadillac	Chevrolet	Chrysler	Citroën
Dacia	Daewoo	Daihatsu	Dodge	Ferrari
Fiat	Ford	GM	Honda	Hummer
Hyundai	Infiniti	Isuzu	Jaguar	Jeep
Kia	Lamborghini	Lancia	Land Rover	Landwind
Lexus	Lotus	Maserati	Maybach	Mazda
Mercedes	MG Rover	Mini	Mitsubishi	Morgan
Nissan	Opel	Peugeot	Pontiac	Porsche
Renault	Rolls-Royce	Saab	Santana	Seat
Skoda	Smart	Ssangyong	Subaru	Suzuki
Tata	Toyota	Volkswagen	Volvo	Dina

2.1.2.2 Producción de Vehículos en el Mundo

Durante 2005, se produjeron **66,465,408** de automóviles, camiones ligeros y camiones pesados en el mundo. La región de mayor producción de vehículos nuevos en el mundo es Asia-Oceanía, integrada por 13 países, los cuales fabrican más de 25 millones de unidades al año, es decir representa casi el 39% de la producción mundial. Es preciso destacar en esta región la dinámica de China, país que, a pesar de ser el segundo en importancia en la región, con más de 5.7 millones de unidades (solo superado por Japón) creció 9% en 2005. En segunda posición a nivel global se encuentra la Unión Europea, región que manufactura más de 18 millones de vehículos cada año, lo que representa poco más del 27% del total mundial, aunque habrá que decir que son 25 los países participantes. Es preciso comentar que, al igual que en el caso de Asia, también en este grupo existe una clara concentración de la producción en pocos oferentes, de tal suerte que sólo 5 países (Alemania, Francia, España, Reino Unido e Italia) generan el 82% de la oferta de la región.

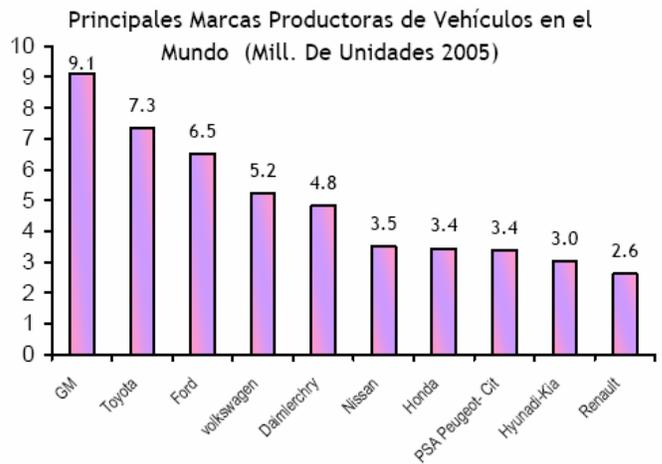
En el caso América, el volumen de más de 19 millones de vehículos que produce anualmente, lo coloca en una posición de 29% a nivel mundial; aunque habrá que destacar que sólo la región del TLCAN² (E.U.A., Canadá y México), produce más de 16 millones de unidades al año, lo que representa el 25% de la producción mundial y de manera particular, E.U.A. constituye el centro de producción y consumo más importante del mundo por sus casi 12 millones de ensamble y más de 16 millones de ventas internas cada año. Dada la importancia de E.U.A., el desempeño de su mercado tiene efectos inmediatos en el resto de la región, de tal suerte que debido a la contracción de la economía de ese país en este año, la producción de la región TLCAN podría registrar un descenso de poco más de un cuarto de punto al finalizar 2006, ello a pesar de la recuperación México, en materia de ensamble, desde finales de 2005.

Producción de Vehículos Automotores en las Principales Regiones del Mundo (Unidades)

Región	2004	2005	Var. %
Asia-Oceanía (13 países)	24,291,548	25,817,187	6.3
Unión Europea (25 países)	18,330,912	18,176,860	-0.8
América del Norte (3 países)	16,278,082	16,339,678	0.4
América del Sur (7 países)	2,669,223	2,984,813	11.8

² Tratado de Libre Comercio de América del Norte

Marcas. Conviene identificar cuál es la participación por marca en la oferta automotriz mundial, variable que tiene también un alto grado de concentración, ya que, si bien se tienen registradas 48 grandes marcas (o grupos), sólo 10 generan el 74% de la producción mundial. Las 10 Top de la producción son las tres americanas (GM, Ford y Dailmerchrysler, aunque esta última es producto de una fusión con una marca alemana), 4 asiáticas (Toyota, Nissan, Honda y Hyundai-Kia) y 3 europeas (Volkswagen, PSA Peugeot-Citroën y Renault)



La información al cierre de 2005 indica que General Motors se mantiene como líder mundial con, una producción de más de 9 millones de vehículos al año, lo que representa el 13.7%. Los casi 7.4 millones de vehículos producidos por Toyota en 2005, lo colocan como el segundo proveedor mundial, lo que le da una participación de mercado de 11%, seguido por Ford, con 6.5 millones de vehículos, que representan el 9.8% de la oferta. Volkswagen y Daimlerchrysler ocupan los lugares 4 y 5 del ranking mundial, con 7.8% y 7.2% de participación mundial. Nissan, Honda y PSA Peugeot-Citroën aportan aproximadamente el 5% de la oferta mundial, cada una; mientras la asociación Hyundai-Kia producen el 4.5% del total y Renault el 3.9%.

2.1.2.3 México dentro del contexto mundial

En el 2005 fueron producidos 63 millones de autos y camiones ligeros en el mundo.

	1,000	2,000	3,000	4,000	5,000	6,000	7,000	8,000	9,000	10,000	11,000	12,000
United States												11,524
Japan										10,064		
Germany						5,543						
China					5,067							
South Korea				3,657								
France			3,495									
Spain			2,677									
Canada			2,624									
Brazil			2,375									
United Kingdom		1,783										
Mexico		1,607										
India		1,406										
Russia		1,264										
Thailand		1,110										
Italy		995										

Imagen 2. TOP 15 de países productores de autos en el Mundo.

Las grandes áreas de libre comercio como la Unión Europea, el TLC³ y el MERCOSUR⁴ atraen a los fabricantes mundiales para producir sus productos debido a sus reducidos costos actuales y controles personalizados, además de estar cerca de sus clientes. En muchos casos, si no es que en la mayoría, la producción de autos en el Tercer Mundo usan tecnología Occidental y modelos de autos, y a veces fábricas completas occidentales desembarcan en esos países.

³ Tratado de Libre Comercio de América del Norte

⁴ Mercado Común del Sur

2.1.3 LA INDUSTRIA AUTOMOTRIZ EN MÉXICO

2.1.3.1 Producción Nacional de Automóviles

Año	Producción Nacional		
	Autos	Camiones Ligeros	Total
2005	976,721	611,927	1,588,648
2004	903,313	601,819	1,505,132
2003	913,309	623,970	1,537,279
2002	1,139,599	625,502	1,765,101
2001	1,125,145	471,306	1,596,451
2000	1,278,839	610,397	1,889,236
1999	993,772	499,948	1,493,720
1998	952,909	474,684	1,427,593
1997	854,809	483,193	1,338,002
1996	797,682	413,648	1,211,330
1995	699,312	231,870	931,182
1994	856,563	240,818	1,097,381
1993	835,090	224,803	1,059,893
1992	776,185	275,001	1,051,186
1991	720,384	240,499	960,883
1990	598,093	205,778	803,871

Asciende a 8 mil mdd inversión automotriz en los últimos seis años

Informa la Secretaría de Economía que generó en el periodo de 2001 a 2006 un millón de empleos totales; aumentaron las ventas 117.7% con respecto al ciclo 1995-1999. Durante el periodo 2001-2006 la inversión directa en la industria automotriz fue de 8 mil millones de dólares, y los empleos totales ligados al sector ascendieron a un millón 60 mil informó la Secretaría de Economía. En dicho lapso las ventas totales del sector en el mercado interno aumentaron 117.7%, con respecto al ciclo 1995-1999, en tanto que las exportaciones lo hicieron 29.2% y la producción total creció 28.8%. Precisó que tan sólo de enero a octubre de 2006 la industria automotriz alcanzó una cifra récord de un millón 655 mil unidades producidas, mientras que en el mismo periodo las unidades exportadas fueron de un millón 271; cifra sin precedente. Agregó que en 2005 la industria automotriz mexicana alcanzó un superávit comercial de 10 mil millones de dólares.

Otras consideraciones sobre la industria China

China es más atractiva que México si se habla del costo laboral, ya que se dice que los obreros especializados en México ganan algo así como 2.5 dólares la hora mientras que en China ganan apenas 0.88 dólares la hora. Esto crea una desventaja para la industria automotriz mexicana, considerando que los costos son fundamentales al momento de la toma de decisiones. No obstante, si se tiene la intención de exportar desde China, entonces su atractivo se ve disminuido al sumarle los costos de logística, los cuales son substancialmente más altos que en México debido a nuestra excelente ubicación geográfica. Además de lo anterior, existen factores culturales que hacen más viable y menos dolorosa la adaptación de los expatriados en nuestro país. Por otro lado, existe una amplia experiencia en el sector mexicano y una excelente mano de obra. ¿Será México capaz de capitalizar estas ventajas? ¿Qué otras ventajas tendrá México?

2.1.3.2 Venta de Automóviles en México

En 1454 Agencias a nivel nacional, se venden más de un millón de Vehículos al año.

MÉXICO

VENTAS MENUDEO DE VEHÍCULOS AUTOMOTORES NUEVOS

Año	Autos	Camiones Ligeros	Pesados	Total
2005	713,379	415,974	31,837	1,161,190
2004	741,667	351,394	26,346	1,119,407
2003	691,818	283,624	24,273	999,715
2002	712,428	263,822	24,662	1,000,912
2001	667,565	249,652	27,402	944,619
2000	599,285	258,216	33,021	890,522
1999	465,063	209,459	24,449	698,971
1998	430,199	214,564	20,985	665,748
1997	303,558	175,578	15,974	495,110
1996	200,102	125,690	8,431	334,223
1995	117,393	78,527	8,723	204,643
1994	413,819	171,023	28,545	613,387
1993	398,743	183,115	20,822	602,680
1992	445,303	219,297	33,634	698,234
1991	392,110	219,420	25,740	637,270
1990	352,608	173,569	15,944	542,121

2.1.3.3 Reporte Económico Enero Diciembre 2006

Durante el ejercicio 2006, se comercializaron en el País 1 millón 177 mil 248 vehículos, lo que representa un incremento marginal del 1.38 por ciento, respecto al 2005. De ese total, el 96.5 por ciento corresponde a la venta de vehículos ligeros, y el 3.5 por ciento restante a vehículos pesados.

Comercialización de Vehículos Ligeros Comparativo de Ventas entre los períodos Ene – Dic. 2005 y 2006.

Ventas Totales de Vehículos Ligeros en el País

Segmento	Ene - Dic 2005 (1)	Estructura %	Ene - Dic 2006 (2)	Estructura %	Variación (2) / (1)
Sub Compactos	447,703	40	386,799	34	-14%
Compactos	217,168	19	236,884	21	9%
De Lujo	41,949	4	48,676	4	16%
Deportivos	7,190	1	8,587	1	19%
Usos Múltiples	221,484	20	239,370	21	8%
Camiones ligeros	193,804	17	216,555	19	12%
Total	1,129,298	100	1,136,871	100	0.7%

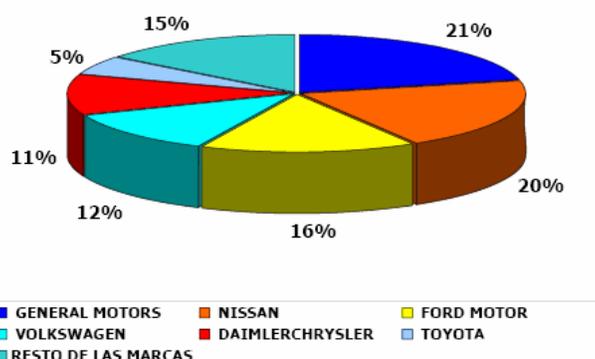
Fuente: Elaborado por AMDA con información de AMIA

A pesar de que el año pasado se registró el mayor crecimiento del ingreso del sexenio, se intensificó y diversificó la oferta de promociones y planes de crédito, y los precios relativos de los vehículos mostraron un comportamiento descendente; el 2006 fue el segundo año de menor crecimiento en ventas internas de todo el sexenio, con una facturación de 1 millón 136 mil 871 unidades, es decir, un incremento marginal de 0.7% respecto al 2005.

Comercialización por Marcas de Ene – Dic de 2005 y 2006 - Unidades

MARCA	ENE-DIC 2005	% DEL TOTAL	ENE-DIC 2006	% DEL TOTAL	DIF % DEL TOTAL
GENERAL MOTORS	249,842	22.22%	245,090	21.57%	-0.65%
NISSAN	234,932	20.90%	228,315	20.10%	-0.80%
FORD MOTOR	190,839	16.97%	177,595	15.63%	-1.34%
VOLKSWAGEN	149,071	13.26%	135,027	11.88%	-1.37%
DAIMLERCHRYSLER	124,353	11.06%	128,446	11.31%	0.24%
TOYOTA	35,318	3.14%	60,088	5.29%	2.15%
HONDA	38,321	3.41%	47,471	4.18%	0.77%
RENAULT	24,086	2.14%	20,274	1.78%	-0.36%
SEAT	21,089	1.88%	19,899	1.75%	-0.12%
MITSUBISHI	15,004	1.33%	16,751	1.47%	0.14%
PEUGEOT	16,963	1.51%	16,068	1.41%	-0.09%
MAZDA	771	0.07%	7,495	0.66%	0.59%
BMW	6,267	0.56%	7,087	0.62%	0.07%
MERCEDEZ BENZ	4,548	0.40%	4,988	0.44%	0.03%
RESTO DE LAS MARCAS*	17,894	1.59%	22,277	1.96%	0.37%
TOTAL	1,129,298		1,136,871		

*Incluye: Lincoln, Suzuki, Audi, Volvo, Mini, Acura, Fiat, Land Rover, Porsche, Smart, Jaguar, Subaru, Bentley, MG/Rover.



Mercado de Vehículos Ligeros Reporte de Ventas 2006 - Resumen

- Contra todos los pronósticos, en el 2006, las ventas decembrinas de vehículos ligeros registraron una disminución de 4 por ciento, respecto al mismo mes del año previo.
- Es la primera vez en los últimos 12 años en que la venta de vehículos ligeros registra una tasa anual decreciente en el último mes del año.
- El segmento más golpeado fue el de los subcompactos, que concentra el 36 por ciento de las ventas totales, con una caída de 23 por ciento respecto a diciembre de 2005.
- Asimismo, en el acumulado anual, el segmento de subcompactos, que concentra a los cinco modelos más vendidos en todo el país, es el único que presenta una disminución respecto al 2005, cayendo de 447 mil 693 a 386 mil 821 unidades durante 2006; es decir, 14% menos.
- En total, la venta de vehículos ligeros presenta un avance marginal de 1.11% durante 2006, en comparación con el año anterior; para quedar en 1 millón 136 mil 761 unidades.
- General Motors mantiene el liderazgo en ventas en territorio mexicano, con una participación de mercado de 21.26 por ciento, seguida de cerca por Nissan con el 20.09 por ciento.
- La marca que reportó una mayor pérdida de participación de mercado fue Ford, al pasar de 17.10 en 2005 a 15.63 en 2006.
- Así mientras Ford perdió 1.47 puntos porcentuales de participación, la alemana VW le siguió de cerca con 1.37 puntos menos entre 2005 y 2006.
- Mientras Toyota, a nivel mundial, amenaza con desbancar a General Motors del primer lugar, en México Toyota ocupa el sexto lugar en ventas, con una participación de 5.29 por ciento durante 2006, 2.22 puntos porcentuales más que en 2005.

2.1.3.4 Panorama actual de las ventas de autos en México

EL MODELO DE EXPERIENCIA DEL CLIENTE AUTOMOTRIZ

La importancia estratégica, para los distribuidores automotrices, de desarrollar relaciones de mediano o largo plazo con sus clientes es cada día más evidente. Ya no se trata sólo de vender un automóvil nuevo o usado, ahora el “nuevo juego competitivo” entre las armadoras y su red de distribuidores es el cómo desarrollar la relación con el cliente y mantenerlo a través de todo el ciclo de vida del automóvil. Es decir, el enfoque transaccional de venta del pasado se convierte ahora en un enfoque de relación con el cliente automotriz. ¿Por qué es importante este enfoque? Si analizamos la tendencia de la utilidad neta promedio de los distribuidores automotrices en los Estados Unidos, encontramos una serie de conclusiones muy importantes:

- ☞ Los negocios de autos nuevos y usados muestran volatilidad, la cual depende de la situación económica del país, el comportamiento de las tasas de financiamiento, la intensidad competitiva, la utilización de incentivos a la venta, etc.
- ☞ Por el contrario, el negocio de servicio y refacciones muestra una tendencia consistente, contribuyendo cada vez más a la utilidad del distribuidor automotriz. En 2005, esta línea de negocio representó casi el 60% de la utilidad neta promedio de los distribuidores.

En resumen, el Modelo de Experiencia del Cliente es una clara ventaja competitiva para las armadoras y los distribuidores. Su instrumentación requiere:

- ☞ Un claro entendimiento de las necesidades y prioridades de los segmentos objetivo de la armadora y de su red de distribuidores
- ☞ La definición de la forma de operación en cada punto de contacto del cliente con el distribuidor y/o la armadora
- ☞ El rediseño de los procesos de venta, atención y servicio, y la identificación del apoyo tecnológico requerido
- ☞ El alineamiento de los esquemas de medición e incentivos, y el cambio cultural del personal

Empresas armadoras pretenden producir y vender en México

En el último año se han anunciado importantes inversiones de las empresas armadoras automotrices en México que en muchos casos parecen contemplar la idea de producir para abastecer el mercado local. Al menos General Motors ha anunciado que la planta que se está construyendo en el estado de San Luis Potosí se destinará a la producción de autos compactos, primordialmente para su venta en el mercado mexicano, lo que augura un buen futuro para este país en términos de creación de empleos y en la mejora de la balanza comercial. Considero que este es un momento muy importante para la industria automotriz nacional que podríamos utilizar incluso para incrementar nuestra participación en otros mercados latinoamericanos.

Oportunidades de México en el mercado automotriz actual

Las grandes empresas automotrices están buscando crecer en mercados emergentes, incrementar la gama de productos y segmentos, así como adoptar nuevas tecnologías. La entrada de países emergentes al escenario automotriz es parte de los retos que tendrán que enfrentar, cada vez más, los productores mexicanos. La oportunidad de México radica en eficientar y automatizar más los procesos productivos y asumir una constante innovación tecnológica, carrera que se desarrolla a grandes velocidades en los países que han decidido permanecer en el escenario automotriz.

2.1.3.5 Perspectiva para México

Panorama Macroeconómico para 2007

A pesar de que la economía de Estados Unidos se desacelera fuertemente y se aprecia un mayor riesgo por una desaceleración más profunda y no por inflación, los analistas auguran un “aterrizaje suave” para 2007. El precio del petróleo tuvo un arranque sorpresivo de año con una baja que ya el 12 de enero quedó por debajo del precio promedio fijado por la SHCP⁵ en el presupuesto de ingresos para 2007. Sin embargo, el Secretario de Hacienda, Agustín Carstens, declaró que de no ser por una baja fuerte y constante por un periodo largo de tiempo, no es necesario hacer ajustes al presupuesto, por lo que el precio promedio para este año se mantiene en 42.8 dólares por barril. Del mismo modo, con todo y un alza en el precio de productos e insumos como la tortilla, el maíz y la azúcar; la SHCP pronostica una inflación menor que la registrada al cierre de 2006, 4.01%, y la ubican en 3.0%. Aunque para el común de los mexicanos las señales no son muy buenas para este primer año del sexenio Calderonista, analistas y el Gobernador del Banco de México pronostican un año positivo.

La industria automotriz mexicana y su perspectiva para el 2007

El 2006 ha sido un excelente año para la industria automotriz mexicana; finalmente estamos en el umbral de superar el record de producción que alcanzamos en el 2000, pero ahora con una perspectiva a futuro mucho más prometedora que en ese año. A pesar de la debilidad que pueda mostrar el mercado interno (solamente crece aproximadamente 1% en forma anual) existen diversos factores que muestran un futuro mucho más alentador. Tan sólo hay que echar un vistazo al anuncio de inversiones que se ha hecho en el transcurso del año, lo que indica que la apuesta de las armadoras incluye sin duda a nuestro país por encima de otras grandes economías. El 2007 se muestra muy interesante con retos importantes en el mercado nacional e internacional. Las automotrices chinas están penetrando en los diversos mercados incluyendo el mexicano, el cual en los próximos meses verá la introducción de algunas marcas chinas que contribuirán a incrementar más la competencia. Varias empresas armadoras han anunciado importantes inversiones tendientes a mejorar la posición exportadora del país y a reducir la importación de autos, ya que ahora se espera producir también para el consumo local. Esto dará un impulso interesante al papel de México en el ámbito automotriz mundial.

La industria automotriz mexicana hacia el 2013

La industria automotriz es una de las más maduras y sólidas en México, sus planes de inversión y desarrollo son un importante indicativo. Aun y cuando analistas internacionales han prendido focos amarillos de alerta por el ambiente preelectoral dominante, tenemos a la vista importantes inversiones por parte de las armadoras. La recuperación de niveles de producción y venta de unidades (principalmente en el mercado de exportación) son ya una realidad. El año pasado México produjo un millón 586 mil 110 unidades, y la perspectiva para este año es que la producción se incrementará cerca de 19 por ciento, para llegar a 1.88 unidades producidas aproximadamente. Lo anterior representaría un seguimiento de la tendencia positiva observada en este rubro en los últimos meses, derivado, principalmente, de las inversiones para modernizar las líneas de producción e incrementar con ello la producción de nuevos modelos, realizadas por las principales empresas ensambladoras establecidas en el País. Para el año 2013, pronosticamos que México puede alcanzar las 2.3 millones de unidades producidas, es decir, unas 500 mil unidades más en un periodo de 7 años. Asimismo, en ese periodo se espera la construcción de 11 nuevas plantas ensambladoras en Norteamérica, de las cuales México podría quedarse con cuatro de ellas.

⁵ Secretaría de Hacienda y Crédito Público

2.2 SISTEMAS DE GESTIÓN AUTOMOTRIZ

2.2.1 LA AGENCIA AUTOMOTRIZ

2.2.1.1 Retos y tendencias de los distribuidores en México

México está en proceso de evolución de la mejora y de la calidad, tanto por parte de los clientes como de los distribuidores, aún falta algo de tiempo para que el grado de maduración sea aceptable. ¿Quién lleva la delantera? Hay algunos distribuidores de marcas enfocadas a la satisfacción del cliente que ya se distinguen por aplicar con éxito prácticas internacionales, otros están aún analizando cómo mejorar en ese aspecto y otros todavía creen que el cliente tiene que entender que ésa es la forma de operar y ni modo. En cuanto a la clientela, cada vez conoce más. Las diferentes marcas en el mercado dan esa oportunidad, con automóviles de alta tecnología y servicios en los que unos y otros deben estar acordes. La gama de modelos disponibles en el mercado mexicano excede los 300. Como consecuencia, los retos se sintetizan en que:

- ☞ El cliente debe salir verdaderamente contento de visitar a un distribuidor, lo que implica una cultura interna de servicio esmerado y bajo el concepto de mejora continua.
- ☞ Capacitación a todo el personal, tanto al que tiene contacto directo con el cliente como al que no lo tiene.
- ☞ Adecuar y modernizar las instalaciones operativas y las visibles para los clientes.
- ☞ Los servicios post-venta⁶ deben cumplir con lo que espera el cliente a precio razonable, lo cual requiere procesos efectivos de supervisión y controles operativos y contables acordes.
- ☞ La optimización de costos, asesorada por un externo con experiencia reconocida, lo cual evitará no caer en reducciones de costos que traen como consecuencia limitaciones en la forma de operar. Optimización y reducción de costos son dos caminos distintos y así son los efectos, aquellos traen aumento de productividad, los segundos pérdida de la misma.
- ☞ Rapidez en el servicio, hay que evaluar para mejorar constantemente los controles operativos manuales y computarizados, los cuales, o facilitan o dificultan estos procesos.
- ☞ Los automóviles “chocolate y usados extranjeros”. Los vehículos nacionales deben desplazarlos, pero los precios de compra, las facilidades para adquirirlos y el servicio post-venta deben ser más competitivos que los beneficios que se obtienen de esos vehículos. Los consumidores apenas están aprendiendo a identificar que hay menor rendimiento en el consumo de combustible, en la seguridad, en las medidas no contaminantes, etc. Lo que sí ya tienen muy bien identificado, es la diferencia en precios, hay que cerrar esa brecha nos guste o no.
- ☞ Talleres calificados de franquicias tienen una excelente oportunidad de negocio en México, porque su esquema de operación es más flexible y menos costoso, estos incrementan la competencia y competitividad en el mercado, por lo que los distribuidores de marcas deben estar en mejora continua.
- ☞ Hay distribuidores de una misma marca que se depredan entre sí. Deben hacerse revisiones externas independientes y objetivas sobre el cumplimiento de las políticas, así como *Benchmarks*⁷ e incluir visitas de *Mystery Shopper*⁸.

⁶ Mantenimiento del Automóvil.

⁷ Comparativa, técnica utilizada para medir el rendimiento de un sistema o componente de un sistema.

⁸ Comprador Encubierto.

2.2.2 SISTEMAS DE GESTIÓN AUTOMOTRIZ

2.2.2.1 Software para Agencias Automotrices en México.

A la hora en que se escribe este reporte, existen un total de 1450 agencias distribuidas en las principales ciudades del territorio nacional. Por ende, 1450 necesidades de un sistema, la cual se han ido cubriendo de varias maneras:

- ☞ Sistemas propios desarrollados en las agencia
- ☞ Comprando subsistemas individuales para subprocesos
- ☞ Software Extranjero
- ☞ Software Nacional de empresas dedicadas.

Comentando las tres primeras opciones, tenemos que existen muchas desventajas, dado que si el sistema se desarrolla en la propia agencia se corre el riesgo de ser muy poco robusto y cubra solo una mínima parte de las necesidades de la agencia; y si se piensa en hacer un programa que cumpla al 100% con las necesidades se tendría que pensar en tener un staff de programación con una plantilla de varios programadores, lo cual resultaría un gasto excesivo.

Si pensamos en la alternativa de tener subsistemas para los procesos, tenemos el riesgo de hacer procedimientos por partida doble, lo cual impactaría de manera dramática en el personal y el tiempo invertido, ejemplo, una orden de servicio que es capturada e impresa al cliente deberá ser recapturada en el área de contabilidad como una póliza contable, y a su vez vuelta a capturar para conformar el saldo de cuentas por cobrar.

Podemos afirmar que prácticamente todos los inversionistas extranjeros que han tratado de hacer negocios en las agencias mexicanas han fracasado. La razón es que el software viene de países desarrollados como Estados Unidos, Canadá o España, donde existen condiciones de trabajo muy diferentes. Y si pensamos importar software de países pobres que tengan la misma situación de pobreza, pues encontraremos un mercado nulo. Los programadores Mexicanos conocen el mercado y están acostumbrados a enfrentar situaciones extremadamente complejas, lo que ha minimizado la inversión extranjera dramáticamente.

Por las razones comentadas anteriormente, la alternativa más viable para una agencia de autos es comprar un sistema integral desarrollado por una empresa mexicana. Lo cual reduce sus costos de operación, aumenta las ganancias, evita la subutilización de sistema y brinda además una gama de consultaría, además de estar a la vanguardia en las nuevas regulaciones.

2.2.2.2 Oferta de Software Automotriz.

Dentro de este contexto existe un mercado para estas nuevas empresas mexicanas de 1450 sistemas, que se venden de entre 1 y 3 millones de pesos, además de una cuota mensual de mantenimiento, por lo cual estamos hablando de una industria mayor a los 2000 millones de pesos, lo cual resulta un mercado bastante tentador. Aunque existe competencia, el mercado esta abierto a cualquier posibilidad. Actualmente existen a nivel nacional 15 empresas dedicadas al software automotriz, de las cuales sobresalen tres, las tres del Distrito Federal las cuales absorben el 50% del mercado nacional: Intelisis, Bussines Pro y Sapsa⁹.

⁹ Sistemas Automotrices Proactive, S.A. de C.V.

2.2.2.3 Sistemas Automotrices Proactive, S.A. de C.V.

La empresa en la cual me desempeño actualmente se llama Sistemas Automotrices Proactive (SAPSA), la cual en este 2007 cumple 25 años. SAPSA surge en 1982 en el departamento de sistemas de CAMSA una de las empresas en la venta de autos en el distrito federal. El dueño de Camasa, cansado de tener sistemas que no cubrían las necesidades de la agencia, se da a atarea de contratar a sus plantilla propia de programadores para hacer sus propia sistema, el cual iba a ser usado en la agencia de plaza universidad y en las otras agencias del grupo Camasa. El resultado fue un software bastante eficiente, y del que se enteraron los demás socios de la asociación de distribuidores, lo que adquirieron el sistema y así sucesivamente por otros clientes no solo en el Distrito Federal, si no el toda la republica. Actualmente Sapsa desarrolla el Proactive 7.0 un software integral que utilizan 120 agencias en México, es decir un 10% del mercado. Su presencia a nivel nacional le ha visualizar planes de internacionalización el mercado de centro y Sudamérica. SAPSA se rige bajo los siguientes valores:

- **Visión.** Generar nuevas tendencias en la tecnología de software automotriz, creando oportunidades de negocio para ser reconocidos como una empresa seria y prestigiosa en el mercado mexicano que ofrece la absoluta seriedad de proporcionar a sus clientes LA MEJOR SOLUCIÓN INFORMATICA.
- **Cultura Organizacional.** Como organización es muy importante tener una cultura empresarial que nos permita contar con identidad y directriz en un mercado altamente competitivo como lo es el Automotriz.
- **Misión.** Ser una empresa líder de excelencia en el desarrollo e integración de soluciones informáticas, para lograr la más alta rentabilidad y competitividad en el mercado automotriz, conjuntado el conocimiento y la eficiencia de nuestros colaboradores para ofrecer a nuestros clientes un servicio que satisfaga completamente sus necesidades.
- **Filosofía.** Creemos en la empresa por su alta rentabilidad en el mercado, gracias a la excelencia en la integración de sus soluciones informáticas; en n nuestra fuerza de trabajo, por ser gente eficiente y comprometida, como un factor vital de crecimiento en un ámbito de competitividad; en la ética profesional con la que cada uno de nosotros desempeña su trabajo; y en la calidad de servicio y valor agregado que Sapsa ofrece para la satisfacción completa de nuestros clientes.
- **Productos y Servicio.** SAPSA es una compañía dedicada al desarrollo de software automotriz, nuestros clientes han incrementado su competitividad al integrar a Proactive a sus áreas de finanzas, servicios, ventas y comerciό electrónico. Otros de los productos y servicio que manejamos son: Consultaría, Desarrollos en Internet, Capacitación, Desarrollos Cliente- Servidor, Asesoría Técnica, Asesoría en Negocios e Implantación de Sistemas.
- **Nuestra Experiencia.** Dentro de nuestros principales clientes están agencias de diversas marcas, lo que asegura la correcta operación y fácil adaptación de nuestro sistema a sus necesidades, esto nos permite colaborar con diversas marcas y metodologías para ofrecer la garantía de satisfacer a nuestros clientes más exigentes: Volvo, Chrysler, Volkswagen, Lincoln, Honda, Ford, Toyota, Chevrolet, Nissan, Land Rover, Peugeot, Jaguar.
- **Certificaciones.** Fuimos certificados por Ford e IBM¹⁰, obteniendo el primer lugar de los distribuidores Ford. SAPSA – Proactive obtuvo el 91.27% de 320 linimientos especificados.

2.2.2.4 Crisis de los Sistemas de Gestión

A pesar de que se ha avanzado en la informatización de las agencias, todavía existe un camino muy largo por recorrer, de tal manera que los sistemas totales se hagan totalmente eficientes. A pesar de los esfuerzos de los ingenieros mexicanos, se tendrán que redoblar esfuerzos para abatir los siguientes problemas que aun persisten.

¹⁰ International Bussines Machines

2.2.3 FUTURO DE LOS SISTEMAS Y NUEVOS RETOS POR ENFRENTAR

2.2.3.1 Retos actuales de la agencia automotriz después de TLC

Automotriz: una industria de cambios y retos. Sin duda alguna, la industria automotriz, en cada uno de los eslabones que la integran, es el más claro ejemplo de los procesos de cambio acelerado que caracteriza a las economías modernas, cambios que tienen su más clara expresión en la comercialización y cuyos efectos trascienden el ámbito de la propia cadena productiva. De hecho la industria automotriz no ha dejado de cambiar y generar cambios en otros sectores, prácticamente desde su nacimiento. La relevancia del sector automotor tanto en el ámbito económico como en el social del mundo entero lo ha colocado como objeto de análisis permanente de diversos especialistas.

Ejemplo de ello es la publicación del artículo *Perpetual Motion* en la revista *The Economist* de septiembre de 2004, en el cual se presentan los aspectos más importantes de cambio en la industria automotriz. El mencionado artículo resulta por demás interesante y útil para comprender la esencia de las transformaciones actuales de nuestro sector y la esencia de la intensa competencia entre marcas por ganar mercados. Es por ello que a continuación se presenta un fragmento del documento (traducido), con la intención de aportar información objetiva que nos permita visualizar hacia dónde va el sector, en beneficio del análisis y planeación “desde nuestra trinchera”. La industria del automóvil ha estado adelante de su tiempo en muchos aspectos. Peter Drucker, quien se hizo de nombre con un estudio sobre General Motors en 1945, acuñó la frase *industria de industrias*. La compañía también era líder en “obsolescencia planeada”, por los frecuentes **cambios** en diseño que tentaban a los consumidores a cambiar de modelo cada año. Fue la primera en sentir el enojo de los consumidores, con la publicación en los 1960’s del ataque de Ralph Nader al historial de seguridad de los Tres Grandes manufactureros de Detroit (The Big Three), “Inseguro a Cualquier Velocidad”. En los 1970’s, cuando se cuadruplicó el precio del petróleo, la industria se encontró bajo el ataque de ecologistas furiosos por los hábitos devoradores de gasolina de sus productos.

También estuvo entre las primeras en quedar bajo cuidadoso escrutinio gubernamental, desde lo concerniente a **seguridad** hasta asuntos **ambientales** y preocupaciones antimonopolio en los días en que General Motors tenía 60% del mercado de EE.UU. y podía deshacerse de competidores con unas pocas bajas de precio bien seleccionadas. Pero también recibió bienvenidas y atenciones gubernamentales. Cuando pequeños, económicos y confiables vehículos japoneses empezaron a comerse la porción de mercado de Detroit, el gobierno americano impuso restricciones en la importación. Muy pronto, la industria en Europa se vio bajo presiones similares. La industria automotriz también se vio en los límites del capitalismo en otro sentido. A medida que las técnicas de producción masiva se desarrollaron en los 1920’s y 1930’s, sus trabajadores se fueron sindicalizando. A veces parecía que las fábricas en el área de Detroit, las ‘Midland’ Británicas o las enormes plantas cercanas a París eran los principales campos de batalla de la guerra de clases. Aún ahora, el sindicato de Trabajadores del Auto Unidos domina Detroit, incluso cuando la pertenencia a sindicatos en el sector privado americano se encuentra por debajo del 10% de la fuerza de trabajo.

Hasta el año 2003 el liderazgo del sindicato parecía tener su cabeza en la arena, olvidándose de la competencia que estaba dañando a General Motors, Ford, y la parte Chrysler de DaimlerChrysler. Súbitamente entendió el mensaje, acordó un aumento moderado de pago y aceptó despidos. El sindicato ha visto su membresía declinar fuertemente en los pasados 20 años, pero aún puede dificultarles a las empresas automotrices la reducción de su

sobrecapacidad. Hoy la industria automotriz es el epítome de la producción en masa, la mercadotecnia en masa y el consumo en masa, con algunas de las marcas más fuertes del mundo. Para la mayoría de los hogares en los países ricos del mundo, es la segunda compra más grande después de una casa o apartamento. Pocas industrias de bienes al consumidor dependen tanto de un mercado secundario saludable para sus productos. Sin embargo, hay fuerzas poderosas que podrían **cambiar** profundamente la industria. Una es la *fragmentación del mercado*, que podría llevar a menores corridas de producción. Otra es la *insatisfacción con el costoso sistema de construir vehículos para inventario*, no a la orden. Una tercera es la innovadora *construcción modular*, en la que una mayor parte del vehículo se construye por los proveedores de partes. Y más adelante, una cuarta fuerza podría ser un *cambio a vehículos eléctricos* con controles electrónicos y eléctricos y ya no mecánicos.

Un mundo cayéndose en pedazos. Ahora mismo, sin embargo, la fuerza más grande para el cambio es el hecho de que la mayor parte de la industria de volumen del vehículo está descompuesta y necesita componerse. El mercado en Estados Unidos, Europa y Japón, donde se vende más del 80% de autos y camiones se está quedando sin crecimiento. En Estados Unidos la llegada de fabricantes europeos, japoneses y sudcoreanos ha creado sobrecapacidad. Aún más, a medida que los fabricantes de Estados Unidos mejoran constantemente su productividad para ponerse a la altura de estos nuevos rivales, su propia eficiencia incrementa la capacidad en 3% al año.

En Alemania y Francia, leyes laborales rígidas han inhibido el cierre de viejas fábricas redundantes, aunque Renault ha puesto un buen ejemplo, y Ford Europa y General Motors Europa han tratado de seguirla. En Japón, las cerradas sociedades industriales conocidas como keiretsu¹¹ han probado ser demasiado rígidas para algunos manufactureros. Sólo Toyota y Honda siguen en manos puramente japonesas. Los productores japoneses más pequeños logran poca o ninguna ganancia en casa y luchan por lograr números negros en Europa. Incluso para las compañías grandes Estados Unidos provee las mejores esperanzas de ganancias crecientes. El resto del mundo presenta un cuadro mixto. En Asia la crisis financiera de 1997 le dio a la industria sudcoreana automotriz –siempre una creación más bien artificial– un gran golpe. Hoy solamente Hyundai sobrevive. En Sudamérica el colapso económico en Brasil y Argentina detuvo la rápida expansión de la industria automotriz allí, dejando a los inversionistas extranjeros como Fiat absorber sus pérdidas. El ‘boom’ en China tiene a todos entusiasmados, pero debe ser mantenido en perspectiva. Por todo su inmenso crecimiento porcentual, el valor anual de ese mercado es equivalente a más o menos un mes de ventas en el resto del mundo. Todas las compañías de autos piensan que si sólo lo intentan más, de algún modo crecerán a costa de sus rivales. Pero en realidad son, como decía Scott Fitzgerald, “botes contra la corriente, echados atrás incesantemente hacia el pasado”. Añadan los crecientes gastos de pensión y salud de los productores tradicionales como los tres grandes de Estados Unidos y los europeos, y es fácil ver porqué la industria se siente bajo sitio.

2.2.3.2 Competencia más Férrea entre las Marcas

- Mercado automotriz: intensa competencia.
- Cada vez más modelos.
- Regulaciones más complejas.
- Legalización de vehículos.

¹¹ Es un término japonés que hace referencia a un modelo empresarial en el que existe una coalición de empresas unidas por ciertos intereses económicos

3. DESARROLLO DE UN CASO PRÁCTICO

A través del presente capítulo recorreremos el camino más corto en la travesía de crear sistemas para la gestión de agencias automotrices.

Empezaremos en el **análisis** por responder a la pregunta básica a partir de la cual comienza todo el ciclo de desarrollo de sistemas: ¿Qué hay que hacer?; Después toda vez que sabemos exactamente lo que necesitamos, vamos a continuar en el **diseño** respondiendo a la pregunta ¿Cómo le vamos a hacer? Y partir de esta premisa iniciaremos la construcción de nuestro producto.

Posteriormente, nos daremos a la tarea organizar el equipo de trabajo para la **construcción** del sistema, basándonos en estándares y reglas bien definidas para que todo sea lo más rápido posible y asegurar la calidad. Una vez terminado nuestro producto, en la **implementación** comenzaremos a usarlo para el objetivo por el cual fue construido. Sin embargo aquí no concluye el proceso, ya que como hemos comentado anteriormente, los sistemas actuales son cada vez más dinámicos y adaptables a los cambios en procedimientos y políticas de operación, por lo cual en las etapas de **atención a clientes, mantenimiento y control de calidad** conoceremos los procedimientos actuales que se llevan a cabo para *evolucionar* nuestro sistema.

Con esto pretendemos ser una guía para aquel que se aventure a desarrollar nuevos sistemas (no necesariamente automotrices), y que el camino que nosotros hemos recorrido les sirva a los otros para tomar atajos en el sinuoso camino del desarrollo de sistemas.



Imagen 1. El proceso para crear sistema involucra diversos elementos que deben trabajar armónicamente para llegar a un resultado óptimo.

3.1 ANÁLISIS

El objetivo del análisis del sistema es comprender situación. Al hacer el análisis sobre una agencia automotriz se hace hincapié en la investigación y el cuestionamiento para conocer como opera la agencia e identificar los requerimientos que tiene los usuarios. Solo después de comprender en su totalidad la agencia, los analistas están en posición de analizarlo y generar recomendaciones para el diseño del sistema.

3.1.1 CONCEPTO DE ANÁLISIS.

Durante la práctica en nuestra empresa, a esta etapa le damos un enfoque muy particular. El equipo de análisis debe tener muy claro que de los que se trata el análisis es de dos cosas: **Descomposición** y **Síntesis**.

Analizar en primera instancia significa “Descomponer un mecanismo complejo, de tal manera que hagamos de una TODO complicado, subsistemas pequeños de menos complejidad que faciliten su estudio”. En este caso podríamos ejemplificar esta situación haciendo la analogía del sistema en estudio con un auto. Si tuviéramos la consigna de analizar el funcionamiento de un automóvil, tendríamos primero que partirlo en sus subsistemas básicos, tal es el caso del sistema eléctrico, el motor, la transmisión, el sistema de frenos, etc.

De la experiencia adquirida en la evaluación de otros sistemas en las agencias automotrices, trazamos planes de trabajo previos al análisis, aplicando ya esta subdivisión de tal manera que se agilice el análisis partiendo a la agencia en dos bloques funcionales: Área administrativa y Área Operativa. La parte administrativa está compuesta por los departamentos de Contabilidad, Cobranza de Créditos y Pago a Proveedores. La parte operativa esta compuesta por los módulos Venta de Autos, Venta de Refacciones y Operaciones de Servicio.

Ahora bien estamos hablando de sistemas integrales, es decir, que van a controlar todas las operaciones de la agencia de autos, y el cual será usado por cientos de usuarios. Por tal motivo será recomendable llevar la descomposición a un tercer nivel para que nos dé desde el principio una visión detallada. Cada uno de los módulos a su vez puede ser dividido en las siguientes partes:

Autos: Nuevos y Seminuevos; Refacciones: Facturación, Inventarios, Clasificación de Inventario; Servicio: Recepción de Unidades, Pagos, etc, etc.

A esta primera etapa de **descomposición**, teóricamente se le conoce como **Determinación de Requerimientos**.



Imagen 2. Menú de la Interfaz del Sistema Proactive

La parte de la **síntesis** trata de volver a unir lo que se descompuso, estableciendo las relaciones entre los diversos módulos a manera de esquemas que ejemplifican la manera que fluye la información. La síntesis es como armar el rompecabezas nuevamente.

Podemos ejemplificar esta situación mediante el proceso de facturación de órdenes de servicio. Una factura de servicio contiene la información del cliente así como de la unidad a la que se le practicó el servicio, a su vez contiene los ítems que se le agregaron es decir la operaciones de mecánicas que se le llevaron a cabo a la unidades, las refacciones que se le metieron al auto, lo que nos da un total. Pero a su vez y de manera simultanea la facturación debe crear también una póliza contable de servicio, en la cual se especifican los costos de las refacciones, los costos de la mano de obra, la afectación contable a las ventas, etc. También simultáneamente, si la orden de servicio se hizo bajo la condición de crédito, se debe crear una cuenta por cobrar para el departamento de cobranza, desde donde se calculará el saldo del cliente en base a sus facturas meno u pagos.

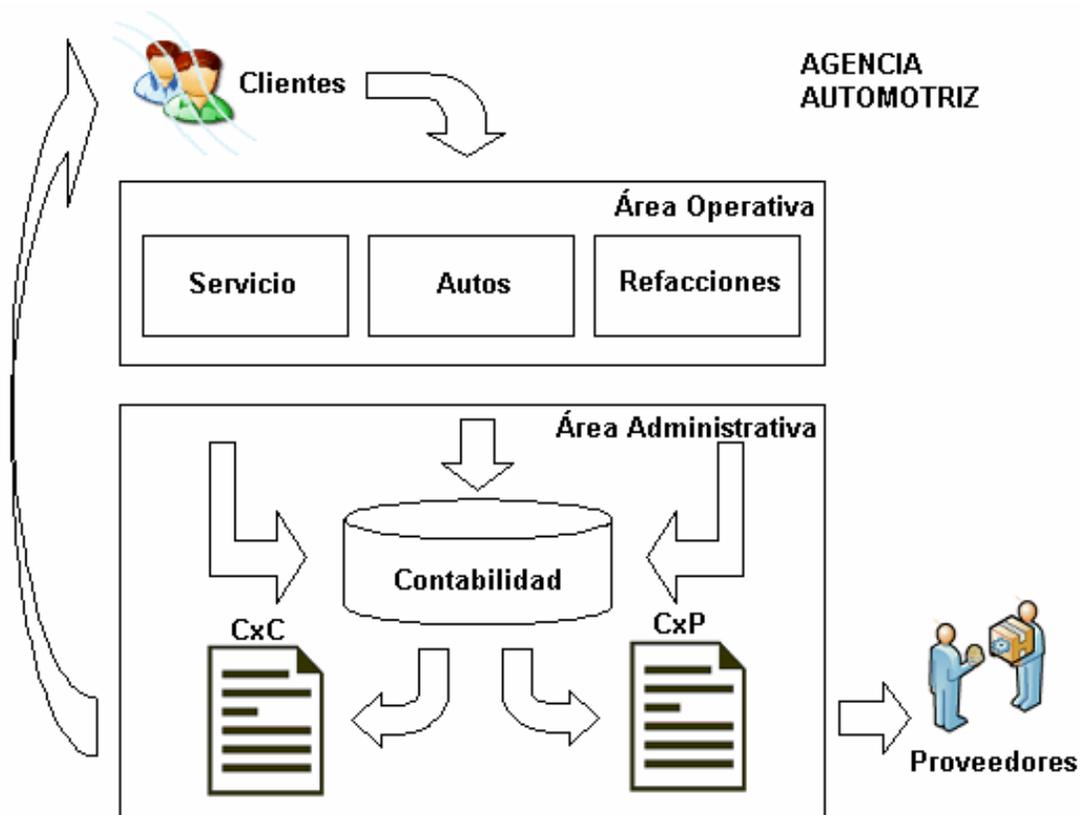


Imagen 3. Diagrama de la forma en que opera una agencia de autos.

A esta segunda etapa de **síntesis**, teóricamente se le conoce como **Análisis Estructurado**.

Ahora, para llevara a cabo este objetivo de comprensión de las actividades de la agencia de autos, se integra un equipo de dos analistas experimentados, con el cual será suficiente para hacer el análisis.

3.1.2 RECOMENDACIÓN PREVIAS AL ANÁLISIS.

Pero estamos en México. Aquí las personas caminan por en medio de la calle, y los autos se estacionan en la banquetta. Si se trata de sistemas, la cosa se complica aun más. Es duro aceptarlo, pero en este punto la teoría sobre desarrollo de sistemas nos sirve de muy poco, y que como es sabido, estas teorías se hicieron en países industrializados, es decir, bajo condiciones ideales de trabajo... bajo otra cultura e idiosincrasia. Aquí en México nos sirve de muy poco esta teoría IDEAL.

Sin embargo, en nuestra empresa a través de la experiencia hemos incluidos diversas técnicas para atacar los problemas que se nos presentan en el desarrollo de sistemas aquí en México. Hemos aprendido que durante el análisis además de tratar de comprender el funcionamiento del sistema, es una fase de conocimiento de todos aquellos factores que puedan ALLANAR el buen camino de nuestro sistema, es decir, escudriñar y descubrir todo aquello que pueda representar una amenaza al óptimo desempeño de nuestro sistema. Nos enfrentamos al arte de controlar al factor humano que con el sistema va a interactuar, si es que verdaderamente queremos que se implante con éxito nuestro programa, ya que como lo ha determinado la experiencia, el usuario en vez de promover el uso se comporta como un verdadero SABOTEADOR.

Dentro de las actividades enfocadas a impedir que nuestro sistema sea saboteado, están las siguientes:

1.- Establecer formalmente las causas del sistema.

Una vez que el empleado de la agencia se entera que se va a poner en marcha la creación de un nuevo sistema de cómputo para llevar la operación de la agencia, el miedo le causa pánico y la incertidumbre se apodera de él por las siguientes causas:

- Que el sistema se sistematice tanto que el empleado salga sobrando.
- Miedo a que se descubran corruptelas y prácticas sucias para ganar dinero extra.
- Ya no ser indispensable y el único que sabe como se llevan a cabo “procesos especiales” por tanto inamovible.
- Se descubra que los procesos no se están llevando a cabo en tiempo y forma.
- Miedo a que se descubran errores maquillados.

Por esta y otras razones los malos usuarios (que por desgracia son la mayoría) tendrán una predisposición contra el sistema, ya que como pudo verse, potencialmente podría convertirse en la llave para abrir la caja de Pandora. Ahora, esta predisposición se manifiesta de la siguiente manera:

- Críticas infundadas hacia el sistema.
- Descalificación argumentando que hay cosas que no pueden hacerse.
- Ocultando bases de datos.
- Ocultamiento de procesos.
- Definiciones de procesos excesivamente complejas que no corresponden a la realidad.
- Exigencia de funcionalidades poco usuales.

Así pues, antes de que sucedan todas estas anomalías. Simultáneamente al proceso de análisis se debe llevar a cabo un proceso de autopromoción del sistema. Esto con la finalidad de ir “vendiendo” el sistema entre los usuarios a fin de ganar aceptación y por otra parte de dar razones de peso para hacer ver que la implantación es ya un proceso inminente y que ya no puede dar marcha atrás:

1. Levantar un acta donde se determine que el sistema surge de una observación de un despacho auditor de una instancia superior.
2. Mostrar en una relación matemática las pérdidas económicas que se están teniendo por no contar con el sistema.
3. Se hace énfasis en el incremento de la productividad y no tanto las ganancias.
4. Amenazar que podemos perder todo en caso de no actualizarnos.

2.- Identificar los usuarios que “saldrán perdiendo” con la puesta en marcha del sistema.

Ya sea un facturista que ya no se necesitará, una cajera, un vendedor que ya no podrá cobrar “comisión” por “debajo del agua”, debemos aceptar que el cambio podrá afectar algunos intereses, de tal manera que el cambio pueda hacer que se despida personal. Será necesario pasar un informe con todas estas personas que pierden a fin de que pueda reubicárseles y que no pierdan para evitar crear enemigos del sistema.

3.- Análisis costo-beneficio.

No todo lo que los usuarios piden debe hacerse, porque a veces son caprichos infundados. Al definir si los usuarios piden “procesos automáticos” injustificados, se deberá proceder viendo la utilidad del mismo.

4.- Personal con limitaciones técnicas.

Es un hecho que ya sea por los bajos sueldos que hay en el país, la rotación del personal, la inclusión al equipo de trabajo de gente recomendada de “confianza”, siempre habrá usuarios que no reúnan los estándares de capacitación técnica para desempeñar determinado puesto. Hay personas que ni siquiera saben manejar una computadora y por lo tanto se tendrá que notificar que no son aptos para el puesto, y ya sea que los cambien o los capaciten.

5.- Infraestructura de cómputo en mal estado.

No se quiere invertir en actualizar equipos de cómputo, redes, telecomunicaciones, servidores, y los cuales al fallar hacen que el que paguen los platos rotos sea el sistema.

3.1.3 DETERMINACIÓN DE REQUERIMIENTOS.

La determinación de requerimientos es el estudio de un sistema para conocer cómo trabaja y dónde es necesario efectuar mejoras. Los estudios de sistemas dan como resultado una evaluación de la forma como trabajan los métodos empleados o si es necesario o posible realizar ajustes. Estos estudios consideran métodos tanto basados en computadoras como manuales; es decir no se circunscriben exclusivamente a estudios de cómputo.

Un requerimiento es una característica que debe incluirse en un nuevo sistema. Esta puede ser la inclusión de determinada forma para capturar o procesar datos, producir información, controlar una actividad de la empresa o brindar soporte a la gerencia. Es así como la determinación de requerimientos vincula el estudio de un sistema existente con la recopilación de detalles relacionados a él.

Dado que los analistas de sistemas no trabajan como gerentes o empleados en los departamentos de usuarios (como mercadotecnia, compras, producción o contabilidad), no tiene los mismos conocimientos, hechos y detalles que los usuarios y gerentes de esas áreas.

Por consiguiente el primer paso del analista es comprender la situación. Ciertos tipos de requerimientos son tan fundamentales que son comunes en casi todas las situaciones. Dar

respuesta a un grupo específico de preguntas le será de gran ayuda para entender los requerimientos básicos. También existe otra clase de requerimientos de si el sistema está orientado hacia transacciones, toma de decisiones o si existe por varios departamentos.

Actividades de la Determinación de Requerimientos:

a) Anticipación de Requerimientos. La experiencia de los analistas en un área en particular y el contacto con sistemas en un ambiente similar al que se encuentra bajo investigación, tiene influencia sobre el estudio que éstos realizan. Su experiencia les permite anticipar ciertos problemas o características que investigan del sistema actual, las preguntas que formulan o los métodos que utilizan están basados sobre esta familiaridad.

La anticipación de requerimientos puede ser una mezcla de bendiciones. Por un lado, la experiencia de estudios previos puede conducir a la investigación de áreas que no investigaría un analista novato. Tener las bases necesarias para saber qué preguntar o qué aspectos preguntar puede ser de beneficio sustancial para la organización.

Por otra parte, si se introducen sesgos o tajos al conducir la investigación entonces es muy probable que la anticipación de requerimientos se convierta en un problema. Por lo tanto siempre deben darse lineamientos para estructurar un investigación alrededor de cuestiones básicas con la finalidad de evitar consecuencias indeseables de la anticipación de requerimientos.

b) Investigación de Requerimientos. Esta actividad es la más importante del análisis del sistema. Los analistas estudian el sistema actual con la ayuda de varias herramientas y habilidades, y documentan sus características para, mas adelante, emprender el análisis.

La investigación de requerimientos depende de las técnicas para encontrar datos, e incluyen métodos para documentar y describir las características del sistema.

c) Especificación de Requerimientos. Los datos obtenidos durante la recopilación de hechos se analizan para determinar las especificaciones de los requerimientos, es decir la descripción de las características del nuevo sistema. Esta actividad tiene tres partes relacionadas entre si:

1. Análisis de datos basados en hechos reales. Se examinan los datos recopilados durante el estudio, incluidos en la documentación de flujo de datos y análisis de decisiones, para examinar el grado de desempeño del sistema y si cumple con las demandas de la organización.
2. Identificación de Requerimientos esenciales. Características que deben incluirse en el nuevo sistema y que van desde detalles de operación hasta criterios de desempeño.
3. Selección de Estrategias para satisfacer los requerimientos. Métodos que serán utilizados para alcanzar los requerimientos establecidos y seleccionados. Estos forman la base para el diseño de sistemas. Los cuales deben cumplir con la especificación de requerimientos.

Las tres actividades son importantes y deben de realizarse de forma correcta. La especificación de requerimientos implica una gran responsabilidad para los analistas de sistemas, ya que la calidad del trabajo realizado en esta etapa se verá reflejada más adelante en las características del nuevo sistema.

EJEMPLO DE DETERMINACION DE REQUERIMIENTOS.

3.1.3.1 a) Anticipación de Requerimientos

Nuestra empresa se ha dedicado a desarrollar sistemas automotrices por mas de 20 años, por lo cual se puede afirmar que no vamos a partir de cero en este analices, ya que conocemos muy bien el funcionamiento de las agencias automotrices como para ver a este etapa como una REINGENIERÍA DE PROCESOS, donde se buscará hacer cada vez más eficientes y adaptados a los tiempos actuales todos lo procedimientos de la agencia, utilizando nuevas técnicas y herramientas operativas. Por tanto será fundamental visualizar previamente el conocimiento previo para tener una clara idea de lo que se va a hacer, para así tener una buena planeación del proyecto, lo que impactará en el éxito del mismo.

-Requerimientos Básicos

¿Cuál es la finalidad de un sistema para la administración integral de agencias automotrices?

El objetivo principal para un sistema de administración integral de agencias automotrices es el de AUTOMATIZAR el proceso principal de la agencia, es decir, el proceso de compra-venta. Como cualquier otro negocio, una agencia automotriz es un intermediario en la planta ensambladora de autos y el cliente final, por lo cual la principal prioridad del sistema en la agencia será la de AUTOMATIZAR el proceso de compra-venta. Por AUTOMATIZAR el proceso de compra-venta lo vamos a conceptuar como el uso de la computadora para SISTEMATIZAR los procesos, AGILIZAR los cálculos relacionados y REGISTRAR la información de bases de datos. Estos tres pasos permitirán básicamente ahorrar tiempo lo cual repercutirá directamente en el aumento de capacidad operativa, lo cual se traduce en mayores ventas con menos recursos humanos, y así optimizando se tendrá ventaja sobre otros competidores en el mercado automotriz.

¿Qué pasos se siguen para el proceso de compra venta?

En una agencia automotriz se siguen básicamente tres procesos de compra-venta:

Autos	Refacciones	Servicio
Recepción de Unidades	Clasificación de Inventario	Elaboración de Presupuesto al cliente
Pedido del Cliente	Generación del Pedido sugerido	Autorización del Presupuesto y Creación de la orden de servicio
Asignación de Unidad	Pedido de Refacciones a Planta	Asignación de operaciones a mecánicos
Reservación de Unidades	Recepción de Refacciones	Surtido de refacciones al taller
Facturación	Inventario de Refacciones	Control de TOT's ¹
Pago del Auto	Remisión de Refacciones	Cierre de Operaciones
Salida del Auto	Facturación de Refacciones	Cierre de la orden
	Pago de Refacciones	Facturación de la Orden
		Pago de la Orden

¿Dónde se realizan esta actividad?

La actividad de las tres áreas operativas mencionadas se realizan en las instalaciones de la agencia:

- Autos → Sala de Venta
- Refacciones → Almacén
- Servicio → Taller

¿Quiénes realizan esta actividad

- Autos → Vendedores

¹ TOT's, Transferencia a otro taller. Es una práctica común de subarrendar un trabajo a cualquier otra empresa.

- Autos → Gerentes de Ventas.
- Refacciones → Almacenistas.
- Refacciones → Jefe de Almacén, Gerente de Refacciones.
- Servicio → Asesores.
- Servicio → Mecánicos, Ingenieros de Servicio, Gerentes de Servicio.

¿Cuánto tiempo toman estas actividades?

Una premisa básica será el conocimiento del tiempo en que se lleva a cabo una actividad en la agencia automotriz. Este conocimiento será primordial para exponer de manera clara y concisa como el sistema coadyuvará a reducir tiempos y aumentar ganancias. Tomemos el ejemplo para el área de servicio y analicemos el proceso principal de venta, teniendo en cuenta los siguientes pasos y tiempos:

Actividad	Tiempo
Elaboración de Presupuesto de servicio usando tablas manuales	10 Minutos
Captura de Orden de servicio a partir del documento del presupuesto	2 Minutos
Asignación de Operaciones a Mecánicos	1 Minuto
Requisición de Refacciones al Almacén	5 Minutos
Elaboración Manual de la Factura	5 Minutos
Recibo de Pago al cliente	3 Minutos
Elaboración del documento y Encuesta de salida	5 Minutos
TOTAL	=10+2+1+5+5+3+5 = 30 Minutos

Ahora, veamos este mismo proceso y la forma en que sistema optimizaría estos tiempos

Actividad	Tiempo
Elaboración de Presupuesto de servicio usando tablas manuales. Aquí se tendrán paquetes de servicio precargados en el sistema, lo cuales contiene la información de las operaciones y refacciones que necesita un mantenimiento de acuerdo al modelo del auto y a sus kilometraje. Ejemplo Paquete de servicio para un Honda Civic 2007 para los 5000 Kmts.	1 Minutos
Captura de Orden de servicio a partir del documento del presupuesto. Se apretará un botón para convertir el presupuesto en orden. Simultáneamente lo mecánicos recibirán la asignación de operaciones sen el taller y las requisiciones de refacciones en la ventanilla del almacén	30 Segundos
Asignación de Operaciones a Mecánicos. Proceso automatizado en el paso 2	0 Minuto
Requisición de Refacciones al Almacén. Proceso automatizado en el paso 2	0 Minutos
Elaboración Manual de la Factura. Impresión de Factura a partir de la Orden previamente elaborada	30 Segundos
Recibo de Pago al cliente. Impresión del recibo automatizada a partir del monto de la factura	30 Segundos
Elaboración del documento y Encuesta de salida. Encuesta precargada sobre la satisfacción del cliente e impresión de la orden de salida al policía.	2 Minutos
TOTAL	=1 + .5 +.5 +.5 + 2 = 5 Minutos

De esta manera bajamos los tiempos de 30 Minutos a 5 Minutos, lo que aumenta la **productividad real en un 500%**

¿Con cuanta frecuencia se realiza esta actividad?

La frecuencia en que se llevan a cabo las actividades de compra-venta en cada departamento es a cada momento, diariamente, de ahí la necesidad de automatizarla para hacerla lo más rápido posible. Además se deben establecer procedimientos institucionales, ya que frecuentemente las cosas suelen hacerse dependiendo del operador, lo cual es incorrecto, ya que puede afectar en la credibilidad de nuestros clientes. Así todos llevarán a cabo exactamente los mismos pasos y en los mismos tiempos en sus procesos diarios. Estos lineamientos serán institucionales, ya que sin importar la rotación del personal, el procedimiento de atención a clientes será siempre el mismo.

¿Quiénes utilizan la información resultante?

Simultáneamente al proceso de Facturación en el área operativa de Autos, Refacciones y Servicio, se asientan registros que utilizarán esa información resultante por otras áreas de la agencia para llevar a cabo su actividad diaria. A Continuación se detalla como se utiliza la información resultante del área OPERATIVA por el área ADMINISTRATIVA

- I. El área de CXC² utilizará la información de las COMPRAS a crédito hechas por la agencia a sus proveedores. Por experiencia podemos decir que estas compras a crédito abarcan el 99 de todas las compras, es decir, prácticamente todas las compras son a crédito. El conocimiento de estos registros de compra será para emitir los cheques de pago a proveedores. En el caso de las Compras de Autos se pagarán a la Planta ensambladora. En el caso de Refacciones a los Proveedores, que pueden ser los fabricantes de refacciones, una refaccionaría genérica, materiales varios; mientras que para el caso de Servicio se pagarán las TOT's³, es decir, cuando la agencia manda a hacer fuera de la agencia el trabajo, por ejemplo una operación de hojalatería y pintura que lo lleve a cabo un maestro hojalatero en su taller, o una operación de afinación y balanceo que sale más barata hacerla en un taller externo que en las propias instalaciones. Toda la información de las compras servirá para el pago a Proveedores
- II. El área CXC utilizará la información de las VENTAS a crédito para llevar el control de los pagos de los clientes. Asimismo establecerá las condiciones crediticias para que al cliente se le pueda facturar a crédito. De esta manera se emitirán los estados de cuenta de los clientes con la información de su saldo.
- III. El área de Contabilidad recibirá simultáneamente los registros tanto de las COMPRAS, como de las VENTAS para elaborar la contabilidad interna en forma de Balanza de Comprobación y Estados Financieros, a fin de que esta información sea enviada a la Gerencia Gerencial, lo cual será utilizado para la toma de decisiones. Asimismo automatizará la emisión y proceso de pólizas contables, para agilizar el manejo de ejercicios contables. Por otra parte ayudará a calcular en base a los registros las obligaciones FISCALES de la agencia como persona moral ante el Gobierno Federal, es decir, el pago de los impuestos.

-Requerimientos de las transacciones de los usuarios

PERFIL DEL SISTEMA. Los sistemas a nivel de transacciones, capturan procesan y almacenan datos por alguna razón. Los analistas seleccionados para trabajar en un sistema para la operación y administración de una agencia de autos deberán conocer todo lo relacionado con la forma en que se procesan estas transacciones. Para entender los requerimientos de las transacciones, los analistas formularán preguntas como las siguientes:

Preguntas que se deben contestar al desarrollar el PERFIL del sistema	
El analista debe dar respuesta a las siguientes preguntas para desarrollar un perfil completo del sistema bajo investigación	
Volumen	¿Cuál es el volumen de las actividades que se presentan? Con que frecuencia ocurren estas actividades? ¿Ocurren las actividades de acuerdo a un ciclo?
Control	¿Qué áreas necesitan un control específico? ¿Cuáles son los métodos de control utilizados? ¿Qué criterios se utilizan para medir y evaluar el desempeño ¿Qué métodos se utilizan para detectar lagunas en los controles? ¿Se toman precauciones específicas de seguridad para protección contra una actividad impropia? ¿Existen métodos para evadir el sistema? ¿Porqué se presentan?
Procesos	¿Qué procesos pasos o funciones contribuyen a esta actividad?

² Cuentas por Cobrar ó Departamento de Cobranza.

³ Transferencias a Otros Talleres ó subarrendar operaciones del Servicio.

	¿Qué es lo que da inicio a la actividad? ¿Cuánto tiempo tarda cada actividad? ¿Qué factores intervienen en la duración de la actividad? ¿Qué retrasos ocurren o pueden ocurrir? ¿Cómo interactúan los elementos entre sí? ¿Cuál es el costo de operación del sistema? ¿Se satisfacen los objetivos específicos de la gerencia?
Datos	¿Qué datos entran al sistema y cuál es su origen? ¿En qué forma se reciben los datos del sistema? ¿En qué forma son almacenados? ¿Qué datos son almacenados en el sistema o como parte de las actividades del mismo? ¿Quiénes utilizan la información generada por el sistema? ¿Con que finalidad la utilizan? ¿Qué es lo que no se utiliza (Partes Extrañas)? ¿Qué datos faltan con mayor frecuencia? ¿Existen datos desatollados o empleados sobre una base <i>ad hoc</i> ? ¿Qué tablas de referencia, diagramas u otros datos se utilizan? ¿Cómo están codificados o abreviados los datos y actividades?
Otros	¿Quiénes son las personas clave en el sistema? ¿Porqué son importantes? ¿Qué obstáculos o influencias del tipo político afectan la eficiencia del sistema?

-Requerimientos de las decisiones de los usuarios

Las decisiones se toman al integrar la información en forma tal que los gerentes puedan saber qué acciones emprender. El sistema debe dar un soporte para las decisiones que son recurrentes, utilizando datos que se originan dentro de la empresa, como los generados por el procesamiento de transacciones.

Cuando el analista investiga sistemas para el soporte de decisiones deben considerar los sistemas de procesamiento de transacciones; y los sistemas eficaces para el soporte de toma de decisiones requieren primero procedimientos adecuados para el procesamiento de transacciones.

Ejemplo: una decisión recurrente que debe tomarse es cuáles son las refacciones que deben comprarse. Hay que considerar que en el modulo de refacciones existen miles de piezas diferentes en venta al público en general, por lo que el stock que se tiene en el almacén puede ser muy grande. El valor que se tiene en el inventario puede ser millones, por lo cual se han desarrollado la estandarización de criterios para generar las compras que se van a ser, ya que no podemos correr el riesgo de comprar piezas que no se van a vender.

La estandarización queda de la siguiente forma. En primer lugar basándonos en la historia de las ventas de cada refacciones, vamos a crear los criterios para clasificar las piezas, quedando de la siguiente manera

Rápida	5 Demandas en 6 Meses de Venta
Mediano Movimiento	3 Demandas en 6 Meses de Venta
Lento	1 Demanda en 6 Meses de Venta
Estancada	3 Demandas en 12 Meses de Venta
Potencialmente Obsoleta	1 Demanda en 12 Meses de Venta
Obsoleta	0 Demandas en 12 Meses de Venta
Nueva	0 Demandas pero con fecha de alta menos a 6 meses

Así diariamente se esta ejecutando este proceso para que se grabe en el KARDEX de la pieza su CLASIFICACIÓN en base a sus ventas. Teniendo las piezas clasificadas se deberá proceder a generar el PEDIDO SUGERIDO, donde vamos a hacer un proceso donde el sistema nos debe recomendar cuantas piezas debemos comprar de determinada refacción, según si esta tiene movimiento rápido, lento, etc.

3.1.3.2 b) Investigación de Requerimientos

TÉCNICAS PARA ENCONTRAR HECHOS. Los analistas utilizan métodos específicos, denominados técnicas para encontrar hechos, con el objeto de reunir datos relacionados con los requerimientos. Entre estos se incluyen la entrevista, el cuestionario, la revisión de los registros (El sitio donde se encuentran estos y la observación. En general, los analistas emplean más de una de estas técnicas para estar seguros de llevar a cabo una investigación amplia y exacta.

-Entrevistas

Los analistas emplean la entrevista para reunir información proveniente de personas o de grupos. Los entrevistados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, los entrevistados son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él. Es importante recordar que los entrevistados y los analistas *conversan* durante la entrevista, es decir no se interroga a los primeros. Las entrevistas dan a los analistas oportunidades para reunir información de las personas que han seleccionado debido a sus conocimientos del sistema que está bajo estudio. A menudo este método es la mejor fuente de información cualitativa (opiniones, políticas, descripciones subjetivas de actividades y problemas). Otros métodos para recolectar hechos son más útiles para recoger datos cuantitativos (números, frecuencias, cantidades).

-Cuestionarios

El uso de cuestionarios permite a los analistas reunir información proveniente relacionada con varios aspectos de un sistema de un grupo grande de personas. El empleo de formatos estandarizados para las preguntas puede proporcionar datos más confiables que otras técnicas; por una parte, su amplia distribución asegura el anonimato de los encuestados, situación que puede conducir a respuestas más honestas. Sin embargo, este método no permite al analista observar las expresiones o reacciones de los encuestados. Asimismo, la respuesta puede ser limitada ya que es posible que no tenga mucha importancia para los encuestados llenar el cuestionario.

-Revisión de Registros

Varios tipos de registros y reportes pueden proporcionar al analista información valiosa con respecto a las organizaciones y a sus operaciones. Al revisar los registros, el analista examina la información asentada en ellos relacionada con el sistema y los usuarios. La revisión de los registros puede efectuarse al comienzo del estudio, como introducción, o también después, y sirve de base para comparar las operaciones actuales, por lo tanto los registros pueden indicar qué está sucediendo.

-Observación

La observación permite al analista ganar información que no se puede obtener por otras técnicas. Por medio de la observación el analista obtiene información de primera mano sobre la forma en que se efectúan las actividades. Este método es más útil cuando el analista necesita observar, por un lado, la forma en que se manejan los documentos y se llevan a cabo los procesos y, por otro, si se siguen todos los pasos especificados. Los observadores experimentados saben qué buscar y como evaluar la significancia de lo que observan.

3.1.3.3 c) Especificación de Requerimientos

HERRAMIENTAS PARA DOCUMENTAR PROCEDIMIENTOS Y DECISIONES. Seguir procedimientos y tomar decisiones son aspectos importantes de cualquier empresa. De hecho, la administración misma es, esencialmente, toma de decisiones. Algunas como aceptar o no ofertas, afectan a todas las organizaciones. Otras, como decidir cuando volver a pedir materiales para el almacén, dependen de pocas personas y siguen procedimientos paso a paso. Sin embargo, las decisiones y procedimientos son de importancia para el analista cuando éste conduce una investigación del sistema dentro de la empresa.

Las herramientas de especificación de requerimientos ayudan al analista a integrar los datos recopilados por los diversos métodos estudiados en la parte de *Determinación de Requerimientos*. Pero como sucede con todas las herramientas, las que emplea el analista para documentar procedimientos y decisiones deben utilizarse adecuadamente.

Conceptos básicos sobre decisiones:

Condiciones y variables de decisión. Cuando se observa un sistema y se pregunta ¿cuáles son las posibilidades? O ¿que puede suceder?, en realidad se está preguntando por las condiciones, que son los posibles estados de una entidad (persona, lugar objetivo o evento). Es indudable que la mayoría de las personas describen automóviles, muebles e incluso a otras personas de acuerdo con sus condiciones buenas y malas. “Bueno” y “malo” son dos condiciones que pueden aplicarse a cada una de las entidades anteriores. Las condiciones cambian y por eso esto que el analista se refiere a ellas como variables de decisión. En una empresa, el manejo de una factura está basada en una condición que constituye una variable de decisión. Algunas organizaciones en que todas las facturas lleven una firma (quizá del contralor o del encargado de efectuar las compras) como requisito para autorizar el pago. En tales casos existen dos alternativas para la recepción de facturas por parte de la organización: con firma o sin ella. La misma factura también puede ser descrita por otras condiciones: autorizada y no autorizada, con el monto correcto o incorrecto. Al documentar la decisión sobre como procesar facturas (o cualquier otro procedimiento), el investigador debe identificar tanto las condiciones permisibles como las relevantes. El hecho de que la factura esté o no firmada es una variable relevante. Sin embargo, el tamaño de la hoja de papel sobre la que está impresa probablemente no lo sea.

Acciones. Cuando se conocen todas las posibles condiciones, el siguiente paso del analista es determinar que hace cuando se presentan algunas de estas. Las acciones son las opciones, que comprenden pasos, actividades o procedimientos, que pueden elegir una persona cuando se enfrenta ante un conjunto de condiciones. En algunos casos las acciones pueden ser bastante sencillas, mientras que en otros muy extensas. El ejemplo del procedimiento de pedidos estudiado anteriormente, demanda acciones específicas que dependen, entre otras cosas, de que la factura esté o no firmada (las condiciones). Si está firmada entonces se verifica si la mercancía fue aceptada (es decir, si llegó la mercancía al almacén y fue aceptada). En este ejemplo las acciones relevantes son iniciar el procesamiento para comprobar el pedido y 2) comenzar el proceso de aceptación de la mercancía. Asimismo, las acciones pueden estar relacionadas con condiciones cuantitativas. Por ejemplo, a menudo las empresas ofrecen descuentos diferentes en la venta de mercancía de acuerdo con el volumen del pedido. Una compañía puede basar el monto de los descuentos sobre tres valores diferentes para la condición VOLUMEN DEL PEDIDO: más de 10000 dólares, entre 5000 y 10000 dólares al menos de 5000 dólares. Las acciones son: 3% de descuento, 2% de descuento y ningún descuento. En muchos procedimientos el analista debe considerar combinaciones de condiciones y acciones.

ÁRBOLES DE DECISIÓN

Las personas tienen diferentes formas de decir lo mismo. Por ejemplo, las condiciones de descuento que se mencionaron en las líneas de arriba también pueden formarse así:

Mayor que 10000 pesos, mayor o igual que 5000 pero menor o igual que 10000 pesos, y menos de 5000 pesos.
No menos de 10000 pesos, no mas de 10000 pero por lo menos de 5000 pesos y no mas de 5000 pesos.

Tener diferentes formas de decir la misma cosa puede crear dificultades de comunicación durante los estudios de sistemas (pueden existir malentendidos sobre los comentarios entre el analista y el gerente u olvidar discutir todos los detalles). Por consiguiente el analista busca las malas interpretaciones. Asimismo, el analista necesita organizar la información recopilada con respecto a la toma de dediciones. Los árboles de de cisión son uno de los tres métodos que se emplean para describir decisiones y que dificultades en la comunicación.

Características de los árboles de decisión.

- Es un diagrama que representa en forma secuencial condiciones y acciones;
- Muestra que condiciones se consideran en primer lugar, en segundo y así sucesivamente.
- Muestra la relación entre cada condición y el grupo de acciones permisibles asociados.
- Los diagramas de este tipo se parecen a las ramas de un árbol, de aquí su nombre.

La raíz del árbol, que aparece en la parte izquierda de la figura, es el punto donde comienza la secuencia de decisión. La rama a seguir depende de las condiciones existentes y de la decisión que debe tomarse. Al avanzar de izquierda a derecha por una rama en particular, se obtiene una serie de toma de decisiones. Después de cada punto de decisión, se encuentra el siguiente conjunto de decisiones o considerar. De esta forma, los nodos del árbol representan condiciones y señalan la necesidad de tomar una determinación relacionada con la existencia de algunas de estas antes de seleccionar la siguiente trayectoria. La parte que se encuentra a la derecha del árbol indica las acciones que deben realizarse, las que a su vez dependen de las secuencia de condiciones que las preceden.

Ejemplo de aplicación del ISAN⁴ en el proceso de facturación de autos nuevos.		
Tipo ISAN	Descripción	Valor
0	Exento	0
1	Isan Tabla	Determinado por un calculo basado en los valores de la tabla suministrada por Hacienda
2	Isan 5%	Corresponde al 5% del precio base de la unidad.
3	Descuento del 7%	Unidades cuyo valor supera un límite establecido, se calcula el del ISAN tipo 1 y luego se le descuenta el 7%.

TABLAS DE DECISIÓN

Más que un árbol la tabla de decisión es una matriz de renglones y columnas que indican condiciones y acciones. Las reglas de decisión, incluidas en una tabla de decisión, establecen el procedimiento a seguir cuando existen ciertas condiciones.

-Características de las tablas de decisión.

Están integrada por cuatro secciones: identificación de condiciones, entradas las condiciones, identificación de acciones y de entradas de acciones. La identificación de condiciones señala aquellas que son relevantes. Las entradas de condiciones indican que valor, si es que lo hay, se debe asociar para una determinada condición. La identificación de acciones enlista el conjunto de todos los pasos que se deben seguir cuando se presenta cierta condición. Las entradas de acciones muestran las acciones específicas del conjunto que deben emprenderse

⁴ ISAN – Impuesto sobre Autos Nuevos

cuando ciertas condiciones o combinaciones de estas son verdaderas. En ocasiones se añaden notas en la parte inferior de la tabla para indicar cuanto utilizar la tabla o para diferenciarla de otras tablas de decisión. Las columnas del lado derecho de la tabla enlazan condiciones y acciones, forman reglas de decisión que establecen las condiciones que deben satisfacerse para emprender un determinado conjunto de acciones. Nótese que se omite el orden de la secuencia (en que las condiciones son examinadas) cosa que no sucede con los árboles de decisión.

-Como construir tablas de decisión.

Para desarrollar tablas de decisión, los analistas deben emprender los siguientes pasos:

1. Determinar los factores considerados como más relevantes en la toma de decisiones. Cada condición seleccionada debe de tener la característica de ocurrir o no ocurrir; no es posible la ocurrencia parcial.
2. Determinar los pasos o actividades mas factibles bajo condiciones que cambian (no solo las condiciones actuales). Esto permite identificar las acciones.
3. Estudiar las diferentes posibilidades de combinaciones de condiciones. Para cualquier numero N de condiciones, existen 2^n combinaciones a considerar. Ejemplo, para 3 condiciones es necesario examinar 8 posibles combinaciones; $2^3=8$ para 4, se tienen $2^4=16$ combinaciones posibles que pueden incluirse en la tabla.
4. Llenar la tabla con las reglas de decisión. Existen dos formas para hacerlo. La primera, y mas larga, es llenar los renglones de condición con valores si o no para cada combinación posible de condiciones. Esto es llenar la primera mitad del renglón con si y la segunda con no. El siguiente renglón se llena alternando con S y N cada 25% del renglón; es decir 25% si, 25% no, 25% y si 25% no. Se repite de nuevo este proceso se llena cada renglón faltante en forma alterna con S y N dividiendo cada vez por potencias sucesivas de 2. (en otras palabras $2^2=4$ para el segundo renglón un $2^3=8$ para el tercer renglón hasta 2^n para el ultimo renglón una N donde N es el numero de condiciones). El otro método para llenar la tabla considera una condición a la vez Y por cada condición adicional la añade a la tabla pero sin considerar las combinaciones de condiciones y acciones duplicadas tal como a continuación se explica.
 - a) Establecer la primera condición y todas las acciones permisibles.
 - b) Añadir la segunda condición duplicando la primera mitad de la matriz y llenando los diferentes valores S y N. De las dos mitades de la matriz aumentada con las nuevas condiciones.
 - c) Para cada condición adicional repetir el paso B.
5. Marcar las entradas correspondientes a las acciones con una X para indicar que estas se emprenden dejar las celdas vacías o marcadas con un – para señalar que en ese renglón no se emprenden ninguna acción. Muestra el resultado obtenido al utiliza este enfoque para desarrollar la tabla de decisión para uno de los ejemplos ya presentados
6. examinar la tabla para detectar reglas redundantes o contradicciones entre estas (caso que se analiza un poco mas adelante).

ESPAÑOL ESTRUCTURADO

El español estructurado es otro método para evitar los problemas de ambigüedad del lenguaje al establecer condiciones y acciones, tanto en procedimientos como en decisiones para describir el proceso. El método no muestra las reglas de decisión; las declara. Aun con esta característica las especificaciones en español estructurado requieren que el analista primero identifique las condiciones que se presentan en un proceso y las decisiones que se deben tomar cuando esto sucede, junto con las acciones correspondientes. Sin embargo, este método también le permite hacer una lista de todos los pasos en el orden en que se llevan a cabo como sin usar símbolos y formatos especiales, características de los árboles y tablas de decisión que para algunos resultan incómodos.

-Desarrollo de declaraciones estructuradas.

El Español Estructurado emplea 3 tipos básicos de declaraciones para describir un proceso: estructuras de secuencia, estructuras de decisión y estructuras de iteración. Estas estructuras son adecuadas para el análisis de decisión y pueden trasladar al desarrollo de software y programación, como se indica en los capítulos de este libro dedicados al desarrollo de software.

Estructuras de secuencia. Es solo un paso o acción incluida en un proceso. Este no depende de la existencia de ninguna condición, y cuando se encuentra, siempre se lleva a cabo, en general, se emplean varias instrucciones en secuencia para describir un proceso. Por ejemplo, es probable que la compra de un libro siga un procedimiento similar al siguiente:

1. escoger el libro deseado
2. llevar el libro al mostrador de salida
3. pagar el libro
4. obtener el recibo
5. abandonar la librería.

Este ejemplo sencillo muestra una secuencia de 5 pasos. Ninguno contiene alguna decisión o condiciones que determinen la realización del siguiente paso. Por otra parte, los pasos se efectúan en el orden mostrado. Por ejemplo, tiene poco sentido pagar por un libro antes de seleccionarlo. Por consiguiente, el procedimiento señala el orden correcto de las acciones.

Estructuras de decisión. El español estructurado es otro camino para mostrar el análisis de decisión. Por tanto, a menudo se incluyen las secuencias de acciones dentro de estructuras de decisión que sirven para identificar condiciones. Es así como las estructuras de decisión aparecen cuando se pueden emprender dos o más acciones, lo que depende del valor de una condición específica. Para esto, primero se evalúa la condición y después se toma la decisión de emprender las acciones o el grupo de acciones asociado con esta condición. Una vez determinada la condición las acciones son incondicionales. Para ilustrar las estructuras de decisión, considérese de nuevo el ejemplo anterior:

Si se encuentra el libro deseado ENTONCES
llevar el libro al mostrador de salida.
Pagar el libro.
Asegurarse de obtener el recibo de compra.
Abandonar la librería.
DE OTRO MODO
No llevar libros al mostrador de salida.
Abandonar la librería.

Estructuras de iteración. En las actividades rutinarias de operación, es común encontrar que algunas de ellas se repiten mientras existen ciertas condiciones o hasta que estas se presentan. Las instrucciones de iteración permiten al analista describir estos casos. La búsqueda de un libro en la librería puede realizarse repitiendo los siguientes pasos.

EJECUTAR MIENTRAS se examinan más libros.
Leer el título del libro
SI el título suena interesante
ENTONCES tomar el libro y ojearlo
Buscar el precio
SI la decisión es llevar el libro
Colocarlo en la pila de LIBROS PARA LLEVAR
Regresar el libro al instante.
FIN DE SI
OTRO continuar
FIN DE EJECUTAR.
SI se encuentran los libros deseados ENTONCES
Llevar los libros al mostrador de salida.
Pagar los libros.
Asegurarse de obtener recibo.
Abandonar la librería.
OTRO
No llevar libros al mostrador de salida
Abandonar la librería.
FIN DE SI

3.1.4 ANÁLISIS ESTRUCTURADO

Cuando los analistas comienzan a trabajar sobre un proyecto de sistemas de información, a menudo tienen que profundizar en un área de la organización con la que tienen poca familiaridad. A pesar de esto deben desarrollar un sistema que ayude a los usuarios de esa área. Cualquier nuevo sistema debe conducir hacia una mejora. Para alcanzar este resultado, se espera que los analistas de sistemas hagan lo siguiente:

- Aprendan los detalles y procedimientos del sistema en uso.
- Obtengan una idea de las demandas futuras de la organización como resultado del crecimiento, del aumento de la competencia en el mercado, de los cambios en las necesidades de los consumidores de la evolución de las estructuras financieras, de la introducción de la nueva tecnología y cambios en las políticas del gobierno entre otros.
- Documentar detalles del sistema actual para su revisión y discusión por otros.
- Evaluar la eficiencia y efectividad del sistema actual y sus procedimientos, tomando en cuenta el impacto sobre las demandas anticipadas para el futuro.
- Recomendar todas las revisiones y ampliaciones del sistema actual, señalando su justificación. Si es apropiado, quizá la propuesta de un nuevo sistema completo.
- Documentar las características del nuevo sistema con un nivel de detalle que permita comprender a otros sus componentes (y su interrelación), y de una manera que permita manejar el desarrollo del nuevo sistema.
- Fomentar la participación de gerentes y empleados en todo el proceso, tanto para aprovechar su experiencia y conocimiento del sistema actual, como para conocer sus ideas, sentimientos y opiniones relacionadas con el nuevo sistema.

También se añade otro requisito: las tareas deben completarse con rapidez y dentro de plazos establecidos, con un mínimo de interrupciones para los usuarios. Para tener éxito, los buenos analistas de sistemas estructuran el proceso que siguen para el desarrollo de un nuevo sistema. Cada lugar donde trabaja un analista es diferente, pero las tareas que llevan a cabo son similares y existe un conjunto común de preguntas por contestar cuando las emprenden.

Análisis estructurado.

El análisis estructurado es un método para el análisis de sistemas manuales o automatizados, que conduce al desarrollo de especificaciones para sistemas nuevos o para efectuar modificaciones a los ya existentes. Cuando los analistas de sistemas abordan una situación poco familiar siempre existe una pregunta de donde comenzar el análisis. Una situación dinámica siempre puede ser vista como abrumadora debido a que muchas de las actividades se llevan a cabo constantemente, el análisis estructurado permite al analista conocer un sistema o proceso (actividad) en una forma lógica y manejable al mismo tiempo que proporciona la base para asegurar que no se omita ningún detalle pertinente.

Significado de “estructurado”.

¿Qué es lo que se desea estructurar? ¿Que significa “estructura”? el objetivo que percibe el análisis estructurado es organizar las tareas asociadas con la determinación de requerimientos para obtener la comprensión completa y exacta de una situación dada. A partir de aquí se determinan los requerimientos que serán la base de un sistema nuevo o modificado.

En el análisis estructurado la palabra estructura significa que: 1) el método intenta estructurar el proceso de determinación de los requerimientos comenzando con la documentación del sistema existente; 2) el proceso esta organizado de tal forma que intenta incluir todos los

detalles relevantes que describen al sistema en uso; 3) es fácil verificar cuando se han omitido detalles relevantes; 4) la identificación de los requerimientos será similar entre varios analistas he incluirá las mejores soluciones y estrategias para las oportunidades de desarrollo de sistemas; 5) los documentos de trabajos generados para documentar los sistemas existente y propuesto son dispositivos de comunicación eficiente.

Componentes del análisis estructurado.

El análisis estructurado hace uso de los siguientes componentes:

- **Símbolos gráficos.** Íconos y convenciones para identificar y describir los componentes de un sistema junto con las relaciones entre estos componentes.
- **Diccionario de datos.** Descripciones de todos los datos utilizados en el sistema. Puede ser manual o automatizado y estar incluido en el diccionario de un proyecto mas grande que quizá contenga las descripciones de los procesos que integran el sistema).
- **Descripciones de procesos y procedimientos.** Declaraciones formales que emplean técnicas y lenguajes que permiten a los analistas describir actividades importantes que forman parte del sistema.
- **Reglas.** Estándares para describir y documentar el sistema en forma correcta y completa.

El método de análisis estructurado se a convertido en el sinónimo de análisis de flujo de datos, que es una herramienta, quizás esto se deba a que la herramienta es esencia para documentar el sistema existente y determinar los requerimientos de información por medio del método estructurado.

¿Qué es el análisis de flujo o de datos?

Los analistas desean conocer las respuestas a cuatro preguntas específicas: ¿que procesos integran el sistema?, ¿que datos emplean cada proceso?, ¿que datos son almacenados? ¿que datos ingresan o abandonan el sistema? De lo anterior es claro que se da la importancia al análisis de los datos. Los datos son la guía de las actividades de la empresa ellos pueden iniciar eventos (por ejemplo, los datos sobre los nuevos pedidos), y ser procesados para dar información útil al personal que desea saber que también se han manejado los eventos (al medir la calidad y tasa del trabajo, rentabilidad, etc.) El análisis de sistemas conoce el papel central que tienen los datos de la empresa en las organizaciones. Seguir el flojo de datos por todos los procesos de la empresa, que es la finalidad del análisis de flujo de datos, les dicen mucho a los analistas sobre como se alcanzan los objetivos de la organización en el transcurso de manejo de transacciones y terminación de tareas, los datos entran, son procesados, almacenados, recuperados, analizados, utilizados, cambiados y presentados como salidas. El análisis de flujo de datos estudia el empleo de los datos en cada actividad documenta los hallazgos con diafragmas de flujo de datos que muestran en forma grafica la relación entre procesos y datos, en los diccionarios de datos que describen de manera formal los datos del sistema y los sitios donde son utilizados.

3.1.4.1 CARACTERÍSTICAS DE LA ESTRATEGIA DE FLUJOS DE DATOS.

El análisis de flujos de datos examina el empleo de los datos para llevar a cabo procesos específicos de la empresa dentro del ámbito de una investigación de sistemas. El análisis puede pensarse de tal manera que se estudian actividades del sistema desde el punto de vista de los datos: donde se originan, como se utilizan o cambian, hacia donde van, incluyendo las paradas a lo largo del camino que siguen desde su origen hasta su destino. Los componentes de la estrategia de datos abarcan tanto la determinación de requerimientos como al diseño de sistemas. Una notación bien establecida facilita la documentación del sistema actual y si análisis por todos los participantes en el proceso de determinación de requerimientos.

Herramientas de la estrategia de flujo de datos.

La estrategia de flujo de datos muestra el empleo de estos en forma gráfica. Las herramientas utilizadas al seguir esta estrategia muestran todas las características esenciales del sistema y la forma en que se ajustan entre sí. Puede ser difícil comprender en su totalidad un proceso de la empresa si se emplea para ello solo una descripción verbal; las herramientas para el flujo de datos ayudan a ilustrar los componentes esenciales de un sistema junto con sus interacciones. El análisis de flujo de datos utiliza las siguientes herramientas:

1. Diagrama de flujos de datos. Una herramienta gráfica se emplea para describir y analizar el movimiento de datos a través de un sistema, ya sea que este fuera manual o automatizado, concluyendo procesos, lugares para almacenar datos y retrasos en el sistema. Los diagramas de flujo de datos son la herramienta más importante y la base sobre la cual se desarrolla otros componentes. La transformación de datos la entrada en salida por medio de procesos puede describirse en forma lógica e independiente de los componentes físicos (por ejemplo, computadoras, gabinetes de archivos, unidades de discos y procesadores de texto) asociados con el sistema. Estos reciben el nombre de diagramas de flujo de datos.
2. Diccionario de datos. El diccionario contiene las características lógicas de los sitios donde se almacenan los datos del sistema incluyendo nombre, descripción, alias, contenido y organización, también identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información. Sirve como punto de partida para identificar los requerimientos de las bases de datos durante el diseño del sistema.
3. Diagrama de estructura de datos. Es una descripción de la relación entre entidades (personas, lugares, eventos y objetos) de un sistema del conjunto de información relacionado con la entidad. No considera el almacenamiento físico de los datos.
4. Grafica de estructura. Herramienta de diseño que muestra con símbolos la relación entre los módulos de procesamiento y el software de la computadora. Describen la jerarquía de los módulos componentes y los datos que serán transmitidos entre ellos. Incluye el análisis de las transformaciones entrada – salida y el análisis de transacciones.

Notación.

Los diagramas lógicos de flujo de datos se puede dibujar con solo cuatro notaciones sencillas, es decir con símbolos especiales o iconos y anotaciones que los asocien con un sistema específico. El uso de iconos especiales para cada elemento depende de que enfoque se utilice, el de Yourdon o el de Gane Isarson:

1. **Flujo de datos** movimiento de datos en determinada dirección desde un origen hacia un destino en forma de documentos, cartas, llamadas u otro medio. Es un “paquete” de datos.
2. **Procesos.** Personas, procedimientos o dispositivos que utilizan y producen (transforma) datos. No se identifica el componente físico.

3. **Fuente o destino de los datos.** Fuentes o destinos externos de datos que pueden ser personas, programas, u otras entidades que interactúan con el sistema pero se encuentran fuera de su frontera. Los términos fuente y pozo son intercambiables con origen y destino.
4. **Almacenamiento de datos.** Es el lugar donde se guardan los datos o al que hacen referencia los procesos en el sistema.

Cada componente es un diagrama de flujo de datos tiene una etiqueta con un nombre descriptivo. Los nombres de los procesos también reciben un número que es utilizado con fines de identificación. El número asignado a determinado proceso no tiene que ver con la secuencia de procesos. Sirve estrictamente para identificación y tiene un valor adicional cuando se estudian los componentes que integran un proceso específico.

Actividades paralelas.

Todas las organizaciones tienen muchas actividades que ocurren al mismo tiempo con flujos de datos concurrentes. Los diagramas de flujo de datos permiten a los analistas representar las actividades con mayor exactitud al mostrarlas cuando ocurren al mismo tiempo. Como su nombre lo sugiere, los diagramas de flujos de datos se concentran en el movimiento de datos a través del sistema no en los dispositivos o el equipo los analistas identifican y describen desde el inicio hasta el final del proceso, para comprender una área de aplicación a los datos que fluyen por todo el sistema entonces explican porque los datos entran y salen y cual es el procesamiento que se realiza con ellos.

Ventajas del análisis del flujo de datos.

Los analistas pueden trabajar con los usuarios y lograr que participen en el estudio de los diagramas de flujo de datos. Los usuarios pueden hacer sugerencias para modificar los diagramas con la finalidad de escribir la actividad con mayor exactitud. Así mismo pueden examinar las graficas y reconocer con rapidez problemas; esto permite efectuar las correcciones necesarias antes de que comiencen otras tareas relacionadas con el diseño si los problemas no son detectados en las primeras fases del proceso de desarrollo, entonces será difícil corregirlos cuando aparezcan mas adelante. Evitar los errores desde el inicio puede prevenir una posible falla del sistema.

El análisis de flujo de datos permite a los analistas aislar áreas de interés en la organización y estudiarlas al examinar los datos que entran en el proceso, de tal manera que puedan observar la manera en que cambian cuando lo abandonan. A medida que los analistas reúnen hechos y detalles, comprenden mejor el proceso; esto los conduce a formular preguntas relacionadas con aspectos específicos del mismo y los lleva a una investigación adicional. Muestra los aspectos generales de esta metodología; la investigación se divide en detalles que tienen cada vez un nivel menor hasta que se comprenden todos los componentes esenciales junto con sus interrelaciones. Una extensa investigación de sistemas producen muchos conjuntos de diagramas de flujo de datos, algunos brindan panoramas de procesos importantes mientras que otros nos muestran con bastante detalle elementos datos, almacenes de datos, y pasos de procesamientos para componentes específicos de un sistema grande. Si los analistas desean revisar después todo el sistema, primero utilizan los diagramas de alto nivel, aquellos que contienen la visión panorámica de todo el sistema. Sin embargo, si están interesados en estudiar un proceso en particular entonces emplean el diagrama de flujo de datos de bajo nivel correspondiente a ese proceso.

3.1.4.2 DESARROLLO DE DIAGRAMAS DE FLUJO DE DATOS.

Un diagrama de flujo de datos DFD ⁵ es una representación gráfica del "flujo" de datos a través de un sistema de información. Un diagrama de flujo de datos también se puede utilizar para la visualización de procesamiento de datos (diseño estructurado). Es una práctica común para un diseñador dibujar un contexto a nivel de DFD que primero muestra la interacción entre el sistema y las entidades externas. Este contexto a nivel de DFD se "explotó" para mostrar más detalles del sistema que se está modelando.

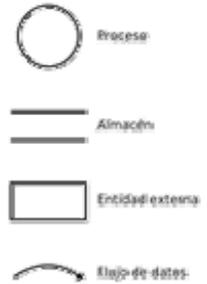
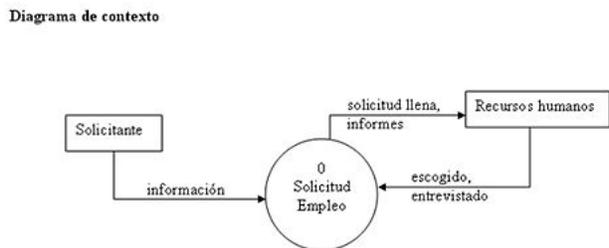


Imagen 4.
Herramientas DFD

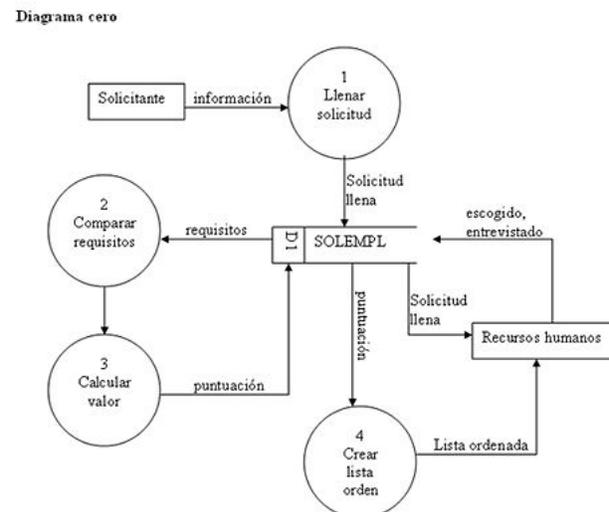
Los datos de diagramas de flujo fueron inventados por Larry Constantine, el desarrollador original del diseño estructurado, basado en el modelo de computación de Martin y Estrin: "flujo gráfico de datos". Los diagramas de flujo de datos (DFDs) son una de las tres perspectivas esenciales de Análisis de Sistemas Estructurados. El patrocinador de un proyecto y los usuarios finales tendrán que ser informados y consultados en todas las etapas de una evolución del sistema. Con un diagrama de flujo de datos, los usuarios van a poder visualizar la forma en que el sistema funcione, lo que el sistema va a lograr, y cómo el sistema se pondrá en práctica. El antiguo sistema de diagramas de flujo de datos puede ser elaborado y se comparó con el nuevo sistema de diagramas de flujo para establecer diferencias y mejoras a aplicar para desarrollar un sistema más eficiente. Los diagramas de flujo de datos pueden ser usados para proporcionar al usuario final una idea física de cómo resultarán los datos a última instancia, y cómo tienen un efecto sobre la estructura de todo el sistema. La manera en que cualquier sistema es desarrollado puede determinarse a través de un diagrama de flujo de datos. El desarrollo de un DFD ayuda en la identificación de los datos de la transacción en el modelo de datos.

Los diagramas derivados de los procesos principales se clasifican en niveles, los cuales son:

Nivel 0. En el diagrama de contexto solo modela, o dibuja, el proceso principal del problema en cuestión con sus respectivas entidades. Cada proceso debe tener al menos una entrada y una salida (de datos).



Nivel 1. En el diagrama de nivel superior se plasman todos los procesos que describen al proceso principal, o sea, éste se descompone en varios procesos. En este nivel aparecen los almacenes, los cuales tienen la capacidad de almacenar o enviar datos para ser usados en los distintos procesos.



Nivel 2. En el diagrama detalle se generan procesos provenientes del nivel anterior. Cabe destacar que en el nivel 1 y 2 siempre los procesos deben tener las entradas y las salidas dadas en el diagrama de contexto.

⁵ Diagrama de Flujo de Datos

Imagen 5. Ejemplo DFD

3.1.4.3 CARACTERÍSTICAS DEL DICCIONARIO DE DATOS.

En muchas aplicaciones de sistemas de información, el volumen de datos es esencial (mas de lo que un analista puede seguir de cerca con facilidad). Cuando se trabajan sobre un sistema varios equipos de analistas, la tarea de coordinar las definiciones de los datos se vuelve más compleja. Los individuos dependen de las definiciones establecidas por otros y de sus propias suposiciones con respecto a las especificaciones de los datos. Debido a esto, a menos que todos se reúnan casi todos los días (lo que ocurre muy rara vez) es virtualmente imposible tener una coordinación aceptable de datos. Los diccionarios de datos son un componente importante del análisis estructurado ya que por si solos los diagramas de flujo de datos no describen el objeto de la investigación. El diccionario de datos proporciona más información relacionada con el sistema.

¿Que es un diccionario de datos?

Un diccionario de datos es un catalogo, un deposito, de los elementos en un sistema. Como su nombre lo sugiere, estos elementos se centran alrededor de los datos y la forma en que están estructurados para satisfacer los requerimientos de los usuarios y las necesidades de la organización. En un diccionario de datos se encuentran la lista de todos los elementos que forman parte del flujo de datos en todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario guarda los detalles y descripciones de todos estos elementos. Si los analistas desean conocer cuantos caracteres abarca un determinado dato o que otros nombres recibe en distintas partes del sistema, o donde se utiliza, encontraran las respuestas en un diccionario de datos desarrollados en forma apropiada. El diccionario se desarrolla durante el análisis de flujos de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema.

Importancia del diccionario.

Los analistas utilizan los diccionarios de datos por 5 razones importantes:

1. Para manejar los detalles en sistemas grandes
2. Para comunicar un significado común para todos los elementos del sistema.
3. Para documentar las características del sistema
4. Para facilitar el análisis de los detalles con la finalidad de evaluar las características y determinar donde efectuar cambios en el sistema.
5. Localizar errores y omisiones en el sistema.

Manejo de detalles.

Los sistemas grandes tienen enormes volúmenes de datos que fluyen por ellos en forma de documentos, reportes e incluso platicas (aun en los sistemas pequeños se observan grandes cantidades de datos). De manera similar, se llevan a cabo muchas actividades que utilizan los datos existentes o que generan nuevos detalles recuérdese, como se menciono en la historia al inicio de este capitulo, que todos los sistemas experimentan cambios continuos y manejar de manera completa todos los detalles es un desafío. Con franqueza, es imposible que los analistas recuerden todo, los que tratan de hacerlo cometen de manera invariable equivocaciones u olvidan elementos importantes. Los mejores analistas no intentan recordarlo todo, en lugar de hacerlo registran toda la información. Algunos lo hacen sobre hojas de papel y otros quizá sobre tarjetas indexadas. Muchos se emplean para tal fin un procesador de palabras y una computadora personal por supuesto. Los analistas mejor organizados y mas eficaces utilizan diccionarios de datos automatizados diseñados de manera especifica para el análisis y diseño de sistemas. En este capitulo se muestra la forma en que operan este tipo de diccionarios.

Comunicación de significados.

Los diccionarios de datos proporcionan asistencia para asegurar significados comunes para los elementos y actividades del sistema. Si se examina una muestra de diagramas de flujo de datos para el procesamiento de pedidos (ejemplo ya presentado en este capítulo) es probable que se tengan pocas dificultades para comprender que datos representan a la factura y al cheque. Los dos son términos comunes en el mundo de los negocios y muchas personas conocen su significado. Pero, ¿lo que una persona entiende significa lo mismo para otra?. ¿FACTURA significa la cantidad adeudada al proveedor?; ¿el monto incluye el impuesto y los gastos de envío; ¿ como se identifica una factura entre varias? Obtener respuestas para estas preguntas o verificar las suposiciones hechas con respecto a lo que significa las respuestas ayudara a aclarar y a definir los requerimientos del sistema al describir de manera mas completa los datos utilizados y generados por el. Los diccionarios de datos registran detalles adicionales relacionados con el flujo de datos en el sistema de tal forma que todas las personas participantes puedan localizar con rapidez la descripción de flujos de datos, almacenes de datos o procesos.

Documentación de las características del sistema.

Documentar las características de un sistema es la tercera razón para utilizar los sistemas de diccionario de datos. Las características incluyen partes o componentes así como los aspectos que los distinguen. Claro esta que además se desea conocer información relacionada con los procesos y almacenes de datos pero también es necesario saber bajo que circunstancias se lleva a cabo cada proceso y con cuanta frecuencia ocurren estas. Tener la descripción formal de las características del sistema produce una comprensión mas completa de este. Una vez que las características están articuladas y registradas, todos los participantes en el proyecto tendrán una fuente común de información con respecto al sistema.

Facilidad de análisis.

La cuarta razón para hacer uso de los diccionarios de datos es determinar si son necesarias nuevas características o si están en orden los cambios de cualquier tipo. Supóngase que un analista trabaja con una universidad que esta considerado permitir a sus estudiantes que se inscriban en los cursos por medio de una llamada telefónica al sistema en línea de inscripciones ¿qué preguntas debe formular el analista y que información desearía tener disponible para examinarla? En cualquier situación, incluida la de este ejemplo, los analistas de sistemas abordan las siguientes características del sistema:

- **Naturaleza de las transacciones.** Las actividades de la empresa se llevan a cabo mientras se emplea un sistema, incluidos los datos necesarios para aceptar, autenticar, y procesar cada actividad. Ejemplo: ¿ permite el sistema el procesamiento de las transacciones de inscripción a cursos para las que el pago se efectúa por medio de una tarjeta de crédito bancaria? ¿ que características adicionales son necesarias para permitir la inscripción por medio de una llamada telefónica? ¿ en que forma se recibirán los pagos si los estudiantes deciden no hacerlos por medio de una tarjeta de crédito?
- **Preguntas.** Solicitudes para la recuperación o procesamiento de información para generar una respuesta especifica. Ejemplo: los datos del estudiante y los que describen los cursos se encuentran en dos archivos separados que no están ligados? Como se pueden poner en forma conjunta a disponibilidad de los asesores que desean ayudar a los estudiantes en la plantación de sus programas de estudio y horarios de clase?
- **Salida y generación de reportes.** Resultados del procesamiento hecho por el sistema que son presentados a los usuarios en una forma aceptable para ellos. Ejemplo: ¿ como identificar aquellos estudiantes que se inscribirán por teléfono para generar un listado de

ellos que pueda incluirse en un reporte por separado? ¿ como dotar a estos estudiantes con expediente firmado igual al que tienen ahora los que acuden a inscribirse a la universidad?

- **Archivos y bases de datos.** Detalles de las transacciones y registros maestros que son de interés para la organización. Ejemplo ¿ que datos deben capturarse para verificar la exactitud y autenticidad de las transacciones que se realizan por vía telefónica?.
- **Capacidad del sistema.** Habilidad del sistema para aceptar, procesar y almacenar transacciones y datos. Ejemplo: ¿cuantos estudiantes se pueden registrar al mismo tiempo por vía telefónica? ¿cual es el numero actual y esperado de estudiantes que pueden inscribirse en una hora?.

Localización de errores y omisiones.

Tener información en un diccionario relacionada con las características del sistema- transacciones, consultas datos y capacidad- dice mucho con respecto al sistema y permite evaluarlo. Pero para esto es necesario saber que la propia información es completa y exacta. Por consiguiente, los diccionarios se emplean por una quinta razón: localizar errores en la descripción del sistema. El conflicto entre descripciones de flujos de datos procesos que nunca reciben entradas o producen salidas, almacenes de datos q nunca se actualizan etc. Son indicadores de un análisis incorrecto o incompleto, se debe corregir esta situación antes de determinar los cambios que sean necesarios los sistemas automatizados de diccionarios de datos tienen características que detectan estas dificultades y las presentan en un reporte. Aun en los diccionarios manuales, el proceso de registrar la información revela los errores.

Contenido de un registro del diccionario.

Todas las partes de un sistema de información – transacciones, consultas, reportes, salidas, archivos y bases de datos- dependen de los datos. El diccionario contiene dos tipos de descripciones para el flujo de datos dentro del sistema: elementos de datos y estructuras de datos. Los elementos de datos se agrupan para formar una estructura de datos.

Elemento dato. El nivel mas importante de datos es el elemento dato (es probable que usted conozca otros nombres que se le dan a este termino: campo, dato o parte elemental. Ninguna unidad mas pequeña tiene significado para los analistas de sistemas o usuarios. Por ejemplo el número de la factura, su fecha de expedición y la cantidad adeudada son elementos dato incluidos en el flujo de datos de la facturación.

Los elementos dato son los bloques básicos para todos los demás datos del sistema. Por si mismos no conllevan suficiente significado para ningún usuario. Por ejemplo, el significado de FECHA en relación con una factura es claro para todos los usuarios: es la fecha en que expidió la factura. Sin embargo, fuera de este contexto no tiene ningún significado. Quizá sea la fecha de pago, de graduación, de inicio o la de expedición de la factura.

Estructuras de datos. Una estructura de datos es un grupo de datos elementales que están relacionados con otros y que en conjunto describen un componente del sistema. Por ejemplo la estructura de datos FACTURA esta definida por un grupo de datos elementales que incluyen la fecha de expedición de la factura, el vendedor, la dirección de este y detalles relacionados con los artículos que ampara la factura.

3.1.4.4 REGISTRO DE LAS DESCRIPCIONES DE DATOS

Dado que las descripciones de los datos se utilizarán una y otra vez durante toda la investigación y después durante el diseño, es aconsejable adoptar un formato fácil de usar tanto para el registro como para la recuperación de detalles cuando sea necesario.

Definición de los flujos y almacenes de datos. Una explicación completa de todos los elementos del diagrama de flujo de datos incluye una descripción de cada flujo de datos, estructuras de datos y procesos. Todos los detalles son capturados en una forma especial para el flujo de datos. Cada flujo de datos recibe un nombre y se describe de manera breve. Asimismo, se incluyen los nombres y la identificación de los procesos asociados con el flujo de datos. Para completar la definición del flujo de datos, se listan todas las estructuras de datos apropiadas.

Definición de estructuras de datos. Los flujos y almacenes de datos son estructuras de datos. Dicho de otra forma, si las estructuras de datos están en movimiento reciben el nombre de flujo de datos. En contraste, las estructuras de datos que no están en movimiento se denominan almacenes de datos. Todas las estructuras de datos están definidas en una entrada del diccionario de datos. El volumen de detalles indica el nivel de actividad, así como el número de transacciones o la rapidez de cambio, todo esto para un dato durante un periodo determinado de tiempo.

Descripción de Procesos. También proporciona una definición por separado de cada proceso del sistema.

Uso de los detalles contenidos en el diccionario de datos. Tener un conjunto de definiciones concisas para todas las entidades del proceso bajo estudio es algo muy valioso. El diccionario de datos es la única fuente común de definiciones para los usuarios e investigadores. El propio proceso de desarrollo del diccionario de datos, obliga a los analistas a clarificar su comprensión de los datos del sistema. Por añadidura, el propio diccionario de datos puede ser procesado para revelar información adicional:

1. Listado de los elementos dato y estructura de datos.
2. Listado de Procesos.
3. Verificación de referencias cruzadas
4. Detección de errores.
5. Descubrimiento de inconsistencias.

3.2 DISEÑO

El diseño es desde mi punto de vista la parte más importante de todo el ciclo. Aquí es donde se concibe la parte medular que soportará toda la operación. En esta etapa crucial el talento y la creatividad del arquitecto del software determinarán el éxito o el fracaso del sistema. Así de simple.

Una columna vertebral firme se traduce en una operación confiable. La carencia de una estructura adecuada será la causa de futuros dolores de cabeza tratando de “parchar” lo que fue incorrectamente concebido desde el principio.

¿Qué es? El diseño es lo que casi cualquier ingeniero quiere hacer. Es el sitio donde manda la creatividad, donde los requisitos del cliente, las necesidades de negocio y las consideraciones técnicas se unen en la formulación de un producto o sistema. El diseño crea una representación o modelo de software, pero a diferencia del modelo del análisis (que se enfoca en la descripción de los datos, las funciones y comportamiento requerido), **el modelo de diseño proporcionará detalles acerca de estructuras de datos, arquitecturas, interfaces y componentes de software que son necesarios para implementar el sistema.**

¿Quién lo hace? Los ingenieros de software encabezan cada una de las tareas de diseño.

¿Por qué es importante? El ingeniero permite al ingeniero de software modelar el sistema o producto que se va a construir. Este modelo puede evaluarse en relación con su calidad y mejorarse antes de generar el código, de realizar pruebas y de que los usuarios finales se vean involucrados a gran escala. El diseño es el sitio donde se establece la calidad del software.

¿Cuáles son los pasos? El diseño presenta el software de diferentes formas. Primero debe representarse la arquitectura del sistema o producto. Después, se modelan las interfases que conectan el software con los usuarios finales, con otros sistemas y dispositivos y con los propios componentes que lo constituyen. Por último, se diseñan los componentes del software que se utilizarán en la construcción del sistema. Cada una de estas versiones representa una acción de diseño diferente, pero todas deben ajustarse a un conjunto de conceptos básicos del sistema que determinan todo el trabajo de diseño.

¿Cuál es el producto obtenido? Un modelo que abarca representaciones arquitectónicas, de interfaz, en el nivel de componentes y de despliegue.

¿Cómo puedo estar seguro de que lo he hecho correctamente? El modelo del diseño lo evalúa el equipo de software en un esfuerzo encaminado a determinar si éste contiene errores, inconsistencias u omisiones; si existen mejores alternativas; y si el modelo puede implementarse dentro de las restricciones, el itinerario y el costo que han sido establecidos.

La meta de la ingeniería del diseño es producir un modelo de representación que muestre firmeza, comodidad y placer. Para lograrlo un diseñador debe practicar la diversificación y después la convergencia. “La diversificación es la adquisición de un repertorio de alternativas, la materia prima del diseño: componentes, soluciones de componentes y conocimiento, todo contenido en catálogos, libros de texto y en la mente”. Una vez que se ha integrado este conjunto de información, el diseñador debe elegir y tomar elementos de repertorio que cumplan los requisitos definidos por la ingeniería de requisitos y el modelo de análisis. Cuando esto ocurre se consideran y se rechazan las alternativas, y el ingeniero de diseño converge en “una configuración particular de componentes y, por lo tanto, en la creación de un producto final”

La diversificación y la convergencia demandan intuición y juicio. Estas cualidades están basadas en la experiencia de construir entidades similares, un conjunto de principios que guían como evoluciona el modelo, un conjunto de criterios que permiten juzgar la calidad, y un proceso de iteración que conduce a una representación del diseño final.

La ingeniería del diseño de software está en un cambio continuo, en la medida en que evolucionan mejores métodos, mejores análisis y una comprensión más amplia. Aun en la actualidad, la mayoría de las metodologías del diseño de software carecen de profundidad, flexibilidad y naturaleza cuantitativa, que por lo general se asocian a disciplinas de diseño de ingeniería más clásicas.

3.2.1 DISEÑO DENTRO DEL CONTEXTO DE LA INGENIERÍA DE SOFTWARE.

El diseño de software se encuentra en el núcleo técnico de la respectiva ingeniería y se aplica de manera independiente al modelo de software que se utilice. Una vez que se analizan y especifican los requisitos, el diseño de software es la última acción de la ingeniería correspondiente dentro de la actividad del modelado, la cual establece una plataforma para la construcción (generación de código y pruebas). Cada uno de los elementos del modelo del análisis proporciona la información necesaria para crear los cuatro modelos de diseño que se requieren para una especificación completa de diseño:

El **diseño de datos-clase** transforma los modelos de análisis y clases en las clases de diseño y las estructuras de datos que se requieren para implementar el software. Las clases y relaciones que definen las tarjetas índice CRC⁶ y el contenido detallado de datos que muestran los atributos de clase y otras notaciones proporcionan la base para la actividad de diseño de datos. Una parte del diseño de clases puede ocurrir en conjunto con el diseño de la arquitectura del software. El diseño de clase más detallado se realiza a medida que se diseña cada componente del software.

El **diseño arquitectónico** define la relación entre los elementos estructurales más importantes del software, los estilos arquitectónicos y patrones de diseño que pueden usarse para satisfacer los requisitos definidos por el sistema, y las restricciones que afectan la manera en que pueden implementarse los patrones arquitectónicos. La representación del diseño arquitectónico – el marco de trabajo de un sistema basado en computadora – puede derivarse de la especificación del sistema, del modelo de análisis y de la interacción de los subsistemas definidos dentro del modelo de análisis.

El **diseño de la interfaz** describe la forma en que el software se comunica con los sistemas que interactúan con él y con los humanos que lo utilizan. Una interfaz implica un flujo de información (por ejemplo, datos o control) y un tipo de comportamiento específico. Por lo tanto, los escenarios de uso y los modelos de comportamiento proporcionarán mucha de la información que se requiere en el diseño de la interfaz.

El **diseño a nivel de componentes** transforma los elementos estructurales de la arquitectura del software en una descripción procedimental de los componentes de éste. La información obtenida de los modelos basados en clases, los modelos de flujo y los modelos de comportamiento sirven como base para el diseño de componentes.

Durante el diseño se toman decisiones que al final incidirán en el éxito de la construcción del software, así como en, con el mismo grado de importancia, la facilidad con que el software puede mantenerse. Pero, ¿Por qué es tan importante el diseño? La importancia del Diseño puede describirse con una sola palabra: *calidad*. El diseño es la etapa en la que se fomentará la calidad en la ingeniería del software. El diseño proporcionará las representaciones del software susceptibles de evaluar respecto a la calidad. El diseño es la única forma en que se puede, de manera exacta, un requisito del cliente se puede convertir en un sistema o producto de software terminado. El Diseño sirve como fundamento para todas las actividades subsecuentes de la ingeniería del software y del soporte de éste. Sin diseño se corre el riesgo de construir un sistema inestable, el cual fallará cuando se realicen cambios pequeños; difícil de probar; cuya calidad no podrá evaluarse.

⁶ El modelo CRC es una colección de tarjetas CRC (Clase – Responsabilidad - Colaborador)

3.2.2 PROCESO Y CALIDAD DEL DISEÑO.

El diseño del software es un proceso iterativo mediante el cual los requisitos se traducen en un “plano” para construir el software. Al inicio, el plano representa una visión holística del software. Es decir, el diseño se representa en un grado alto de abstracción, el cual puede rastrearse de manera directa hasta conseguir el objetivo específico del sistema y requisitos más detallados de comportamiento, funcionales y de datos. A medida en que ocurren las iteraciones del diseño a grados mucho más bajos de abstracción. Estos grados aún se pueden rastrear hasta los requisitos, pero la conexión es más sutil.

A través del proceso del diseño, la calidad en evolución de este se evalúa con una serie de revisiones técnicas formales o con revisiones de diseño. McGlaughlin sugiere tres características que sirven como guía en la evaluación de un buen diseño:

- Debe implementar todos los requisitos explícitos contenidos en el modelo del análisis, y debe ajustarse a todos los requisitos implícitos que desea el cliente.
- Debe ser una guía legible y comprensible para quienes generan código y quienes realizan pruebas y, en consecuencia, dan soporte al software.
- Debe proporcionar una imagen completa del software -dando dirección a los dominios de datos, funcionales y de comportamiento- desde una perspectiva de implementación.

Cada una de estas características es en realidad una meta del proceso del diseño. Pero, ¿cómo se alcanza cada una de ellas?

Directrices de Calidad. Con el fin de evaluar la calidad de una representación de diseño se deben establecer los criterios técnicos para un buen diseño. En secciones posteriores de este capítulo se expondrán los conceptos de diseño que también sirven como criterios de calidad de software. Por ahora se presentan las siguientes directrices:

1. Debe presentar una estructura arquitectónica que a) se haya creado mediante patrones de diseño reconocibles, b) la integren componentes que exhiban buenas características de diseño, y c) puedan implementarse de manera evolutiva, para que de esta forma facilite la implementación y las pruebas.
2. Debe ser modular, esto es, el software deberá dividirse de manera lógica en elementos y subsistemas.
3. Debe contener distintas representaciones de los datos, la arquitectura, las interfaces y los componentes.
4. Debe conducir a estructuras de datos que sean apropiadas para las clases que habrán de implementarse y que procedan de patrones de datos reconocibles.
5. Debe conducir a componentes que presenten características funcionales independientes.
6. Debe conducir a interfaces que reduzcan la complejidad de las conexiones entre los componentes y el ambiente externo.
7. Debe obtenerse por medio de un método repetible que se base en la información obtenida durante el análisis de requisitos del software.
8. Debe representarse por una notación que comunique de manera eficaz su significado.

Estas directrices de diseño no se logran por casualidad. El proceso de diseño del software fomenta el buen diseño aplicando principios fundamentales de diseño, una metodología sistemática y una revisión cuidadosa.

Atributos de Calidad. Hewlett-Packard desarrolló un conjunto de atributos de calidad, entre ellos la funcionalidad, la facilidad de uso, la confiabilidad, el desempeño y la soportabilidad. Estos atributos de calidad representan un objetivo para todo el diseño de software:

- La funcionalidad se estima al evaluar el conjunto de características y capacidades del programa, la generalidad de las funciones que se entregan y la seguridad total del sistema.
- La facilidad de uso se valora al considerar los factores humanos, la estética, consistencia y documentación generales.
- La confiabilidad se evalúa al medir la frecuencia y severidad de las fallas, la precisión de los resultados de salida, la media del momento de fallas, la habilidad para recuperarse de las fallas y la previsibilidad del programa.
- El desempeño se mide con la velocidad de procesamiento, tiempo de respuesta, consumo de recursos, rendimiento y eficacia.
- La soportabilidad combina la habilidad de extender el programa (extensibilidad), la adaptabilidad y la serviciabilidad -estos tres atributos representan un concepto más común, *facilidad de mantenimiento*- además, resistencia a pruebas, compatibilidad, configurabilidad (habilidad de organizar y controlar elementos de configuración del software), la facilidad con la que puede instalarse el sistema, y la facilidad con que se pueden localizar los problemas.

No todos los atributos de la calidad del software tienen el mismo peso cuando se desarrolla el diseño de software. Es posible que otra demande desempeño con un enfoque particular en la velocidad de procesamiento. Una tercera puede centrarse en la confiabilidad. Sin importar el peso, es importante puntualizar que estos atributos de calidad deben considerarse al comienzo del diseño, no después de que el diseño esté completo y haya comenzado la construcción.

3.2.3. CONCEPTOS DEL DISEÑO.

A través de la historia de la ingeniería de software ha evolucionado un conjunto de conceptos fundamentales de diseño de software. Aunque el grado de interés en cada concepto ha variado con los años, han pasado la prueba del tiempo. Cada uno ofrece al ingeniero de software un fundamento sobre el cual pueden aplicarse métodos de diseño más elaborados. Los conceptos fundamentales del diseño de software ofrecen el marco de trabajo necesario para hacer las cosas "del modo correcto".

Abstracción. Cuando se considera una solución modular a cualquier problema se pueden exponer muchos grados de abstracción. En un alto grado de abstracción una solución se establece en términos generales con el lenguaje del entorno del problema. En los grados de menos abstracción se proporciona una descripción más detallada de la solución. En la medida en que cambien los diferentes grados de abstracción se trabaja para crear abstracciones procedimentales y de datos. Una abstracción procedimental se refiere a una secuencia de instrucciones que tienen una función específica y limitada. El nombre de abstracción procedimental implica estas funciones, pero se omiten detalles específicos. Ejemplo de abstracción procedimental sería la palabra *abrir* para una puerta. Abrir implica una larga secuencia de pasos procedimentales (caminar a la puerta, tomar la manija, dar vuelta a la manija y empujar la puerta, hacerse a un lado para abrir la puerta, etc.). Una abstracción de datos es una colección nombrada de datos que describe un objeto de datos. En el contexto de abstracción procedimental, *abrir* se puede definir como una abstracción de datos llamada puerta. Como cualquier objeto de datos, la abstracción de datos para puerta abarcaría una serie de atributos que la describan (tipo, dirección de apertura, mecanismo de apertura, peso, dimensiones). Se puede decir que la abstracción procedimental *abrir* emplearía la información contenida en los atributos de la abstracción de datos *puerta*.

Arquitectura. La *arquitectura de software* alude a "la estructura del software y las formas en que la estructura proporciona una integridad conceptual para un sistema". En su forma más simple, la arquitectura es la estructura u organización de los componentes del programa (módulos), la manera en que estos componentes interactúan, y la estructura de datos que utilizan los componentes. En un sentido más amplio, sin embargo, los componentes pueden generalizarse para representar elementos importantes del sistema y sus interacciones. Una de las metas del diseño de software es derivar una representación arquitectónica de un sistema. Esta representación sirve como el marco de trabajo a partir del cual se conducen actividades de diseño más detalladas. Un conjunto de patrones arquitectónicos permite que un ingeniero de software reutilice conceptos en el nivel de diseño. El diseño arquitectónico puede representarse al usar uno o más de muchos modelos diferentes.

Patrones. Brad Appleton define un *patrón de diseño* de la siguiente manera: "Un patrón es una semilla de conocimiento, la cual tiene un nombre y transporta la esencia de una solución probada a un problema recurrente dentro de cierto contexto en medio de intereses en competencia" Dicho de otro modo, un patrón de diseño describe una estructura de diseño que resuelve un problema de diseño particular dentro de un contexto específico y en medio de "fuerzas" que pueden tener un impacto en la manera en que se aplica y utiliza el patrón. La finalidad de cada patrón de diseño es proporcionar una descripción que le permita al diseñador determinar 1) si el patrón es aplicable al trabajo actual, 2) si el patrón se puede reutilizar (por ende, ahorrar tiempo de diseño), y 3) si el patrón puede servir como guía para desarrollar un patrón similar, pero diferente en cuanto a la funcionalidad o estructura.

Modularidad. Los patrones de arquitectura y diseño de software materializan la *modularidad*, es decir, el software se divide en componentes con nombres independientes y que es posible abordar en forma individual. Estos componentes llamados *módulos* se integran para satisfacer los requisitos del problema. Se ha establecido que la "modularidad es el atributo particular del software que permite que un programa sea manejable de manera intelectual". El software monolítico (es decir, un programa grande compuesto por un módulo sencillo) no puede entenderlo con facilidad un ingeniero de software. El número de rutas de control, la amplitud de las referencias, el número de variables y la complejidad general imposibilitaría comprenderlo. Esto conduce a una estrategia de "divide y vencerás" (es más fácil resolver un problema complejo cuando este se divide en piezas más manejables). Esto tiene implicaciones importantes con respecto a la modularidad y al software. Es un argumento para la modularidad.

Ocultación de información. El principio de *ocultación de información* sugiere que los módulos "se caracterizan por las decisiones de diseño que (cada uno) oculta a los otros". En otras palabras, los módulos deben especificarse y diseñarse de manera que la información (procedimientos y datos) que está dentro del módulo sea inaccesible para otros módulos que no necesiten esa información. La ocultación implica que se puede conseguir una modularidad efectiva al definir un conjunto de módulos independientes que se comuniquen entre sí y que intercambien solo la información necesaria para lograr la función del software.

Independencia funcional. Se consigue al desarrollar módulos con una función "determinante" y una "aversión" a la interacción excesiva con otros módulos. Dicho de otra manera, se desea diseñar el software de tal manera que cada módulo aborde una subfunción específica de los requisitos y tenga una sola interfaz cuando se observe desde otras partes de la estructura del programa. Es justo preguntarse porque es importante la independencia.

El software con una modularidad efectiva, es decir, con módulos independientes, es más fácil de desarrollar porque la función se puede fraccionar y las interfaces se simplifican (considérese las ramificaciones cuando el desarrollo se realiza en equipo). Los módulos independientes son más fáciles de mantener (y probar) porque se limitan a los efectos secundarios que originan las modificaciones al diseño o al código, se reduce la propagación de errores, y es posible emplear módulos reutilizables. En resumen, la independencia funcional es una clave para el buen diseño, y el diseño es la clave para lograr la calidad del software.

Refinamiento. El refinamiento paso a paso es una estrategia de diseño descendente que propuso inicialmente Niklaus Wirth. El desarrollo de un programa se realiza al refinar de manera sucesiva los niveles de detalle procedimentales. Una jerarquía se desarrolla al descomponer el enunciado macroscópico de una función (una abstracción procedimental) paso a paso hasta alcanzar las oraciones del lenguaje de programación. En realidad, el refinamiento es un proceso de elaboración. Se inicia con el enunciado de una función (o de una descripción de datos) que se define con un alto grado de abstracción. Esto es, el enunciado describe los datos o la función de manera conceptual, pero no proporciona información acerca de los trabajos internos de la función o de la estructura interna de los datos. El refinamiento hace que el diseñador trabaje sobre el enunciado original y proporcione más y más detalles conforme se realiza cada refinamiento sucesivo (elaboración). La abstracción y refinamiento son conceptos complementarios. La abstracción permite especificar procedimientos y datos sin considerar detalles de menor grado. El refinamiento ayuda a revelar los detalles de grado menor.

Refabricación. Técnica de reorganización que simplifica el diseño (o código) de un componente sin cambiar su función o comportamiento. Se define de la siguiente manera: “La refabricación es el proceso de cambiar un sistema de software de tal forma que no se altere el comportamiento externo de su código [diseño] y aun así se mejore su estructura interna.” Cuando un software se refabrica el diseño existente se examina en busca de redundancias, elementos de diseño inútiles, algoritmos innecesarios o insuficientes, estructuras de datos inapropiadas o construidas de manera incorrecta, u otra falla de diseño que pueda corregirse para lograr un mejor diseño.

Clases de Diseño. Conforme evoluciona el modelo de diseño, el equipo de software debe definir un conjunto de *clases de diseño* que 1) refine las clases de análisis al proporcionar detalles del diseño que permitirán la implementación de las clases, y 2) produzca un conjunto nuevo de clases de diseño que implementen una infraestructura de software para soportar la solución del negocio. Se sugieren cinco diferentes tipos de clases de diseño, cada uno representa una capa distinta de la arquitectura de diseño:

- Las *clases de interfaz con el usuario* definen todas las abstracciones necesarias para IHC⁷. En muchos casos, la IHC ocurre dentro del contexto de una metáfora (ejemplo: un libro, un formato, un fax) y las clases de diseño para la interfaz pueden ser representaciones visuales.
- Las *clases del dominio de negocios* a menudo son refinamientos de las clases de análisis definidas antes. Las clases identifican los atributos y servicios (métodos) necesarios para implementar algún elemento del dominio de negocios.
- Las *clases de proceso* implementan abstracciones del negocio en un nivel más bajo, las cuales se requieren para manejar por completo las clases del dominio de negocios.
- Las *clases persistentes* representan almacenamientos de datos (por ejemplo, una base de datos) que persistirán más allá de la ejecución del software.
- Las *clases de sistema* implementarán funciones de gestión y control del software que permitan al sistema operar y comunicarse dentro de su entorno y el con el mundo exterior.

⁷La interacción humano-computadora

3.2.4. EL MODELO DE DISEÑO.

El *modelo de diseño* puede verse en dos dimensiones diferentes. La dimensión del *proceso* indica la evolución del modelo de diseño conforme se ejecutan las tareas de diseño como una parte del proceso del software. La dimensión de *abstracción* representa el grado de detalle a medida que cada elemento del modelo de análisis se transforma en una equivalente del diseño y después se refina de una manera iterativa.

Los elementos del modelo de diseño utilizan muchos de los diagramas en **UML**⁸ aplicados en el modelo del análisis. La diferencia es que estos remates están refinados y elaborados como parte del diseño; se proporciona un mayor detalle para la implementación específica y se resaltan la estructura y el estilo arquitectónicos, los componentes que residen dentro de la arquitectura y las interfaces entre los componentes y con el mundo exterior.

3.2.4.1 Elementos del diseño de datos.

Al igual que otras actividades de la ingeniería de software, el *diseño de datos* (algunas veces llamado *arquitectura de datos*) crea un modelo de datos o información que se representa con un alto grado de abstracción (la visión de los datos del cliente/usuario). Después, este modelo de datos se refina en representaciones que de manera progresiva tiene una implementación específica y que pueden procesarse mediante el sistema basado en computadora. En muchas aplicaciones de software la arquitectura de los datos tendrá una profunda influencia sobre la arquitectura del software que los debe procesar. La estructura de los datos siempre ha sido una parte importante del diseño del software. A nivel de los componentes del sistema, las estructuras del diseño de datos y los algoritmos con que se manipulan son esenciales para la creación de aplicaciones de alta calidad. A nivel de aplicación, la traducción de un modelo de datos (obtenido como una base de la ingeniería de requisitos) a una base de datos es esencial para alcanzar los objetivos de negocio de un sistema. Al nivel de negocios, la colección de información almacenada en bases de datos dispersas y reorganizadas en una “colección de datos” permite la explotación de datos o el descubrimiento de un conocimiento que puede tener un impacto sobre el éxito del mismo negocio.

3.2.4.2 Elementos del diseño arquitectónico.

El *diseño arquitectónico* para el software es el equivalente al plano de planta de una casa. Este plano muestra la configuración general de las habitaciones, tamaño, forma, las puertas y ventanas, etc. El plano de planta proporciona una visión global de la casa. Los elementos del diseño arquitectónico dan una visión general del software. El modelo arquitectónico se obtiene a partir de tres fuentes: 1) la información acerca del dominio de aplicación para el software que se va a construir; 2) los elementos del modelo de análisis específico, tales como los diagramas de flujo de datos o clases de análisis, sus relaciones y colaboraciones para el problema que se tiene a la mano, y 3) la disponibilidad de patrones y estilos arquitectónicos.

3.2.4.3 Elementos del diseño de interfaz.

El *diseño de interfaz* para el software es el equivalente a un conjunto de dibujos detallados para puertas, ventanas y utilidades de una casa. Los elementos del diseño de interfaz para software muestran como fluye la información hacia o fuera del sistema y como éste está comunicado entre los componentes definidos como parte de la arquitectura. Existen tres elementos importantes del diseño de interfaz: 1) la interfaz con el usuario; 2) interfaces externas a otros

⁸ Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language)

sistemas, redes u otros consumidores de información; 3) interfaces internas entre varios componentes de diseño. Estos elementos de diseño de interfaz permitan al software comunicarse de manera externa y permiten la comunicación y colaboración interna entre los componentes que pueblan la arquitectura del software.

En algunos casos, una interfaz se modela de una manera muy parecida a una clase. El UML define una interfaz de la siguiente manera: “Una interfaz es un determinante de las operaciones [públicas] visibles de manera externa de una clase, componente u otro clasificador (incluidos los subsistemas) sin especificación de estructura interna”. Dicho de un modo más simple, una interfaz es un conjunto de operaciones que describe parte del comportamiento de una clase y proporciona acceso a esas operaciones.

4.4 Elementos de diseño a nivel de componentes.

El diseño al nivel de componentes para el software equivale a un conjunto de dibujos detallados (y especificaciones) para cada cuarto en una casa. Estos dibujos muestran el alambrado y la instalación sanitaria dentro de cada cuarto, la ubicación de los receptáculos eléctricos e interruptores, llaves, lavabos, tinas, desagües y armarios. También describen los pisos que se usarán, los moldes que se aplicarán, y cualquier otro detalle asociado con el cuarto. El diseño a nivel de componentes para software describe por completo el detalle interno de cada componente del software. Para lograrlo el diseño al nivel de componentes define estructuras de datos para todos los objetos de datos locales, así como detalle algorítmico para todo el procesamiento que ocurre dentro de un componente y una interfaz que permite el acceso a todas las operaciones de los componentes (comportamientos).

4.5. Elementos del diseño a nivel del despliegue.

Los elementos de diseño al nivel del despliegue indican como se ubicarán la funcionalidad y los subsistemas dentro del entorno computacional físico que soportará al software. Por ejemplo, los elementos del producto *HogarSeguro* están configurados para operar dentro de tres entornos de computación primarios: una PC doméstica, el panel de control de *HogarSeguro* y un servidor ubicado en la empresa. (lo que proporciona acceso al sistema a través de Internet. Durante el diseño se desarrolla un diagrama de despliegue en UML y después se refina.

3.2.5. DISEÑO DE SOFTWARE BASADO EN PATRONES.

Los mejores diseñadores en cualquier campo de trabajo tiene la misteriosa habilidad de vislumbrar patrones que caracterizan un problema y los patrones correspondientes que pueden combinarse para crear una solución. A través del proceso de diseño, un ingeniero de software debe buscar toda la oportunidad de reutilizar patrones de diseño existentes (cuando cumplen las necesidades de un diseño) en vez de crear nuevos.

Descripción de un patrón de diseño. Las disciplinas maduras de la ingeniería utilizan miles de patrones de diseño. Por ejemplo, un ingeniero mecánico utiliza un eje de dos pasos como patrón de diseño clave. Los atributos (diámetro del eje, dimensiones del orificio de las llaves, etc) y las operaciones (ejemplo, la rotación del giro y la conexión del giro) son inherentes al patrón. Un ingeniero eléctrico utiliza un circuito integrado (un patrón de diseño en extremo complejo) para resolver un elemento específico de un problema nuevo. Los patrones de diseño pueden describirse como “Plantilla del patrón de diseño”.

Una descripción del patrón del diseño puede considerar también un conjunto de fuerzas de diseño. Las *fuerzas de diseño* describen requisitos no funcionales (por ejemplo, facilidad de

mantenimiento, portabilidad) asociados con el software en el que se aplicará el patrón. Además, las fuerzas definen las limitaciones que restringen la manera en que se implementará el diseño. En esencia, las fuerzas de diseño describen el ambiente y las condiciones que deben existir para que el patrón del diseño sea aplicable. Las características del patrón (clases, responsabilidades y colaboraciones) indican los atributos ajustables del diseño para permitir que el patrón se ajuste a una variedad de problemas. Estos atributos representan características del diseño que pueden buscarse (por ejemplo, a través de una base de datos) para que sea factible encontrar un patrón apropiado. Por último, la guía asociada con el uso de un patrón de diseño indica las ramificaciones de las decisiones del diseño.

Los nombres de los patrones de diseño deben elegirse con cuidado. Uno de los problemas técnicos clave en la reutilización de software es la falta de habilidad para encontrar patrones reutilizables existentes, a pesar de que existen cientos o miles de patrones. La búsqueda del patrón “correcto” tiene un apoyo inmenso si se cuenta con un nombre significativo del patrón.

Utilización de patrones en el diseño. Los patrones de diseño pueden usarse durante el diseño del software. Una vez que se ha desarrollado el modelo de análisis, el diseñador puede examinar una representación detallada del problema que debe resolver y las restricciones que impone el problema. La descripción del problema se examina en varios grados de abstracción para determinar si es flexible para uno o más de los siguientes tipos de patrones de diseño.

Patrones Arquitectónicos. Estos patrones definen la estructura general del software, indican las relaciones entre los subsistemas y los componentes del software, y definen las reglas para especificar las relaciones entre los elementos (clases, paquetes, componentes, subsistemas) de la arquitectura.

Patrones de Diseño. Se aplican a un elemento específico del diseño, relaciones entre los componentes o los mecanismos para efectuar la comunicación de componente a componente.

Idiomas. A veces llamados *patrones de código*, estos patrones específicos de lenguaje por lo general implementan un elemento algorítmico o un componente, un protocolo de interfaz específico o un mecanismo de comunicación entre los componentes.

Cada uno de los tipos de patrones difiere en el grado de abstracción en el que proporciona una guía directa para la actividad de construcción (codificación) del proceso de software.

Marcos de Trabajo. En algunos casos es necesario proporcionar una infraestructura esquelética específica de implementación, llamado *marco de trabajo*, para el trabajo de diseño. Esto es, el diseñador puede seleccionar una “*miniarquitectura reutilizables* que ofrezca el comportamiento y la estructura genérica para una familia de abstracciones del software, junto con un contexto... que especifique su colaboración y uso dentro de un dominio dado”

Un marco de trabajo no es patrón arquitectónico, sino un esqueleto con una colección de “puntos de conexión” (también llamados ganchos y ranuras) que le permiten adaptarse aun dominio de un problema específico. Los puntos de conexión permiten al diseñador integrar clases o funcionalidad específica del problema dentro del esqueleto. En un contexto orientado al objeto, un marco de trabajo es una colección de clases que cooperan.

En esencia, el diseñador de un marco de trabajo argumentará que una miniarquitectura reutilizable se puede aplicar a todo el software que se desarrollará dentro de un dominio limitado de aplicación. Para que sean más efectivos, los marcos de trabajo se aplican sin cambios. Se pueden agregar elementos de diseño adicionales, pero solo a través de los puntos de conexión que permiten que el diseñador desarrolle el esqueleto del marco de trabajo

3.3 CONSTRUCCIÓN

La construcción del sistema inicia cuando se ha determinado qué es lo que se tiene que hacer y cómo se le va hacer. Es entonces cuando un equipo de programadores coordinado por un líder de proyecto se reúne para ponerse de acuerdo en la manera que han de ejecutarse las instrucciones documentadas en las fases anteriores.

Uno de los factores más importantes a considerarse en esta fase es en cuanto a lo relativo a los tiempos de entrega de cada uno de los productos. Una parte crucial es cuando se elabora el plan de entrega de cada uno de los productos de acuerdo a su tiempo de construcción.

De aquí se depende en mucho del talento del líder de proyecto, ya que deberá maximizar este plan, ya que por una parte se deberá considerar la mayor holgura para que cada uno de los programadores tenga el tiempo suficiente para entregar las productos con una gran calidad, así como que tengan un “colchón de tiempo” para posibles eventualidades surgidas durante la construcción de los productos; y por otro lado tendrá que ser el menor tiempo posible para satisfacer los requerimientos cada vez mas exigentes de los clientes.

Así entonces, el tiempo y la calidad son las variables más importantes a considerar dentro de la construcción del software, mismas que determinan la manera en que deberá planearse y estructurarse esta fase en particular.

En cuanto a su proceso operacional, la fase de construcción puede resumirse de la siguiente manera:

- Desarrollo recibe de los analistas de sistemas crean carpetas de trabajo donde se documentan la hojas de especificación, que contienen el detalle de cada unos de los productos.
- Desarrollo recibe del equipo de diseño el modelo o prototipo del backbone⁹ o espinazo de la aplicación.
- Se seleccionan las herramientas a utilizar.
- Se elaboran los estándares de programación.
- Se ensambla la base de datos.
- El líder de proyecto define los tiempos de entrega de los productos.
- El líder de proyecto define la dificultad de cada uno de los productos.
- Se establece el plan de trabajo tomando en cuenta la experiencia y pericia de cada uno de los programadores que integran el grupo de desarrollo.
- Se realizan pruebas unitarias de cada uno de los productos.
- Se realizan pruebas integrales del proceso del sistema.
- Se libera la aplicación.

⁹ Columna Vertebral

3.3.1 ELECCIÓN DE HERRAMIENTAS.

En la actualidad existen una infinidad de herramientas de programación. El mercado ofrece múltiples productos con características diferentes, mismas que el líder de proyecto debe conocer para que su elección sea la más adecuada al proyecto que va a construirse.

Dentro de una arquitectura cliente-servidor el precio, la robustez, la confiabilidad, el control, ó la manejabilidad por mencionar algunas de las variables mas importantes a considerar para hacer la mejor elección de herramientas.

3.3.1.1 Elección del Lenguaje de programación.

Es una importante decisión pues dicha elección tendrá resultados en el tiempo y tipo de desarrollo. El resultado definitivo dependerá del lenguaje utilizado. Según los requerimientos existen lenguajes más o menos convenientes. Existen cerca de 250 lenguajes comerciales en uso. Los programas pueden ser entre otros:

- De procedimiento. Conducen al programador durante cada instrucción paso a paso. Ejemplos: BASIC, C, C++.
- De no procedimiento. Se deja al lenguaje el comportamiento durante las instrucciones. Ejemplos, SQL¹⁰, dBase, etc.
- Lenguajes imperativos. El lenguaje dirige a la computadora en una secuencia exacta de instrucciones. La mayoría de los programas de alto nivel son imperativos. Ejemplos, Fortran, Pascal, C, C++.
- Lenguajes orientados a objetos. Estos lenguajes definen objetos que tienen una clase, y por tanto pueden heredar características a otros objetos y posibilitan el polimorfismo en dichas clases. Ejemplos C++ y JAVA.
- Lenguajes visuales. El desarrollo de la interfaz gráfica ha permitido la aparición de lenguajes visuales. Ejemplos son Visual Basic, Visual C++.
- De cuarta generación. Casi todos son de no procedimiento y se distinguen por la facilidad en su aprendizaje. Ejemplo: SQL.
- Lenguajes de quinta generación. Aun no aparecen, sin embargo se supone que serán los programas que permitirán comunicarnos con las computadoras de manera verbal y visual.

Cualquiera que sea la elección, deberá considerarse siempre la premisa básica de que una interfaz grafica del usuario deberá proveer un ambiente amigable al usuario final, además deberá ofrecer la posibilidad de utilizar los recursos de internet para la interconectividad con otros sistemas.

3.3.1.2 Elección del Manejador de base de datos.

Es preciso conocer hacer muy de fondo las diferentes plataformas o manejadores de bases de datos para poder optar por la más adecuada para ser implantada. Se recomienda principalmente conocer las ventajas y desventajas de los cuatro grandes manejadores como lo son SQL Server de Microsoft, Oracle, Informix, y DB2 de IBM, ya que todos los demas manejadores o son muy pequeños (Access) o son muy antiguos (Clipper) o no ofrecen la seguridad adecuada (My SQL).

¹⁰ Structure Query Language – Lenguaje Estructurado de Consulta

SQL Server de Microsoft.

Microsoft SQL Server 7.0 constituye un lanzamiento determinante para los productos de bases de datos de Microsoft, continuando con la base sólida establecida por SQL Server 6.5. Como la mejor base de datos para Windows NT, SQL Server es el RDBMS¹¹ de elección para una amplia gama de clientes corporativos e ISV's¹² que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento.

SQL Server 7.0 disminuye el costo total de propiedad a través de características como administración multi-servidor y con una sola consola; ejecución y alerta de trabajos basadas en eventos y seguridad integrada. Esta versión también libera al administrador de base de datos para aspectos más sofisticados del trabajo al automatizar las tareas de rutina. Al combinar estos poderosos servicios de administración con las nuevas características de configuración automática, Microsoft SQL Server 7.0 es la elección ideal de automatización de sucursales y aplicaciones de base de datos insertadas.

Oracle.

Es manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Es el conjunto de datos que proporciona la capacidad de almacenar y acude a estos de forma consecuente con un modelo definido como relacional. Además es una suite de productos que ofrece una gran variedad de herramientas.

Es el mayor y más usado RDBMS en el mundo. La Corporación Oracle ofrece este RDBMS como un producto incorporado a la línea de producción. Además incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios. Oracle corre en computadoras personales, microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitectura de hardware y software distintos sin tener la necesidad de cambiar una sola línea de código. Esto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

Informix

Oracle siempre ha sido considerada una base de datos para uso más general que Informix. Informix por su lado, se especializó más en aplicaciones tipo GIS¹³, Datawarehouse¹⁴ y Datamining¹⁵. Sin duda a los gurús, les agrada más Informix que Oracle. En cuanto a precios, Informix es más caro que Oracle en configuraciones parecidas.

En la práctica, como buenos enemigos acérrimos, ambas tienen parecidas características y funcionalidades. Cada una tiene las típicas ampliaciones que permiten especializar la base de datos a un cierto tipo de aplicación. Para competir, en cada nueva versión que sacaban, las dos iban añadiendo de serie muchas de las extensiones que en la versión anterior eran opcionales (de pago por separado). En el momento de escribir este artículo las dos están en la versión 9.X. De cara a una instalación y configuración básicas, las diferencias frente a Oracle que el lector va a notar son dos:

¹¹ RDBMS: (Relational Database Management System ó Sistema de Gestión de Base de Datos Relacional)

¹² ISV's: Fabricantes de Software Independientes

¹³ GIS: Sistema de Información Geográfica

¹⁴ Un almacén de datos (del inglés data warehouse) es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.)

¹⁵ Data Mining ó Minería de Datos, la extracción de información oculta y predecible de grandes bases de datos.

- En Oracle, tienes que definir los usuarios dentro la base de datos (gestión interna de usuarios). En cambio, Informix utiliza los mismos usuarios de Linux (los que creas con adduser), simplificando la administración.
- Para conectar externamente vía TCP/IP¹⁶ a Oracle, hacía falta colgar un *daemon*¹⁷ llamado *listener* de un puerto. Luego el *listener* traducía las llamadas al SQLNet para hablar con la base de datos. Luego había dos procesos: el *daemon* de la base de datos (Oracle), y el *daemon* que escuchaba el puerto (*listener*). En Informix, el mismo *daemon* de la base Oninit atiende los puertos. Informix ocupa menos memoria y recursos.

Con estas diferencias podemos ver que Informix es una base de datos más moderna e integrada con Linux que Oracle. Pero los precios son los precios, y marketing es marketing.

DB2

Los productos Data Management¹⁸ proveen las habilidades de administración de datos necesarias para administrar, almacenar y acceder a los datos operacionales y no estructurados necesarios de la Web. Estos productos proporcionan soporte a las actividades de reunir, administrar, analizar y compartir informaciones para fines estratégicos. Las necesidades de informaciones de una compañía evolucionan al mismo paso que e-business¹⁹. Las transacciones de e-business necesitan una base de información cada vez más heterogénea para sustentar aplicativos que diferencian los procesos de negocio. Las transacciones aún deben ser rápidas, seguras y estar siempre disponibles. Sin embargo, las transacciones de e-business deben agregar valor a las informaciones con análisis incorporado para dar soporte a la toma de decisiones en tiempo real.

Las transacciones de e-business también necesitan ser "completas en contenido" y administrar formularios de información, como datos XML²⁰, imágenes, streaming²¹ video, y otros formatos, que van más allá de simples líneas y columnas de información alfa-numérica. Estos nuevos tipos de información y de aplicativos impulsan demandas de nuevos servicios, como la administración de derechos digitales. Teniendo el total de sus activos de informaciones digitalizados, las compañías pueden mejorar el servicio a clientes, operar con mayor eficiencia, y hacerse más competitivas. A lo largo de los años, IBM Data Management invirtió en los mercados de bases de datos, inteligencia empresarial, integración de información y herramientas e-business, lo que coloca a IBM en posición única para la provisión de infraestructura de información integrada que puede proporcionar soporte a las necesidades del modelo de transacciones de e-business que surge.

- Capacidad de administración DB2 proporciona capacidades significativas de automatización como auto-configuración, auto-optimización y auto-administración.
- Nuevos niveles de información integrada DB2 ayuda a resolver problemas críticos a través de la integración de informaciones, potenciando los Servicios de Web y XML.
- Base robusta de e-business.

¹⁶ Conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

¹⁷ Daemon (Disk and Execution Monitor, Monitor de ejecución y disco). Un daemon es un programa que se ejecuta en segundo plano y se activa automáticamente cuando se requiere. Por ejemplo, el daemon HTTP (httpd) responde a las peticiones HTTP.

¹⁸ Manejo de Datos

¹⁹ Comercio Electrónico

²⁰ XML: XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos

²¹ Streaming es un término que se refiere a ver u oír un archivo directamente en una página web sin necesidad de descargarlo antes al ordenador

3.3.1.3 Elección de utilerías

Adicionalmente seleccionaremos lo que conocemos como utilerías, las cuales son pequeños programas que nos ayudan a hacer más rápida la construcción de los productos:

- Grids: Rejillas para mostrar los datos en los formularios, como las marcas DataFlex ó Spread.
- Reporteadores: Ayudan a emitir información al usuario mas rápidamente, como las marcas Cristal Reports o ActiveReports
- Iconos
- Controles, como calendarios, manejo de imágenes, formatos de números, etc.

3.3.1.4 Sistema Operativo del Servidor.

Existen básicamente dos grandes grupos bien diferenciados, los servidores basados en Windows (NT, 2000, 2003), y los basados en Unix²² (Linux²³, FreeBSD²⁴, Solaris²⁵, AIX²⁶, etc.). La elección depende de las necesidades específicas del webmaster, y condicionará el crecimiento del sitio, por sus diferentes características y funciones.

Hay un temor a contratar servidores Unix por parte de los webmasters con conocimientos de Unix muy escasos o nulos, al desconocer con qué han de enfrentarse. Gracias a los paneles de control que incorporan todos los planes de alojamiento, el sistema bajo el que trabaje el servidor es prácticamente transparente en su uso para la mayoría de aplicaciones. Sólo en un uso avanzado puede necesitarse hacer uso del shell²⁷ del sistema, si es que su plan de alojamiento lo permite.

Al elegir el sistema operativo básicamente consideramos dos cosas:

- a) Seguridad. No debemos olvidar que la información será una parte importante del patrimonio de la agencia, por lo cual nunca estará de más cualquier medida o dispositivo que ayude a proteger tanto la integridad de la información, como el que nuestra información no caiga en manos ajenas
- b) Operatividad. Debemos pensar sobre las personas que manejaran el sistema en la agencia, ya que por lo regular nos hemos encontrado que se tiene escasos conocimientos cuando el sistema operativo es diferente de Windows.

²² Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T.

²³ Linux es un sistema operativo tipo Unix (también conocido como GNU/Linux) que se distribuye bajo la Licencia Pública General de GNU (GNU GPL), es decir que es software libre. Su nombre proviene del Núcleo de Linux, desarrollado en 1991 por Linus Torvalds. Es usado ampliamente en servidores y super-computadores,[1] y cuenta con el respaldo de corporaciones como Dell, Hewlett-Packard, IBM, Novell, Oracle, Red Hat y Sun Microsystems.

²⁴ FreeBSD es un sistema operativo multiusuario, capaz de efectuar multitarea con apropiación y multiproceso en plataformas compatibles con múltiples procesadores; el funcionamiento de FreeBSD está inspirado, como ya se dijo, en la variante 4.4 BSD-Lite de UNIX.

²⁵ Solaris es un sistema operativo desarrollado por Sun Microsystems. Es un sistema certificado como una versión de UNIX.

²⁶ AIX (Advanced Interactive eXecutive) es un sistema operativo UNIX System V propietario de IBM.

²⁷ Shell: línea de comandos.

3.3.2 ESTÁNDARES DE PROGRAMACIÓN.

Un aspecto muy importante para un programador es definir el "estilo" de programación que este utiliza. Algunos, los más principiantes, usan nombres de sus seres queridos para nombrar objetos y variables en el programa, otros, que no desea pensar mucho, usan nombres aleatorios para sus variables de código, y así podemos seguir con una interminable lista de "estilos" o "formas" de programación.

La pregunta clave que todos nos hacemos es: ¿Cual es el estilo adecuado?, ¿que terminología es la mas apropiada para mi?

A decir verdad, no existe "terminología" o "estilo" que sea mejor que otro. La valoración de dichas "terminologías" se basa no en lo que al programador le guste, sino primordialmente en el uso adecuado de un "terminología" específica. Esto es lo que denominamos "estándares de programación", que no es mas que el usar y seguir ciertas reglas de notación y nomenclatura durante la fase de implementación (codificación) de una aplicación.

Criterios de un buen estándar.

Hay muchos estándares de programación que podemos usar. Debemos elegir aquel que se adecue más a nuestro estilo de programación. Si aun no se tiene uno, pues en este artículo veremos brevemente algunos de ellos. Un buen estándar de programación generalmente considerará los siguientes factores:

- Factor mnemotécnico: Para que el programador pueda recordar el nombre de una variable fácilmente
- Factor sugestivo: Para que otros programadores puedan leer y entender rápidamente nuestro código
- Consistencia: Tiene que ver con usar las mismas convenciones de nomenclatura en todo el programa y hacer que el texto del código sea "legible"

Ventajas del uso de estándares.

Establecer un estándar de programación y nomenclatura puede tomar mucho tiempo. De allí la necesidad de "encontrar" o "elaborar" aquel que sea ajuste más a nosotros (según nuestra experiencia). Pero una vez establecido este estándar, los beneficios son muchos:

- Los nombres de variables serán mnemotécnicos con lo que se podrá saber el tipo de dato de cada variable con sólo ver el nombre de la variable
- Los nombres de variables serán sugestivos, de tal forma que se podrá saber el uso y finalidad de dicha variable o función fácilmente con solo ver el nombre de la variable
- La decisión de poner un nombre a una variable o función será mecánica y automática, puesto que seguirá las reglas definidas por nuestro estándar.
- Permite el uso de herramientas automáticas de verificación de nomenclaturas

¿Porque los estándares son usados muy poco?

Si los estándares tienen tantos beneficios, entonces la pregunta es ¿porque los programadores los usan muy pocas veces? La razón tiene que ver más con los seres humanos que con la tecnología:

- Trabajan en un proyecto que no ha adoptado ningún estándar
- No entiende o no pueden recordar el estándar
- No ven el beneficio
- Están muy apurados o cansados
- Prefieren creatividad y consistencia arbitraria
- Piensan que es divertido usar nombres "bonitos" en código
- Son "artistas del software" y no pueden estar regidos por convenciones

Cabe recalcar que el buen programador debe estar en la capacidad de adaptarse a cualquier Standard de programación que establezca el equipo de desarrollo a la que pertenece dicho programador. Esto es común cuando se trabaja en equipos de programadores de 2 o más personas.

¿Qué comprende un estándar?

Un estándar de programación no solo se busca definir la nomenclatura de las variables, objetos, métodos y funciones, sino que también tiene que ver con el orden y legibilidad del código escrito. Siguiendo esta idea, podemos definir 3 partes principales dentro de un estándar de programación:

- Convención de nomenclatura: Como nombrar variables, funciones, métodos, etc.
- Convenciones de legibilidad de código: Como identificar el código, etc.
- Convenciones de documentación: Establecer comentarios, archivos de ayuda, etc.

Estándares más comunes

A continuación daremos algunas pautas sobre los principales estándares de programación:

Notación húngara

Esta convención se basa en definir prefijos para cada tipo de datos y según el ámbito de las variables. También es conocida como notación: REDDICK (por el nombre de su creador). La idea de esta notación es la de dar mayor información al nombre de la variable, método o función definiendo en ella un prefijo que identifique su tipo de dato y ámbito. A continuación un ejemplo:

intEdad: Según la definición vemos que esta variable es de tipo INTEGER y que representa la edad de alguna persona

prStrNombre: En este caso la variable tiene el prefijo: "prInt", lo cual significa que es un parámetro por referencia (pr) de tipo STRING que representa un nombre

gStrConexion: En este caso se trata de una variable global (g) de tipo STRING que representa cierta información de conexión

Notación PascalCasing

Pascal-Casing es como la notación húngara pero sin prefijos. En este caso, los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula. A continuación un ejemplo:

DoSomething: Este nombre de método esta compuesto por 2 palabras, ambas iniciando con letra mayúscula.

Notación camelCasing

Camel-Casing es común en Java. Es parecido al Pascal-Casing con la excepción que la letra inicial del identificador no debe estar en mayúscula. A continuación un ejemplo:

doSomething: Este nombre de método esta compuesto por 2 palabras, la primera todo en minúsculas y la segunda iniciando con letra mayúscula.

Crear tu propio estándar

También, como se dijo anteriormente, podemos establecer nuestros propios estándares de programación, los cuales pueden basarse en los ya existente, extenderlos o modificarlos. A continuación les presentare un ejemplo de un estándar de programación personalizado

	AMBITO		
	Local o a nivel de procedimiento	Nivel de modulo o form	Global
Controles			
Classes	cls< ClassName>	cls< ClassName>	cls< LassName>
Módulos	mod< ModuleName>	mod< ModuleName>	mod< ModuleName>
Formularios	frm< FormName>	IFrm< FormName>	gFrm< FormName>
Combobox	cbo< ComboName>	ICbo< ComboName>	gCbo< ComboName>
Command	cmd< commandName>	ICmd< CommandName>	gCmd< CommandName>
Datagrid	grd< GridName>	IGrd< GridName>	gGrd< GridName>
Listbox	Lst< ListboxName>	ILst< ListboxName>	gLst< ListboxName>
Option buttons	opt< OptionName>	IOpt< OptionName	gOpt< OptionName>
checkboxes	chk< CheckName>	IChk< CheckName>	gChk< CheckName>
Textboxes	txt< TextName>	ITxt< TextName>	gTxt< TextName>
Tipos primitivos			
Integer	int< Nombre >	lInt< Nombre >	gInt< Nombre >
Long	lng< Nombre >	lLng< Nombre >	GLng< Nombre >
Boolean	bln< Nombre >	lBln< Nombre >	gBln< Nombre >
Object	obj< Nombre >	lObj< Nombre >	Gob..< Nombre >
String	str< Nombre >	lStr< Nombre >	gStr< Nombre >
Double	dbl< Nombre >	lDbL< Nombre >	gDbL< Nombre >
Constantes	C_< NOMBRE>	LC_< NOMBRE >	GC_< NOMBRE >

Legibilidad del código

Debemos poner énfasis especial en la legibilidad del código. Cualquiera que sea el proyecto, los miembros del proyecto pasarán mucho tiempo escribiendo, leyendo y revisando el código fuente. Se calcula que un programador pasa la mitad del tiempo tratando de entender "que es lo que tal o cual bloque de código hace". Podemos hacer este trabajo mucho más fácil siguiendo las convenciones de nomenclatura de los estándares y a la vez haciendo nuestro código legible y bien documentado.

A continuación les daré algunas pautas de cómo puede dar mayor legibilidad a su código fuente:

```
1)
Public Sub Procedure(parameters list)
On Error GoTo errCatch
If bSomeCondition = False Then
DoSomething
Else
For each object in colObjects
If bSomeOtherCondition = false then
DoOtherThings
End If
Next
End If
Exit Sub
errCatch:
MsgBox Err.Description, .....
End Sub
```

```
2)

Public Sub Procedure(parameters list)
  On Error GoTo errCatch
  If bSomeCondition = False Then
    DoSomething
  Else
    For each object in colObjects
      If bSomeOtherCondition = false then
        DoOtherThings
      End If
    Next
  End If
  Exit Sub
errCatch:
  MsgBox Err.Description, .....
End Sub
```

Claramente podemos notar que el Segundo bloque de código es mucho más entendible que el primero.

Documentación del código

Esto tiene que ver con los comentarios explicatorios y aclaratorios que establecemos en nuestro código para futura referencia. Muchas veces podemos olvidar la finalidad de un proceso con complicada algoritmia. Los comentarios nos ayudan a recordar los puntos claves de cada parte de nuestro código (sobre todo cuando ha pasado un tiempo largo de haberlo codificado). Además sirve como que los demás miembros del equipo de desarrollo puedan entender también dichos bloques de código. Podemos establecer una convención de documentación que puede darse de muchas formas. A continuación una forma sencilla y practica de comentar nuestro código.

```
'Created by: Helkyn Coello 'Date of creation: 18/08/2004
'Function description: Descripción de función
'Parameters description: descripción de la finalidad de cada parámetro
Function Process1 (par1 as integer, par2 as integer) as Integer
    code goes here
    ...
    ...
End Function
```

Qué herramientas usar.

Algo también importante a la hora de usar estándares es usar una herramienta de verificación de estos estándares. Para aquellos programadores que no estén muy acostumbrados a uso de estas convenciones, estas herramientas pueden resultar muy útiles, puesto que nos harán aprender "a la mala". Por lo general estas herramientas analizar nuestro archivos de código fuente y verifica que partes de el no cumple con nuestras convenciones establecidas. Lógicamente estas herramientas incluyen motores de verificación de distintos tipos de estándares y esto es configurable según nuestras necesidades. Incluso algunos de estas herramientas nos permiten definir nuestras propias reglas y convenciones, como es el caso de FxCop, un software libre de la comunidad gotDotNet, que ha sido incluido como parte del Visual Studio.NET 2005. FxCop tiene una interfaz muy amigable en la cual podemos elegir que reglas queremos sean verificadas y además podemos crear nuevas reglas con el Introspection Engine²⁸ de FxCop. Otra herramienta interesante es "Standard Master" el cual se adhiere al editor de código de VisualStudio.NET y permite verificar el código mientras este se escribe.

Conclusiones.

Nombres de libre imaginación eran permitidos hace décadas, dado el tamaño limitado de los programas y las herramientas. Pero hoy en día, el programador que persiste en usar variables con nombre triviales o sin un orden establecido traerá muchas dificultades en un equipo de desarrollo. Se puede ahorrar algunos segundos al no usar ningún estándar, pero se perderán mucho tiempo después. La esencia de los estándares de programación en mantener la consistencia del código siguiendo una determinada convención de nombres. Esto le ayudará a escribir, mantener y re-usar código en una forma más eficaz y eficiente. Puede elegir el estándar que más le guste, e incluso puede crear y personalizar su propio estándar. Elija aquel que establezca nombres claros, descriptivos y significativos, y que sean fáciles de recordar. Sea que hablemos de nombre de un negocio, producto, programa o de un bebe recién nacido, un buen nombre siempre es esencial. Para lograr "acostumbrarse" mejor a cualquiera fuere el estándar que eligió, puede además usar herramientas de verificación de estas convenciones que puede detectar automáticamente errores de nomenclatura.

²⁸ Motor de introspección

3.3.2.1 Uso de estándares de programación.

Un estándar de programación es una forma de "normalizar" la programación de forma tal que al trabajar en un proyecto cualquiera de las personas involucradas en el mismo tenga acceso y comprenda el código. Ya vimos la definición y ventajas de los estándares de programación, por lo tanto la fase de Construcción deberá primeramente contemplar que en esta actividad el líder de proyecto cree un documento con reglas específicas y rigurosas sobre la técnica de programación que empleará el grupo de programadores.

Coordinar un equipo de programadores no es un tarea fácil, ya que cada programador tiene técnicas diferentes para realizar un mismo trabajo, por lo que el uso de estándares asegurará la uniformidad tanto exterior como interior, de tal forma que se facilitará tanto el uso como el mantenimiento. Además de Todo esto se ve reflejado en una mayor velocidad en los tiempos de entrega de los productos, es decir, mayor rapidez en la programación y a su vez se facilita el mantenimiento.

- Definir la escritura y organización del código fuente de un programa.
- Definir los datos que debe poner un programador al modificar el propio código fuente.
- Reglas para definir una determinada nomenclatura para la declaración de variables.
- Reglas para declarar las clases.
- Reglas para poner los comentarios.
- Reglas para la apariencia de los formularios.
- Reglas para el modo captura de una pantalla
- Reglas para el modo consulta de una pantalla
- Reglas para la operación de las pantallas.
- Reglas y Parámetros para la emisión de reportes.

3.3.2.2 Uso de funciones preprogramadas.

Imaginemos que durante todo el programa, en todas las pantallas de consulta de datos el programador hace siempre el mismo trabajo: hacer un query²⁹ con los parámetros que le da el usuario; conectarse a la base de datos; crear un recordset³⁰ con los registros, llenar un Grid (rejilla) con los registros resultantes: formatear el Grid y mandar un mensaje de aviso al usuario. Y esto mismo se repite en la consulta de refacciones, en la consulta de autos en inventario, en la consulta de órdenes de servicio por mecánico, etc. Para agilizar el trabajo del programador, al principio de la construcción del sistema se crean las funciones preprogramadas, las cuales son un script que a manera de herramienta el programador va a usar para que se acorten los tiempos de entrega de los productos.

Plantillas.

También, como las pantallas deben tener una apariencia similar, es recomendable al principio crear plantillas para los formularios, de tal manera que el programador ya no pierda el tiempo formateando, y se dedique solo a crear el código para el proceso interno de la pantalla.

Validaciones.

También será recomendable programar funciones para verificar que los datos que se guardan tengan valores apropiados.

²⁹ Del inglés Query ó Consulta, instrucción en SQL para extraer un conjunto de datos de la base

³⁰ Recordset es una estructura de datos usada en programación cuya utilidad es la de almacenar información desde una tabla de una base de datos.

3.2.2.3 Uso de Transacciones.

Una transacción es un conjunto de operaciones que van a ser tratadas como una única unidad. Estas transacciones deben cumplir 4 propiedades fundamentales comúnmente conocidas como ACID³¹ (atomicidad, coherencia, asilamiento y durabilidad). La transacción más simple en SQL Server es una única sentencia SQL. Por ejemplo una sentencia como esta:

```
UPDATE Products SET UnitPrice=20 WHERE ProductName = 'Chai'
```

Esta es una transacción autocommit³². Cuando enviamos esta sentencia al SQL Server se escribe en el fichero de transacciones lo que va a ocurrir y a continuación realiza los cambios necesarios en la base. Si hay algún tipo de problema al hacer esta operación el SQL Server puede leer en el fichero de transacciones lo que se estaba haciendo y si es necesario puede devolver la base de datos al estado en el que se encontraba antes de recibir la sentencia.

Por supuesto este tipo de transacciones no requieren de nuestra intervención puesto que el sistema se encarga de todo. Sin embargo si hay que realizar varias operaciones y queremos que sean tratadas como unidad tenemos que crear esas transacciones de manera explícita.

Sentencias para una transacción.

Como decíamos una transacción es un conjunto de operaciones tratadas como una sola. Este conjunto de operaciones debe marcarse como transacción para que todas las operaciones que la conforman tengan éxito o todas fracasen. La sentencia que se utiliza para indicar el comienzo de una transacción es 'BEGIN TRAN³³'. Si alguna de las operaciones de una transacción falla hay que deshacer la transacción en su totalidad para volver al estado inicial en el que estaba la base de datos antes de empezar. Esto se consigue con la sentencia 'ROLLBACK TRAN³⁴'. Si todas las operaciones de una transacción se completan con éxito hay que marcar el fin de una transacción para que la base de datos vuelva a estar en un estado consistente con la sentencia 'COMMIT TRAN³⁵'.

Aplicación

Imaginemos que cuando facturamos un auto, se grabe la factura pero no se grabe la afectación contable, y tampoco la cuenta por cobrar para el departamento de cartera. Por lo tanto, el líder de proyecto no deberá permitir que su equipo grabe información si que se tenga una estructura transaccionada, tomando en cuenta la siguiente precedencia:

Facturación de Autos

- Cambiar el estatus del pedido a facturado.
- Cambiar el estatus del auto a vendido.
- Insertar la factura correspondiente.
- Insertar la póliza contable correspondiente.
- Insertar la cuenta por cobrar correspondiente.

³¹ ACID a la propiedad de una base de datos para realizar transacciones seguras.

³² Una transacción autocompletada

³³ Especifica que va a empezar una transacción

³⁴ Indica que se ha alcanzado un fallo y que debe restablecer la base al punto de integridad.

³⁵ Le indica al motor que puede considerar la transacción completada con éxito.

3.4 IMPLEMENTACIÓN

3.3.1 Metodología de Implementación

En función de cumplir con las necesidades cada vez más exigentes del mercado de tecnología de información, en la empresa se diseñó una metodología para proveer al equipo de consultoría con un marco de trabajo que les permita aumentar las posibilidades de éxito en la implementación del Software.

La metodología establece actividades, funciones, roles, tiempos, costos, estándares y guías que facilitan la planeación, desarrollo, administración, y cierre de un proyecto de implantación teniendo como base, la línea de acción establecida por los objetivos y expectativas del cliente. De igual forma integra actividades de diferentes áreas de especialización que permiten generar un plan de implantación multifuncional que de como resultado actividades de valor tangible y medible para el cliente. Algunos de los beneficios de utilizar la metodología diseñada son:

- Aprovechamiento eficaz de tiempos
- Incremento de la productividad
- Comunicación efectiva
- Mitigación de riesgos
- Mejor control de costos y tiempos
- Aseguramiento de la calidad

Los encargados de llevar a cabo la Implementación en las agencias es un equipo conformado por profesionales. Dado que la implementación es una actividad exigente que demanda buenos profesionales para hacer frente a los continuos desafíos que el cliente exige, el personal deberá contar con las siguientes características para llevar a cabo con éxito las tareas que les han encomendado:

- **Método:** O lo que es lo mismo, una forma de hacer las cosas. Es quizás la pieza más importante. Muchas veces el cliente sabe lo que quiere pero no sabe cómo conseguirlo; le falta método. ¿Necesita de alguien? ¿De un consultor? que le oriente y le ayude a poner en orden todas sus ideas, y eso lo aporta la metodología. Son las distintas fases o pasos a seguir hasta dar con una solución satisfactoria para el cliente. El método permite llegar a la esencia del problema, desgranarlo, descomponerlo en varias partes para hacerlo más sencillo y darle solución.
- **Flexibilidad:** El consultor se enfrenta a dificultades nuevas las cuales resuelve de inmediato. El consultor es tolerante a la ambigüedad y la incertidumbre, o dicho de otro modo cuenta con la facilidad para adaptarse a cada situación.
- **Capacidad de Trabajo en equipo:** El equipo de consultores trabaja por objetivos y a plazos. No es difícil tener ideas, pero si lo es llevarlas a cabo, sentarse e implementarlas. El equipo de consultoría tiene la valentía y el arrojo para implementarlas.
- **Capacidad de Escucha y Entendimiento:** La mayor parte de las veces no es necesario hacer grandes descubrimientos, sino prestar un poco más de atención a lo que el cliente nos cuenta. Esta capacidad la desempeña de manera amplia el equipo de consultores. En ocasiones no se trata más que de llevar a la operación dentro del sistema los requerimientos del cliente.

- **Pasión por el Trabajo Bien Hecho.** En una palabra, profesionalismo; no sólo en lo que al fondo se refiere, sino también a la forma. La calidad de trabajo es algo que siempre obsesiona al consultor.
- **Actitud Positiva:** La consultoría es un desafío continuo. Cada proyecto es un reto diferente. Uno nunca sabe muy bien con lo que se va a encontrar, pero de lo que sí estamos seguros es que el equipo de consultoría va a ser capaz de darle salida y resolver la situación. Con actitud de fondo positiva.
- **Creatividad:** Que no es más que la capacidad de relacionar conceptos, la habilidad para ensamblar ideas de distintos ámbitos y aprovecharlas de la mejor manera posible.
- **Formación Continua:** El consultor nunca deja de adquirir nuevos conocimientos, los cuales le ayudan a resolver de la mejor manera las exigencias del cliente.
- **Visión Práctica:** El equipo de consultoría cuenta con personal altamente resolutivo. Por lo general una solución medianamente buena pero inmediata, vale más que una solución perfecta tiempo después.

La estructura de la metodología permite tomar el manejo del proyecto desde diferentes enfoques o perspectivas de implantación ya que lo divide en fases, actividades por áreas de especialización y tareas que hacen de la implementación un proceso entendible, flexible y controlable. El control y medición de se lleva mediante la producción de “*entregables únicos*” resultantes de actividades de valor agregado que tienen como objetivo final, la eficaz puesta a punto del sistema Software en el tiempo, forma y costo definidos desde el inicio del proyecto.

3.3.2 Equipos de Implementación

La metodología contempla la separación de la implementación por fases secuenciales que en conjunto conforman el ciclo de vida del proyecto. La transición entre las fases se realiza mediante la entrega, revisión y validación de los *entregables definidos* para cada una de ellas; es importante señalar que de acuerdo a los riesgos identificados en cada proyecto, algunas actividades de fases subsecuentes pueden ser iniciadas antes de la entrega y validación de entregables de las fases anteriores. De igual forma, para determinar el cierre de cada fase, se deben comunicar las tareas concluidas y pendientes, así como los siguientes pasos.

Dentro de cada una de las fases se tienen definidas actividades, responsables, duración y entregables, mismos que deben ser comunicados al cliente con el fin de trabajar bajo la misma línea. Con respecto a lo anterior, es de vital importancia definir desde el inicio del proyecto los criterios de entrega, aceptación y cambios de entregables con el fin de tener claros los objetivos y resultados que el cliente debe perseguir y esperar por parte del equipo de implantación.

3.3.3 Áreas de Especialización

La metodología de Implementación Integral está diseñada bajo un esquema de cinco actividades especializadas. Estas actividades agrupan tareas con naturalezas similares por lo que pueden ser tomadas como paquetes independientes de trabajo. En base a las áreas de especialización, se puede dar un seguimiento efectivo a las tareas correspondientes al equipo de implantación, ya que se tienen claras las dependencias entre actividades, así como las habilidades requeridas por los responsables para poder ser llevadas a cabo eficazmente. El equipo de profesionales de implementación se divide en 5 grupos que desarrollan las siguientes actividades:

3.3.3.1 Logística.

Administración del Proyecto. Actividades respaldo que corresponden al control administrativo del proyecto; auxilian y dan soporte al seguimiento de las demás actividades a lo largo de toda la implantación, teniendo como fin el cumplimiento del proyecto en tiempo y forma.

Tiene como objetivo definir los alcances y límites del producto (Sistema) y del proyecto (Implantación) de acuerdo con las expectativas del cliente. En esta fase se definen las reglas, procedimientos, actividades, responsables, tiempos y costos a seguir en el resto de la implantación mediante un levantamiento general de información. Una vez definido lo anterior, es necesario comunicar lo a todos los involucrados en el proyecto con el fin de establecer el objetivo común a perseguir. Documentación. (Solicitar los formatos por departamento; Determinar la información a capturar; Elaborar programa de capacitación; Junta de acuerdos). Esta fase es el paso inicial del ciclo del proyecto, es aquí en donde se realizan actividades de dimensionamiento del proyecto de Implantación. En esta fase se tienen los siguientes objetivos:

1. Entender claramente las expectativas y objetivos de los involucrados en el proyecto.
2. Realizar un análisis de los procesos de la empresa a alto nivel, identificando los procesos críticos del negocio.
3. Definir el alcance tanto del producto como del proyecto. Esto deberá ir en base a las expectativas y objetivos de los principales involucrados y al análisis de alto nivel realizado.
4. Establecer los límites del proyecto y del producto.
5. Diseñar un Plan de Proyecto en términos de tiempo, costo, calidad y alcance.
6. Identificar y clasificar los riesgos detectados para el proyecto de implantación.
7. Establecer y comunicar al cliente los roles, responsabilidades, actividades y entregables que le corresponden dentro del flujo del proyecto.

Documentos Generados por la empresa:

- Carta de Proyecto
- Objetivo del Documento
- Alcance del Proyecto
- Alcance del Producto
- Límites del Proyecto
- Matriz de Procesos (Vs. Módulos) Consolidada
- Estrategia de Implantación
- Plan de Transición
- Reporte de Interfaces Identificadas
- Análisis de Riesgos
- Criterios de Aceptación del Proyecto
- Criterios de Aceptación del Producto
- Factores Críticos de Éxito
- Organización Inicial Proyecto
- Inventario Áreas de Oportunidad y Mejora por Proceso
- Matriz de Reportes y Formatos / Desarrollos (Resultado análisis GAP³⁶)
- Dimensionamiento de Infraestructura y Propuesta Tecnológica
- Análisis de Tecnología Adicional Requerida
- Cronograma del Proyecto
- Documentación Situación Actual
- Matriz de Reportes

³⁶ ANÁLISIS GAP. Las organizaciones evalúan la diferencia entre su posición actual y el futuro deseado mediante el análisis gap

Carta de Proyecto

La carta de Proyecto tiene como finalidad describir el alcance que se tomará en cuenta como guía para la implementación del **ERP**³⁷ en las diferentes empresas que hayan determinado. A partir de su firma el cliente será parte integral del Contrato de Adquisición de un Sistema Integral de Información, celebrado entre la Empresa y Cliente, por concepto de Implantación.

Estructura de la carta de Proyecto

- I. Objetivo del documento
- ii. Objetivo del proyecto
- iii. Definición del alcance del producto
- iv. Definición del alcance del proyecto
- V. Descripción del alcance de acuerdo al contrato
- Vi. Alcance procesos especiales
- Vii. Límites del proyecto
- Viii. Mapeo procesos contra módulos
- Ix. Plan de gestión del cambio
- X. Metodología de implantación
- Xi. Requisitos del proyecto
- Xii. Productos entregables del proyecto
- Xiii. Criterios de aceptación del producto
- Xiv. Restricciones del proyecto
- Xv. Organización inicial del proyecto
- Xvi. Riesgos iniciales definidos
- Xvii. Políticas y procedimientos pendientes de definir por parte del cliente
- Xviii. Áreas de oportunidad para la dirección general del cliente
- Xix. Programas paralelos de adecuaciones
- Xx. Formatos legales a modificar
- Xxi. Objetivos de la dirección general del cliente
- Xxii. Planes de expansión y estrategia de incorporación
- Xxiii. Anexos
 - a. Procedimiento de Aceptación de Entregables
 - b. Procedimiento de Control de Cambios a Entregables
 - c. Procedimiento de Control de Cambios al Alcance
 - d. Datos Maestros para el **ERP**
 - e. Asunciones del Proyecto
 - f. Hitos del Cronograma (Requerimientos de la Dirección)
 - g. Mapeo General de Procesos
 - h. Estructura Organizacional de la Organización
 - i. Alcance Definido en el Contrato
 - j. Ubicación de Sucursales
 - k. Directorio de Almacenes y Oficinas
 - l. Número de Registros o Transacciones
 - m. **FODA**³⁸ de dueños de negocio
 - n. Análisis de Resistencia al Cambio
 - o. Estrategia de Gestión del Cambio
 - p. Análisis de Requerimientos de infraestructura tecnológica.

³⁷ Los sistemas de planificación de recursos empresariales (**ERP**) son sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía comprometida en la producción de bienes o servicios.

³⁸ Análisis FODA (en inglés SWOT - Strengths, Weaknesses, Opportunities, Threats) es una metodología de estudio de la situación competitiva de una empresa dentro de su mercado y de las características internas de la misma, a efectos de determinar sus Debilidades, Amenazas, Fortalezas y Oportunidades

3.3.3.2. Parametrización.

El objeto principal de este grupo es coordinar los procesos necesarios para adaptar los parámetros del sistema a las necesidades concretas de gestión de cada cliente. En pocas palabras se trata de subir los catálogos básicos para la aplicación, como los empleados, los modelos de autos, los colores, las operaciones de servicio, etc.

Procedimiento:

Comienza con la entrega por parte de Logística de la copia controlada de la Carta del Proyecto. Finaliza con la revisión final de los datos una vez instalados en el servidor del cliente.

Parametrización:

- Emite un formato, anotando el nombre del cliente, su código, la fecha de emisión, y la versión. Se pondrá en contacto con el cliente para completar este documento, y solicitar la información que sea necesaria para la parametrización. Será responsable del buen manejo de la información así como de estar en permanente comunicación con el cliente para tener a tiempo la Carta del Proyecto completa y correcta. En caso que no tenga respuesta por parte del cliente, deberá contactar a Logística para que de seguimiento y se cumplan los tiempos establecidos con el cliente.
- Cuando el técnico tenga que realizar además la adaptación de formatos de impresión, seguirá los pasos que detalla la instrucción de trabajo "Adaptación formatos de impresión". Emitirá el formato "Adaptación formatos de impresión" para registrar el trabajo realizado. Anotará todas las incidencias que se produzcan en su trabajo, así como las observaciones que crea conveniente en el lugar.
- Enviará el documento "Aceptación plan de cuentas" para que el cliente lo firme y lo devuelva, con el fin de garantizar que el plan de cuentas que se va a utilizar para la parametrización cuenta con la aprobación del cliente. Anotará la fecha de solicitud del fax, y la persona a la que se le ha solicitado.
- Al recibir el fax lo archiva junto con el resto de los documentos de parametrización.
- Comienza labores de parametrización, en el servidor de parametrizaciones. Se apoya en sus conocimientos y documentación técnica que crea necesaria. Anotará en la casilla establecida al efecto la **dirección IP**³⁹ del servidor donde lleva la parametrización.
- Anota en la casilla de cada página las observaciones e incidencias que se den en el proceso de parametrización, así como las especialidades o diferencias que el técnico de la instalación deba conocer.
- Anota la fecha de realización en el formato.
- Cada una de las tareas contenidas en las casillas deben ser verificadas, anotándose en el formato el resultado de la comprobación.
- Se entrega a Logística el formato indicando que ha finalizado.
- Incorpora los datos en el servidor del cliente.
- Realiza las últimas comprobaciones, en caso de que las considere necesarias.
- Graba en una memoria USB⁴⁰ el informe financiero que el técnico debe instalar en el concesionario.
- Devuelve los formatos a Calidad, junto con la memoria USB que contiene el informe financiero para hacerlos llegar al técnico que hará la instalación.
- Abre una carpeta definitiva en soporte físico para el cliente, identificándola adecuadamente con el código del cliente y nombre, y archiva en ella la copia controlada de la Carta del Proyecto y el formato.

³⁹ Una dirección IP es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol)

⁴⁰ USB: Universal Series Bus

3.3.3.3 Plataformado.

El objeto principal de este grupo es coordinar los procesos necesarios para el control de los equipos suministrados por el Cliente, su manipulación, su correcto embalaje y su entrega. Se refiere actividades relacionadas con la infraestructura y diseño tecnológico necesarios para la puesta a punto del sistema.

Procedimiento:

Comienza con la llegada del equipo del Cliente a las oficinas. Finaliza con la entrega del equipo en el domicilio del Cliente, incluidos los componentes suministrados por nuestra empresa. Comprende los productos más usuales suministrados por el cliente los cuáles son equipos servidores para parametrizar, a los que se presta especial atención.

Plataformado:

- Revisa diariamente la llegada de equipos.
- Abre un nuevo formato en el que registra los datos referentes al envío, anexando la orden de entrega del mensajero.
- Realiza una comprobación físico-administrativa del contenido y una inspección visual del estado del envío, cumplimentando los datos de comprobación de entrada.
- Si observa algún defecto o falta alguno de los componentes necesarios para la instalación, el responsable hará la anotación correspondiente en el apartado de Incidencias del formato, lo pondrá en conocimiento del cliente, y hará las gestiones precisas para subsanar el defecto. En caso de que el responsable estime que la anomalía no obliga a paralizar el curso normal de la instalación, continuará con el presente procedimiento. En caso contrario se paraliza la instalación hasta que el defecto quede subsanado. Se etiquetará convenientemente el equipo para reflejar la situación.
- Se emplaza el equipo en el área de Servicios habilitado para ello, de manera que queda dispuesto para la instalación y parametrización.
- Si la mercancía recibida corresponde a un componente en sustitución de uno defectuoso, se consignará en el apartado de Incidencias de Entrada del formato correspondiente. Las devoluciones que procedan serán gestionadas por el responsable, quien adjuntará las órdenes de entrega al correspondiente formato.
- Da comienzo al proceso de "Instalación del Software"
- Una vez terminada la instalación, entrega el formato a Calidad para que lo archive en la carpeta correspondiente.
- Avisa a parametrización que el servidor está listo e indica la dirección IP.
- Una vez que ha terminado el proceso de instalación, parametrización e incluida la prueba de funcionamiento final, efectúa una última inspección visual del estado del servidor asegurando que esté dispuesta la documentación para la entrega, reflejando el resultado en el formato.
- Se empaqueta el equipo en su embalaje original, adjuntando los paquetes de software, más los routers⁴¹, asegurándose de que se embalan todos los componentes que figuran en el formato y firmando dicho registro, adjuntando la nota de envío.
- Si el envío se remite al domicilio del cliente a través de alguno de los empleados de nuestra empresa que se desplacen a las instalaciones del Cliente, el técnico de nuestra empresa responsable del transporte firma una copia de la nota de envío que se adjuntará al formato y se hará responsable del equipo.
- Entrega a calidad el formato firmado, con las órdenes de entrada / salida, para archivar.

⁴¹ Enrutador (en inglés:router), ruteador o encaminador es un dispositivo de hardware para interconexión de red de computadoras.

3.3.3.4 Migración.

El objeto principal de este grupo es la captura de los datos del cliente convertirlos e incorporarlos a la aplicación.

Procedimiento:

Comienza con la recepción de la Carta del Proyecto por parte del Área Logística. Finaliza con el archivo del acuse de recibo de devolución de cintas del cliente.

Migración:

- Emite un formato, y se pone en contacto con el cliente para solicitar la información necesaria para su trabajo. Será responsable de la gestión de los datos y control de las versiones.
- El cliente envía o comunica los datos que se le han solicitado.
- Recibe el formato completado por el cliente, o bien rellena éste con los datos que el cliente comunica. Identifica este documento con el código del cliente, nombre, fecha y responsable del cliente que proporciona dichos datos y abre una carpeta provisional para el cliente, identificándola con el nombre y el código del cliente.
- Durante el período de la conversión, el formato y la copia controlada de la Carta del Proyecto están archivados en la carpeta provisional del cliente.
- El soporte en el que se reciben los datos del cliente es sellado con el código, nombre del cliente y la fecha de recepción. Colgando de las carpetas del PC "Originales" y "Convertidos" se crea una carpeta con el nombre del cliente. En la carpeta 'Originales' se guardarán los datos originales extraídos del soporte, y en "Convertidos" se dejarán los ficheros extraídos de los originales con el formato adecuado para poder ser posteriormente procesados. En caso de que se reciban los datos por e-mail, se archivarán éstos en la carpeta "Originales".
- Analiza la información recibida del cliente, revisando que cumple los requisitos que se encuentran en el documento "Normas de integridad de los datos a convertir". Si ésta no es correcta, no es suficiente o está dañada, se pondrá de nuevo en contacto con el cliente para que repita el envío, o envíe nuevos datos. Los nuevos datos enviados conformarán una nueva versión, con su fecha correspondiente y responsable del cliente que los suministra.
- En caso de una incidencia con los datos, emite un *Reporte* y se lo envía al cliente, con copia a Logística. Archiva este formulario en la carpeta informática del cliente, dentro de "Conversiones".
- El técnico que realiza la conversión será el encargado de aclarar al cliente cualquier duda que le surja relacionada con los *Reportes*.
- Si el número de bases de datos que el cliente entrega para convertir no coincide con el contenido en la casilla establecida al efecto del formato, se lo comunicará a Administración para que tome las medidas correspondientes.
- El técnico mantendrá informado a Administración sobre la evolución de la incidencia, y la viabilidad o inviabilidad de la conversión.
- Cuando la información permite empezar a trabajar, se analiza si la base de datos es o no conocida.
- Cuando la base de datos es completamente desconocida se le solicita información al cliente, y se recopila datos de otras fuentes (Internet, manuales, etc).
- Habla con el cliente para que le envíen los datos a extraer de su aplicación. Estos datos pueden recibirse por fax, correo electrónico, y excepcionalmente por teléfono.
- Si los datos llegan por fax, anota el código del cliente, su nombre, y se archiva en la carpeta provisional del cliente abierta para la conversión. Si se envían por e-mail, se

archiva el fichero en la carpeta "originales" del PC. Si se saca una copia del correo, se archiva en la carpeta provisional del cliente con su código, nombre del cliente y fecha de recepción. Si el cliente ha suministrado la información por teléfono, ésta se anotará en un formato telefónico con la fecha en la que se obtuvieron, el código del cliente y su nombre, y la persona que suministra estos datos. En este documento también se anotarán todas las incidencias que se produzcan en el proceso de conversión. Este documento también se archiva durante el proceso en la carpeta provisional del cliente.

- Una vez que se dispone de suficiente información sobre la base de datos, se envía la información a Servicios (Área Conversión) para su conversión.
- Se entrega a Logística el Formato de Migración.
- Logística entrega a Migración el formato de Migración con la fecha de finalización, y la dirección IP del servidor del cliente.
- Si se trata de una instalación nueva, se introducen los datos en el servidor del cliente.
- Si se incorporan los datos a un cliente que ya existía, se comprueba que estos datos no existieran antes, para evitar duplicarlos.
- Una vez incorporados los datos, se entrega a Logística el formato de migración, con lo que se comunica que las tareas de incorporación han terminado.
- Entrega junto con el formato de migración el reporte de incidencias, rellenando todos los datos, salvo el del nombre del técnico, que será completado por Administración "Certificación sobre la calidad de los datos integrados", para que el técnico certifique con el cliente durante la instalación, la calidad de los datos integrados.
- Cuando se comienza la instalación, se solicitan al cliente el envío de los ficheros (mediante cintas, e-mails) con la última información actualizada de que se disponga.
- Al ser recibidas se les identifica con el nombre y código del cliente, y la fecha de recepción. Si se recibe por e-mail, se vuelven a archivar en la carpeta "Originales", con la fecha de recepción. Con estas cintas se realiza el mismo proceso de la conversión, para los datos de almacén y de taller.
- Se solicita información al cliente con los últimos datos del almacén, taller. Se procedería a su identificación y a archivo de la misma forma que en el caso anterior (1ª petición de información), utilizándose también en este caso el reporte de incidencias.
- Realizará una copia de seguridad en formato CD de todos los ficheros convertidos, que posteriormente se enviará al cliente.
- Realización del reporte, recogiendo el resultado de la conversión, junto con las incidencias que se hubieran podido dar.
- Se envía una copia del reporte al cliente, archivándose el original en la carpeta del cliente en el sistema informático.
- Si es necesario se pone en contacto con el técnico, para solventar las posibles incidencias durante la certificación de los datos convertidos, apoyándose en el reporte.
- Abre la carpeta definitiva del cliente en soporte físico, identificándola adecuadamente con su nombre y código de cliente, y traslada todo el contenido de la carpeta provisional del cliente a la carpeta definitiva del cliente propia del Área de Conversión. Esta carpeta forma un archivo de carpetas de clientes que controla el área de conversión.
- Archiva el formato de migración junto con la copia controlada de la Carta del Proyecto en la carpeta del cliente propia del área de conversión.
- Una vez finalizada la conversión se devolverán: a) las cintas originales o cualquier otro soporte original enviado por el cliente. b) La copia de seguridad en formato CD de todos los ficheros convertidos y c) Se enviará por correo ordinario con acuse de recibo, anotará en la petición la fecha de envío de todo ello y la fecha de recepción del acuse de recibo, éste quedará archivado en la carpeta física del cliente propia del departamento de conversión.
- Migración será el responsable de recibir, controlar y depositar los acuses de recibo.

3.3.3.5 Entrenamiento y Liberación.

El objeto principal de este grupo es el Desarrollo Humano, es decir actividades que contemplen el impulso de las habilidades y conocimientos de los recursos humanos asignados al proyecto de implantación; así como el Aseguramiento de la Calidad, es decir, actividades que den seguimiento al cumplimiento de las expectativas del cliente tanto para el producto como para el proyecto de implantación.

Una vez definido, probado, corregido y validado el modelo, se capacita a los usuarios en la instalación, configuración y utilización del sistema de acuerdo a su perfil, funciones y responsabilidades, así como con base en el diseño realizado en la fase anterior. El alcance de la capacitación podrá manejarse en base a usuarios finales o a un esquema de **Train the Trainers**⁴², quedando esto claramente definido y documentado. Enseguida, al estar capacitados los usuarios, se prepara el arranque de la aplicación para cumplir con la liberación del producto. En esta actividad, se realiza el check list de arranque y se libera el piloto en producción.

Durante esta etapa se diseñan y realizan las pruebas requeridas al modelo, para garantizar el correcto funcionamiento de la solución (con base en los criterios de aceptación del producto) una vez realizadas las pruebas, se da seguimiento a los puntos críticos, se corrige el modelo y se valida con el cliente.

Validaciones.

- Capturar inventario de autos.
- Validar información.
- Archivos de Cartera.
- Revisar información de cartera.
- Inventario físico de refacciones.
- Armar archivo de familias, ubicaciones y maestro de partes.
- Revisar inventario de Refacciones
- Revisar y validar catálogo de cuentas contables.
- Configurar clientes y proveedores de crédito.
- Prevenir e identificar riesgos.
- Identificar y documentar situaciones de riesgo.
- Elaborar planes de contingencia.
- Poner en marcha los planes de contingencia.

Matriz de Pruebas,

- Validar procesos autos.
- Validar procesos refacciones.
- Validar procesos servicio.
- Validar procesos contabilidad.
- Validar procesos cuentas por cobrar.
- Junta de avances.
- Capturar póliza de saldos iniciales.
- Soporte de la aplicación.
- Junta de avances.

⁴² Formación de Formadores

3.5 ATENCIÓN A CLIENTES

3.5.1 Objetivo de la Atención a Clientes

Identificar las tendencias de satisfacción al cliente, ubicando sus necesidades para desarrollar planes de servicio que generen mejoras en los procesos para el cumplimiento de sus expectativas. Atención a Clientes se encuentra dividida en dos Subáreas: Call desk y Posventa.

Descripción.

- Canal directo de comunicación con los clientes, a fin de que ellos puedan reportar los errores del sistema y hagan las peticiones de nuevas funcionalidades del sistema.
- Los materiales de trabajo de Helpdesk son los siguientes: Línea Telefónica: Comunicación con los clientes; Messenger: MSN, Yahoo, Skipe; **VPN**⁴³: con la cual el Helpdesk se conecta al servidor del cliente para ver su problema.
- **SAC**⁴⁴: Herramienta a través de la cual el Helpdesk le dará seguimiento a las observaciones reportadas por el cliente.
- Se le entregarán los manuales al cliente, y se capacitará al administrador del sistema para enfrentar las eventualidades que se presentan en la operación diaria del sistema.
- El usuario en particular detecta una anomalía o una carencia del sistema, por lo cual reporta a sus coordinador del modulo. El coordinador del modulo revisa el problema y determina si se trata de una error del sistema o falta de experiencia del usuario. Si se trata de un error o requerimiento, el jefe de departamento deberá preparar una explicación teórica del problema o requerimiento, el cual entregará al “administrador del sistema para reportar a la empresa. El administrador del sistema enviará por e-mail el problema mediante un formato especial previamente asignado.
- Adjunto se enviara la explicación adicional después de enviar la información el gerente de sistemas deberá llamar telefónicamente.
- El Helpdesk recibe la observación del cliente y debe capturar el formato en **SAC** el cual generará un folio de atención con el que se le responderá al cliente con su respectivo e-mail.
- Una vez generado el folio de atención el Helpdesk está obligado a que en un lapso no menor a 24 horas dar respuesta al cliente, mediante el siguiente procedimiento:
- Determinar la naturaleza de la observación, la cual se dividirá de la siguiente manera: Garantía. Falla del sistema; Nuevo Desarrollo: Requerimiento de una nueva funcionalidad que no tenía contemplado el sistema; Soporte Técnico. Duda del usuario respecto a la operación del sistema; Error del cliente. Realización de un procedimiento de manera incorrecta; Capacitación. Cuando por rotación de personal se necesita enseñar a usar el sistema a una persona nueva.
- Par el caso de las garantías se atienden inmediatamente y no implica un costo adicional para el usuario, el Helpdesk debe determinar el tiempo estimado de atención así como verificar que se tengan toda la información necesaria para ser atendida la garantía, en caso ambigüedad en la definición del problema se deberá llamar al cliente para pedir información adicional. Con los complementos juntados y os tiempos definidos se pasará el requerimiento al departamento de desarrollo.
- El los casos de nuevos desarrollos se le informará al cliente que es algo con lo que no cuenta actualmente el sistema y que para su adecuación deberá pagar una suma adicional de dinero proporcional al trabajo en horas que se necesita para programar dicho requerimiento. Posteriormente se responderá al cliente a través de una cotización para que el cliente evalúe los tiempos y los costos necesarios para el desarrollo, y deberá enviar la confirmación del requerimiento con el respectivo pago o la cancelación de dicha adecuación.

⁴³ VPN: Acrónimo de siglas en Inglés Virtual Private Network ó Red Privada virtual

⁴⁴ SAC: Sistema de Atención a clientes

- En la parte de soporte técnico el Helpdesk resuelve dudas de los clientes, ya que cuenta con la **VPN** y puede ver lo mismo que el cliente esta viendo, así le puede explicar como debe ir el proceso. Una vez atendida la duda se le enviará al cliente la conclusión por correo para formalizar la atención al requerimiento.
- Con la **VPN** se detectan os errores, por lo cual el Helpdesk procederá a cancelar el requerimiento e indicar la causa del problema al cliente mediante un e-mail.
- La capacitación se deberá concertar el horario con los clientes para que dentro de las instalaciones del la empresa se de la capacitación al personal que lo requiera.
- Cabe destacar que el sistema de atención a clientes deberá trabar sobre la Web, para recortar los tiempos de atención con los clientes de provincia.

Objetivos Generales:

- Asegurarse de que los requisitos establecidos se determinen y se cumplan con el propósito de aumentar la satisfacción del cliente.
- Comunicar a la empresa la importancia de satisfacer los requisitos del cliente.
- Transmitir a los empleados el pensamiento y acción de mejora. No se trata de trabajar más sino de trabajar eficientemente.
- Tener el compromiso de resultados con el cliente.
- Brindar apoyo al cliente.
- Mostrar la responsabilidad adquirida sobre el proceso frente al cliente.
- Enfocarse a las necesidades de cada cliente como único para ofrecer calidad en el servicio logrando su satisfacción.
- Monitorear frecuentemente la calidad del servicio y dar seguimiento a las necesidades constantes de nuestro cliente.

Políticas de Atención a Clientes:

- Mantener una actitud positiva ante los clientes.
- Cumplir con una metodología práctica que permita establecer las bases para una cultura de servicio y calidad.
- Registrar en la base de datos la información de manera eficiente, que permita monitorear a los clientes.
- La política de quejas de la empresa es la base de recuperación y fidelidad de los clientes, estas deberán ser atendidas hasta la validación del cliente.
- Los clientes deberán ser monitoreados y atendidos de manera continua.
- Todas las quejas deberán ser registradas y atendidas hasta la validación del cliente.

Actividades Generales:

- Generar la estadística referente a Servicio y Atención a Clientes.
- Generar reportes del área para medir los avances en la satisfacción del cliente.
- Establecer mecanismos de acercamiento periódico como encuestas, cuestionarios para verificar satisfacción al cliente.
- Coordinar y dar seguimiento en los proyectos, garantizando la viabilidad y el resultado del valor agregado para la organización y los clientes.
- Identificar oportunidades de mejoras en los clientes.
- Coordinar y dar seguimiento a las necesidades requeridas por los clientes.
- Identificar junto con la Gerencia, Desarrollo y Soporte Técnico aquellos requerimientos que pueden ser factores de un valor agregado para los clientes.
- Trabajar estrechamente con todo el personal de la empresa para identificar mejoras y necesidades generales del mercado.

Atención a Clientes

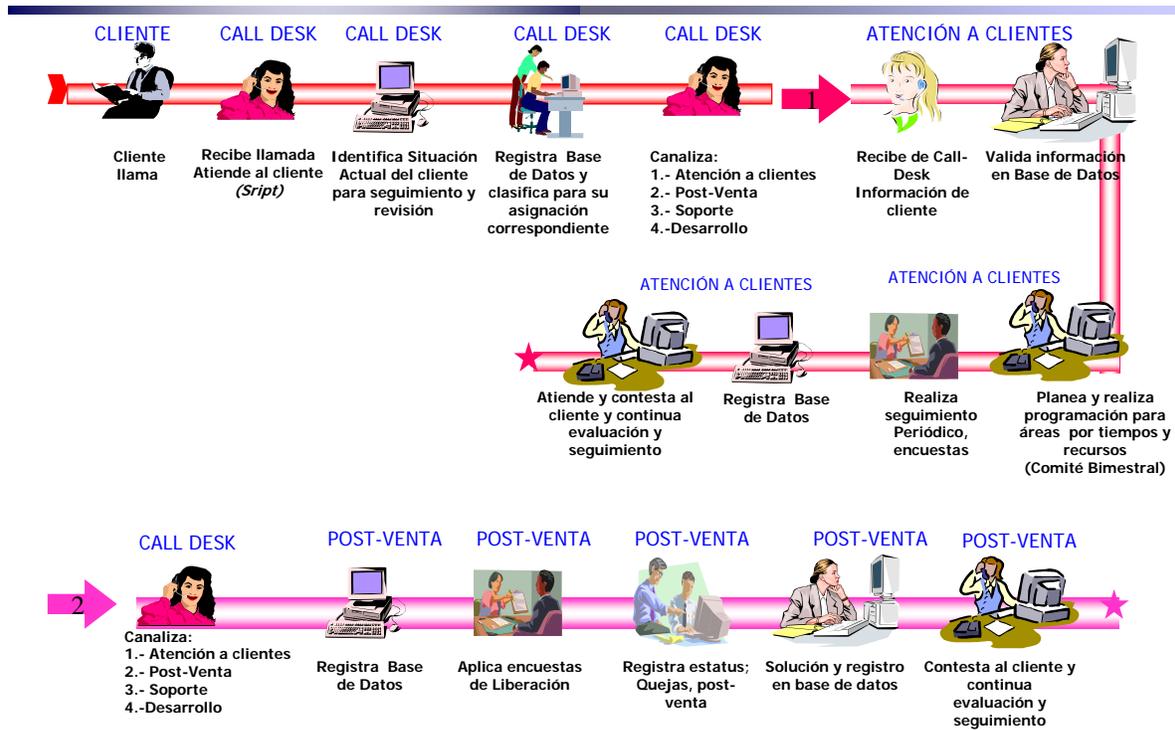


Imagen 6. Ilustración del Proceso Operativo del Área de Atención a Clientes de la Empresa

Procesos de Atención a Clientes:

1. Determinar estado actual de la agencia.
 - Definir prioridades del cliente.
 - Actualizar información del cliente en la base de datos.
 - Programar visitas, presentaciones, cursos de capacitación, con Soporte Técnico y Desarrollo.
 - Calendario de citas en coordinación con las áreas de respaldo.
 - Seguimiento y continuidad a las necesidades, sugerencias y quejas del cliente.
 - Determinar que problemas periféricos puede haber en le requerimiento del cliente.
2. Organización del Comité bimestral de Agencias.
 - Determinar el lugar físico, el día y el horario.
 - Notificar al cliente.
 - Detectar nuevas oportunidades de negocio.
 - Posicionar en las agencias la imagen de Sapsa con artículo de promoción que se les obsequia a clientes distinguidos.
3. Generar la estadística de satisfacción al cliente.

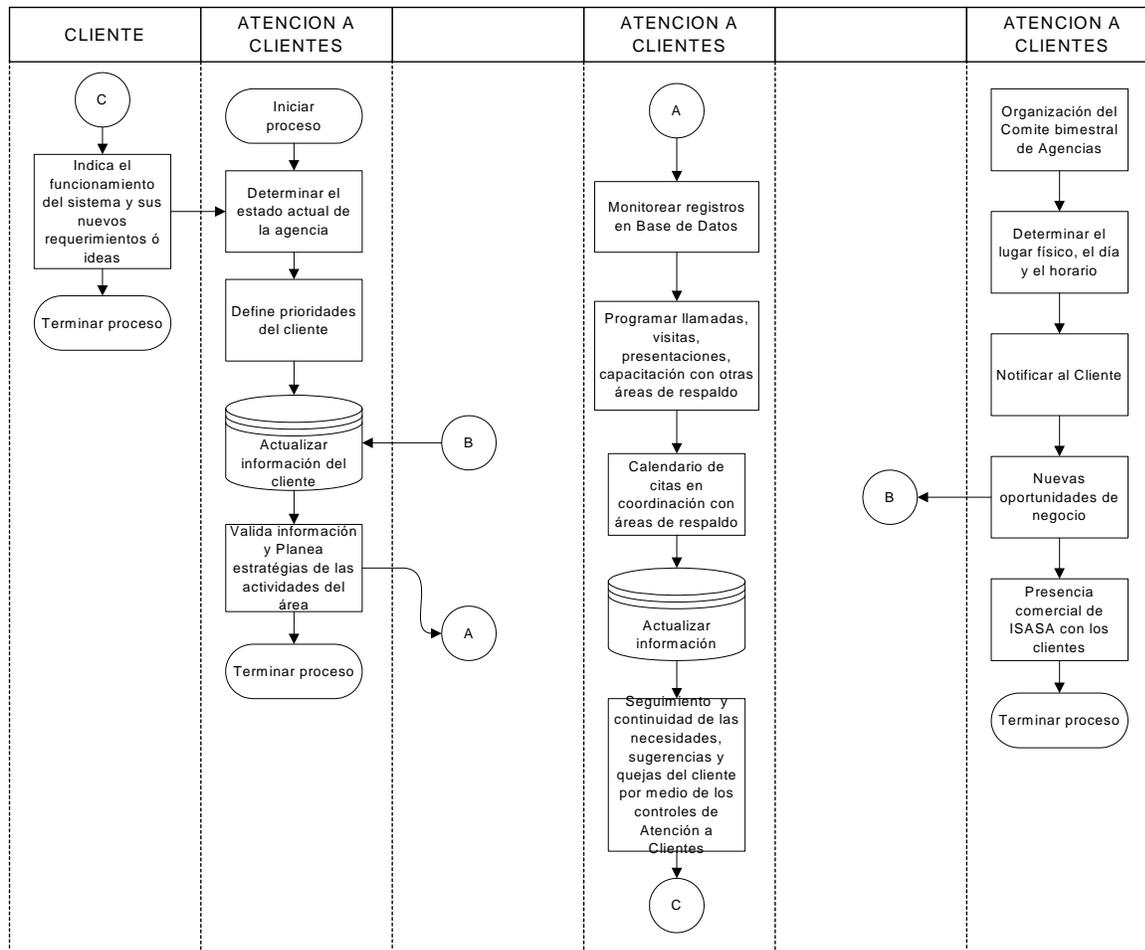


Imagen 7. Diagrama de Flujo de Información del Proceso Operativo de atención a clientes.

3.5.2 Call Desk⁴⁵

Objetivo de Call Desk:

- Registrar de manera completa y oportuna los requerimientos del cliente, para proporcionar la información necesaria a las áreas relacionadas en la solución del problema.
- Mantener informado al cliente sobre el estado de sus requerimientos.

Indicadores de Calidad:

- Número de quejas atendidas y resueltas
- Entrega contra fecha del cliente
- Entrega contra fecha prometida
- Índice de asistencia y puntualidad
- Carga de trabajo por persona

Políticas de Call Desk:

- Seguir el protocolo establecido para la atención de clientes.

⁴⁵ El Servicio técnico (en inglés Service Desk o también dependiendo de su implicación, Call Center (Centro de Llamadas), Contact Center, (centro de contacto) o Help Desk es una capacidad fundamental dentro de la Gestión de Servicios de Tecnologías de la Información.

- Obtener toda la información solicitada por las demás áreas para ayudar en la solución de los requerimientos y la detección de oportunidad de negocio.

Actividades Generales:

- Atender vía telefónica, fax ó e-mail, los requerimientos del cliente.
- Definir con el cliente su problema para establecer el requerimiento de acuerdo a las necesidades de la empresa.
- Monitorear constantemente el avance de los requerimientos para mantener informado oportunamente al cliente.
- Notificar al Coordinador del área de Atención a Clientes sobre los retrasos reportados en la programación de entrega de soluciones.
- Canalizar las llamadas de suma importancia al área correspondiente en caso de que la prioridad del requerimiento lo amerite.

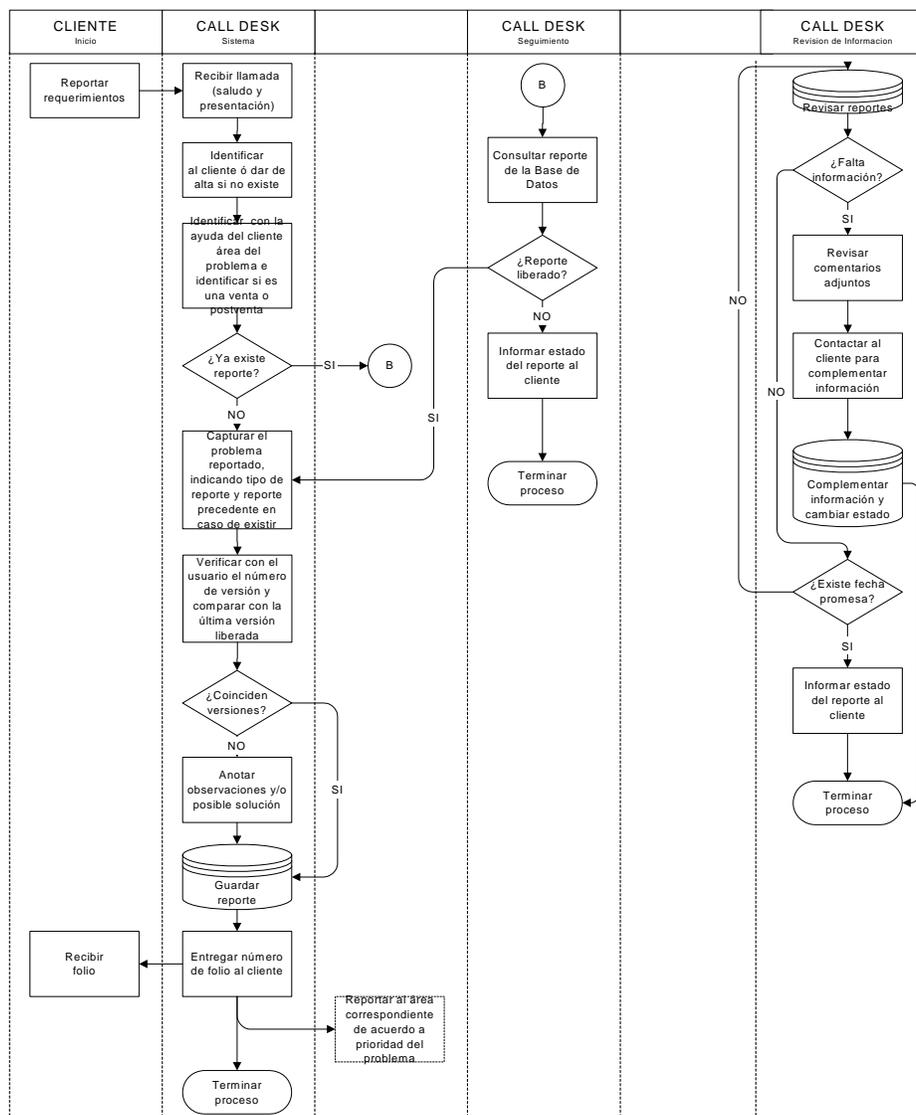


Imagen 8. Diagrama de Flujo de Información del Proceso Operativo de Call Desk

3.5.3 Postventa

El objetivo del área es dar seguimiento a los requerimientos, Post Implantación y necesidades del cliente brindando un servicio de calidad, detectando nuevas oportunidades de negocio y evaluando periódicamente el grado de satisfacción.

Indicadores de Calidad:

- Número de quejas atendidas y resueltas
- Entrega contra fecha del cliente
- Entrega contra fecha prometida
- Índice de asistencia y puntualidad
- Carga de trabajo por persona

Políticas de Postventa:

- Seguir el protocolo establecido para la atención de clientes.
- Obtener toda la información solicitada por las demás áreas para ayudar en la solución de los requerimientos y la detección de oportunidad de negocio.

Actividades Generales:

- Revisar base de datos para detectar el status del cliente.
- Realizar visitas a los clientes correspondientes.
- Aplicar encuestas de Liberación al cliente.
- Detectar quejas, sugerencias y/o postventa, dar seguimiento y solución a las mismas
- Cuando surja alguna venta, programar visita con el cliente acompañado de alguna persona del área a la que le corresponda.
- Dar solución y da de alta en la base de datos el nuevo requerimiento.
- Atender al cliente vía telefónica para continuar con la evaluación y da seguimiento.

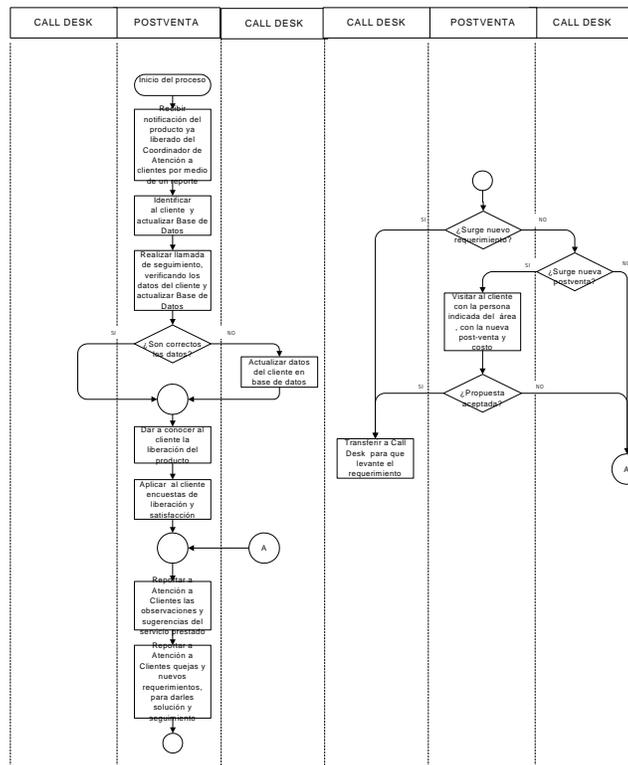


Imagen 9. Diagrama de Flujo de Información del Proceso Operativo de Postventa

3.6 MANTENIMIENTO Y DESARROLLO DE NUEVAS FUNCIONALIDADES

Descripción

- Departamento encargado de la programación del sistema. La organización del departamento es: Gerente de Sistemas, Líderes de Proyecto y Desarrolladores.
- Por cuestiones de logística, cada líder de proyecto atiende a un grupo determinado de agencias. La división de depende de la versión que tenga instalada el cliente, o por el tamaño y número de usuarios de la agencia o a veces se tienen proyectos regionales de acuerdo a los diferentes estados de la república donde se tenga instalado el sistema. Dependiendo de las necesidades de la empresa podrán abrirse más proyectos, por ejemplo hay proyectos específicos por marca. Es decir, cada cliente-agencia tiene un líder de proyecto asignado para atenderle y junto a él un equipo de programadores.
- El líder de proyecto es el encargado de recibir directamente del sistema las observaciones que se deriven de las agencias a su cargo.
- A través del Filtros en el **SAC** el líder de proyecto recibirá todas aquellas observaciones que sean “Garantías” o “Nuevos Desarrollos”.
- De acuerdo a la dificultad de la observación, la experiencia del desarrollador, y las cargas que se tengan de trabajo, y la prioridad, el líder de proyecto asignará la resolución del requerimiento a algún desarrollador de su equipo. Al momento de asignar el líder de proyecto establece una relación entre el programador y la observación del cliente (con un folio particular), así como también establece un tiempo en horas aproximado para la resolución del problema. Asimismo puede ayudar a reducir el tiempo de respuesta al usuario, ya que puede discutir con el programador la posible causa del problema o tener algún antecedente del mismo problema que ha sido resultado para otro cliente.
- Ahora, por otra parte cuando el programador entra al **SAC**, en él aparecerán los requerimientos que le fueron asignados por el líder de proyecto. Aquí se debe registrar cuándo comenzó a trabajar en cada uno de los requerimientos.
- El programador debe tener en cuenta la “prioridad” del requerimiento. Al referirnos a prioridad entendemos un concepto fundamental en la atención al cliente, ya que los problemas en el sistema de acuerdo a su naturaleza tienen diferentes “prioridades”, como se muestra a continuación: Urgente: se trata de un requerimiento que está parando la operación normal del sistema. Generalmente se trata de requerimientos donde no se puede ya sea cobrar o facturar; Rápida: Se refiere a problemas que entorpecen la operación del sistema; Lenta: Desarrollo de nuevos requerimientos
- El programador dará inicio a su jornada de trabajo con todos aquellos requerimientos que sean urgentes y así sucesivamente los irá cubriendo. El programador deberá tener en cuenta que sus soluciones deben considerar los siguientes elementos:
 - Capacidad de análisis para entender rápidamente el problema.
 - Dar las soluciones lo más rápido posible.
 - Las soluciones no deben mover otra funcionalidad que trabajen correctamente.
 - Se deberá evitará **Hardcodeo**⁴⁶, establecimiento de parámetros por código duro.
 - Hacer productos amigables para el cliente. Fáciles de comprender
- Toda vez que el desarrollador termine de implementar la corrección en el sistema o la nueva funcionalidad, procederá a terminar sus requerimientos en el **SAC**, capturando la causa del problema y la solución dada.

Objetivo:

- Desarrollar las adecuaciones, los nuevos productos y el mantenimiento de aplicaciones.

⁴⁶ Hardcodeo: Término entre programadores para escribir código sin hacer uso de parámetros modificables en la configuración del programa.

- Cumplir con las expectativas de tiempo, calidad y funcionamiento del cliente.
- Desarrollar productos innovadores y a la vanguardia tecnológica tomando en consideración el costo del producto final.
- Garantizar el cumplimiento de los planes de trabajo realizados en desarrollo
- Registrar todas las actividades generadas en desarrollo

Indicadores de Calidad:

- Numero total de horas de mantenimiento
- Estadística de Tiempos de respuesta
- Estadística de desviaciones en resultados por desarrollador
- Índice de asistencia y puntualidad

Políticas:

- Coordinar los esfuerzos del personal para el cumplimiento de la misión de la empresa.
- Capacitar constantemente al personal.
- Garantizar la confidencialidad de los productos de la empresa.
- Registrar las actividades y estatus
- Supervisar la integración de los productos
- Garantizar el desarrollo de todos los productos de la empresa.
- Todo desarrollo deberá alinearse de acuerdo a la metodología
- Todo desarrollo deberá integrar los estándares tanto funcionales, como técnicos.

Actividades básicas:

- Detectar y reconocer oportunidades de negocio para la innovación de productos.
- Garantizar el cumplimiento de los planes de trabajo.
- Generar reportes sobre el cumplimiento de las actividades del área.
- Realizar juntas semanales para detectar posibles desviaciones en el plan de trabajo.
- Elaborar planes de contingencia
- Llevar el control de cambios
- Administrar las bases de datos.
- Generar los indicadores de calidad, para tomar mejores decisiones
- Innovar en metodologías
- Identificar mejoras.

Desarrollo (Mantenimiento)

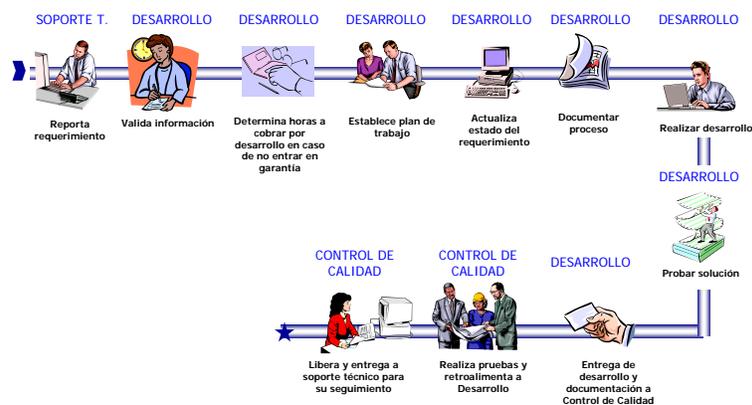


Imagen 10. Ilustración del Proceso Operativo del Mantenimiento al Sistema.

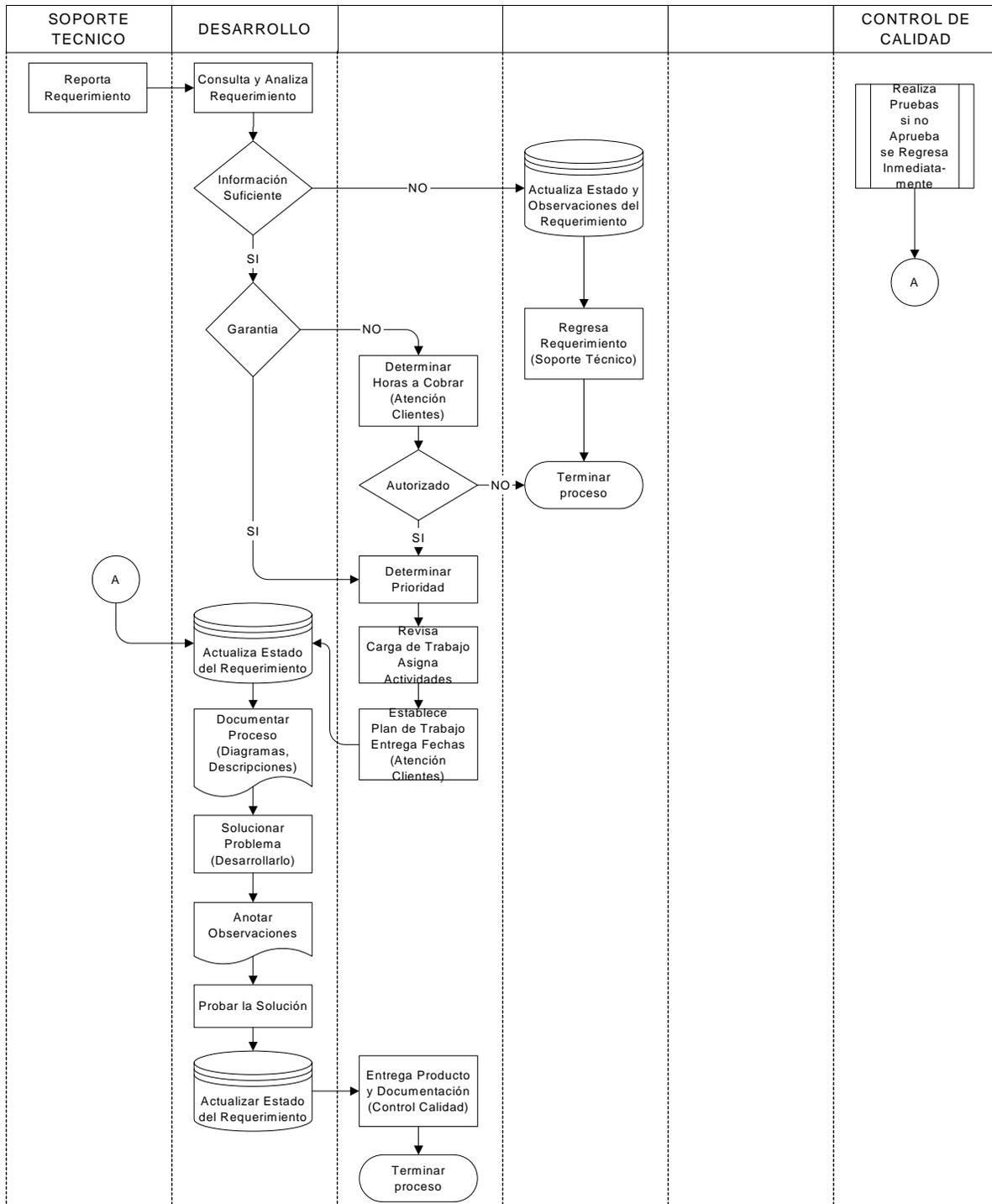


Imagen 11. Diagrama de Flujo de Información del Proceso Operativo de Mantenimiento.

Desarrollo (Nuevas Versiones)



Imagen 12. Ilustración del Proceso de Evolución de Producto

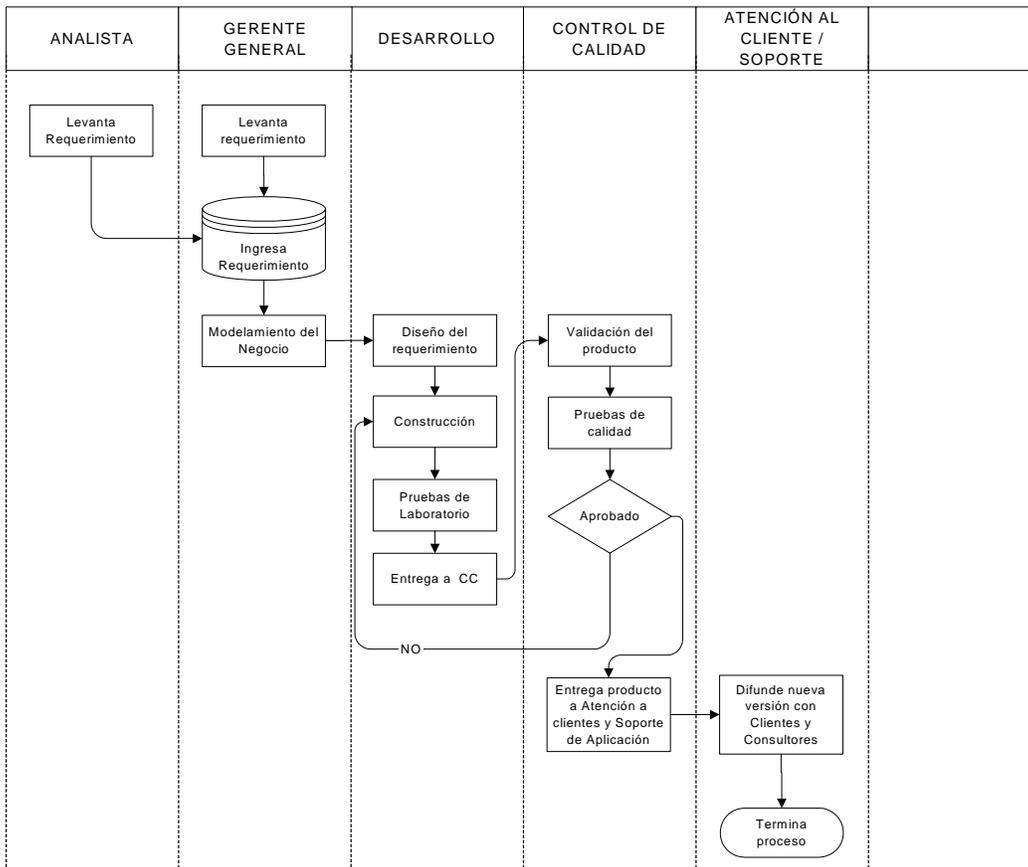


Imagen 13. Diagrama de Flujo de Información del Proceso Operativo de Evolución de Producto.

3.7 CONTROL DE CALIDAD

El objetivo de esta área es Recopilar, agrupar y evaluar evidencias para determinar si un producto cumple con los requisitos, mantiene la integridad en los datos, lleva a cabo los fines de la organización y utiliza eficientemente los recursos, de acuerdo a estándares establecidos.

Objetivos del área:

- Proporcionar al cliente la confianza y satisfacción en los productos de la empresa, creando las condiciones necesarias para evitar en la medida de lo posible errores y situaciones no previstas.
- Dar a conocer el grado de cumplimiento de los indicadores de calidad marcados al sistema. Así mismo, la detección de los riesgos donde existan inestabilidad del sistema y recomendaciones para realizar acciones correctivas y solucionar las mismas.
- Trabajar estrechamente con el área de Atención a clientes para lograr el mayor grado de satisfacción en los clientes de la empresa.
- Garantizar la confidencialidad, exactitud e integridad de la información
- Asegurar la eficiencia del sistema

Indicadores de calidad:

- Estadística de errores presentados (técnicos, estándares, concurrencia, performance)
- Porcentaje de éxito en pruebas realizadas.
- Porcentaje de productos nuevos vs. garantías al Software.
- Comparativo de porcentaje de éxito por desarrollador (Mes a Mes).
- Estadística de mayores fallas (modulo, área)

Políticas del área de Control de Calidad:

- El personal que realice los cambios ó modificaciones al sistema de acuerdo a los requerimientos del cliente, debe ser competente con base a la formación, habilidades y experiencia apropiadas.
- La empresa debe proporcionar y mantener la infraestructura necesaria en software y hardware para lograr la conformidad con los requisitos del producto.
- Establecer canales efectivos de comunicación dentro de la empresa y fomentar el trabajo en equipo.
- Crear procedimientos de control y análisis rápidos, sencillos, prácticos y confiables.
- Asegurarse de que se identifiquen y se apliquen los cambios requeridos al sistema.
- Asegurarse de que las versiones de los documentos que conciernen al sistema se encuentren actualizados y disponibles.
- Será responsabilidad de Control de calidad liberar los productos finales de la empresa

Actividades:

- Obtener y revisar requerimientos de las áreas correspondientes.
- Evaluar y validar los productos finales.
- Reportar y actualizar estatus en base de datos de los resultados obtenidos
- Motivar a todo el personal de la empresa, para que trabaje activamente en la búsqueda de la calidad
- Proporcionar a la empresa, indicadores de calidad, que permitan tomar mejores decisiones en la construcción y mantenimiento de los productos que la empresa genera.
- Cumplir con los requisitos que marca el cliente.

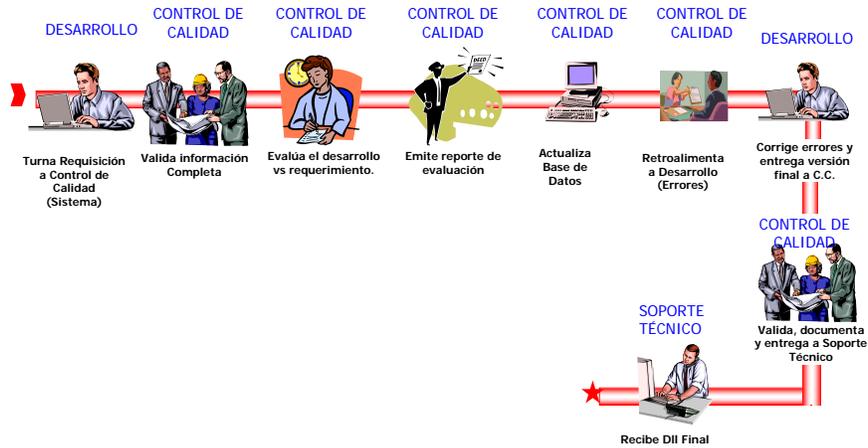


Imagen 14. Ilustración del Proceso Operativo de Control de Calidad.

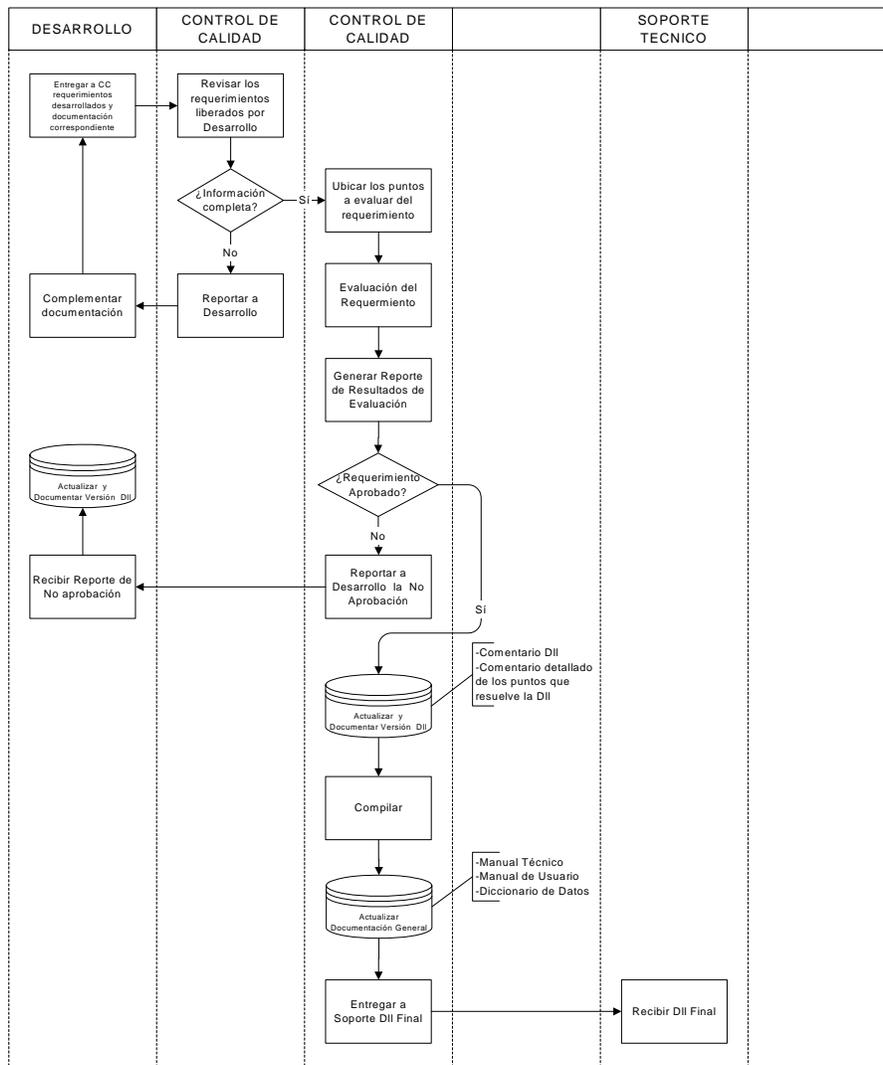


Imagen 15. Diagrama de Flujo de Información del Proceso Operativo de Control de Calidad.

4. EVALUACIÓN DE RESULTADOS GLOBALES

4.1 EVALUACIÓN DE RESULTADOS

A lo largo de este informe hemos tratado de ilustrar el proceso operativo mediante el cual una empresa mexicana produce software multiusuario. Conocer de antemano éste proceso resulta una gran ventaja para todo aquel que decida enfrentar el mismo problema. Esta premisa la hemos tomado como objetivo general del presente trabajo. Ahora, retomado los objetivos particulares establecidos al principio de tal manera que nos sirvan para evaluar la confiabilidad de la información, tenemos que en base a lo desarrollado a través de los capítulos se **obtuvieron los siguientes resultados:**

Respecto a objetivo particular número uno sobre el **Informar cómo funciona una empresa mexicana de IT¹**, podemos destacar que en el apartado anterior obtuvimos como resultado:

- El **Análisis** se desmenuza la manera de operar de la agencia, partiéndola en módulos más simples para facilitar su comprensión, y una vez que se comprende el proceso operativo se documenta y se establece *qué es lo que se necesita* automatizar y programar en el sistema;
- En la fase **Diseño** del sistema abordamos el problema del *cómo le vamos a hacer* para satisfacer las necesidades de la agencia, y establecimos como la parte crucial y más importante, ya que si se hace un mal diseño este hará que fracase todo el proyecto.
- Una vez diseñado el sistema comenzamos con la fase de **Construcción** donde en base a hojas de especificación, estándares de programación y funciones preprogramadas se crea la base de datos y se programa la interfaz con el usuario.
- La **Implantación** es un proceso que inicia con el *Plataformado* donde se instala todo lo necesario en el servidor del cliente, posteriormente en la *Parametrización* con la ayuda del cliente se llenan los catálogos principales en la aplicación, después en la fase de *Migración* se cargan los datos de ejercicios pasados así como las operaciones que todavía estén abiertas, y finalmente con la *Consultaría* se capacita al personal para operar .
- En la parte de **Helpdesk** tenemos una oficina especializada para atender todos aquellos problemas que el cliente enfrente con el sistema, ya sea vía telefónica, e-mail o pagina Web, el cliente hará peticiones las cuales *registraremos* todas en nuestro sistema de atención a clientes SAC y le daremos un *prioridad* de atención y un nivel de *dificultad* de resolución dejando en el nivel 1 toda aquellas peticiones que sean dudas de operación de los clientes, en el nivel 2 tenemos todos aquellos errores del sistema o Parametrización y para el nivel 3 tenemos a todos los nuevos requerimientos que se hagan del sistema.
- Para la parte de **Desarrollo** tenemos a un grupo de programadores especializados en atender peticiones de nivel 2 y 3, es decir *fallas del sistema y nuevas funcionalidades*;
- Por ultimo el departamento de **Control de calidad** se encargará de realizar *matrices de prueba*, enviar las DLL's² y *liberar* conjuntamente con el cliente los requerimientos, asegurando la completa satisfacción del cliente.

Ahora, hay que considerar que no solo es importante mencionar lo que se debe hacer en cada fase del proceso, si no que también es trascendental hacer notar que dado que la ingeniería de software es mas una ciencia que una técnica, tendremos varias maneras de obtener los mismos resultados, por ende resulta de vital importancia **enumerar las mejores prácticas encontradas en la industria automotriz que garanticen el óptimo funcionamiento de una agencia y por ende el éxito de nuestro sistema de información.**

¹ Information Technology – Tecnologías de la Información.

² DLL es el acrónimo de Dynamic Linking Library (Bibliotecas de Enlace Dinámico), término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo

Una de las partes importantes del informe es mencionar las trampas que enfrenta la empresa de **IT**, es decir que los pequeños problemas que si no controlamos se convierten en una bola de nieve y hacen que perdamos todo el control sobre el proyecto:

- El departamento de evolución de producto permite adaptar al sistema a nuevos requerimientos de información, como los cambios por nuevas políticas fiscales o requisitos por parte de la planta ensambladora. Para que se autorice la programación de un nuevo desarrollo este deberá ser documentada su necesidad, ya Cada vez que el cliente pide un nuevo desarrollo, y sus expectativas se ven cumplidas, el nivel de exigencia se vuelve superior y cada vez pide más, haciéndose insaciable e imposible de satisfacer, es decir el usuario no siempre tiene la razón y no siempre hay que programarle todo lo que pide.
- Cada nuevo requerimiento deberá estar sujetado a un análisis costo-beneficio, no siempre es factible poner en riesgo la estabilidad del sistema por cumplir caprichos de los clientes.
- La contabilidad es una técnica, por lo cual hay diferentes formas de aplicarse, sin embargo aunque los resultados sean los mismos, hay técnicas mas depuradas unas que otras. Siempre será recomendable proponer a los clientes los mejores Planes de cuentas contables encontrados, para evitar vía crucis con planes contables obsoletos o mal estructurados.
- Hay procesos engorrosos y mal planificados, pero bajo la excusa de que siempre se ha hecho así, los clientes se aferran a que se programe de esa manera. Es necesario advertirle al cliente de lo que puede pasar si programamos lo que el quiere.
- Los consultores que reciben las peticiones de los clientes deben conocer perfectamente la operación del sistema, para evitar aceptar peticiones extravagantes o ambiguas.
- Dejar que el cliente condicione el pago de mantenimiento a la creación de desarrollos.
- No tener un mecanismo para que en determinado momento se pueda “parar” el sistema en caso de que el cliente no pague su cuota de mantenimiento mensual.
- Aceptar información incompleta de parte del usuario para hacer la migración, ya que el consultor debe saber qué se debe subir y que no.
- Dejar Que el cliente cumpla con los planes de implantación a última hora.
- No preocuparnos por los clientes que no nos dan problemas, ya que aparentemente están bien, pero puede que estén pensando en cambiar de sistema.

Es necesario conocer las mejores prácticas encontradas, ya que hay que **identificar factores externos a la metodología de desarrollo que inciden directamente sobre el éxito o fracaso del sistema**, porque a veces aunque apliquemos al pie de la letra y con todo rigor la metodología de desarrollo, esta puede fallar si se deja influenciar por los factores externos que rodean a nuestro sistema:

- En México hay sueldos bajos, por eso existe un alto índice de rotación de personal, con lo cual se pierde expertiz en el sistema. Hay que programar capacitaciones constantes así como establecer las responsabilidades por cada perfil de usuario.
- En México es difícil que las personas acepten sus errores, y no solo eso, sino que suelen culpar a los demás. Se recomienda hacer que firmen para aceptar ya sea los que definieron en el análisis, o que recibieron la capacitación o que liberaron el requerimiento.
- En México los usuarios entregan todo de última hora por lo cual hay que darse un colchón suficiente para programar las tareas.
- En México hay personal en puestos claves por recomendación y que no cuentan con los conocimientos necesarios para tomar las decisiones correctas. Se recomienda si esto se presenta formar un comité especializado para que las personas que toman decisiones sean las que tengan los conocimientos técnicos necesarios.

Pero sabemos que de alguna manera **la enseñanza recibida incide de manera decisiva en el desempeño profesional**, porque los egresados de la unam se diferencian por tener una actitud proactiva frente a los problemas y de ser necesario no escatiman en implantar una

reingeniería de proceso a fin de tener los resultados esperados. Es factible reafirmar a manera de recomendación los factores en que los universitarios marcamos la diferencia:

- Cuando se parametriza no hay que esperar recibir la información de los catálogos, en vez de pensar en un departamento de recepción de información, hay que pensar en el departamento de recopilación de información.
- Las mejores soluciones no siempre son las mas caras o sofisticadas. Hay que pensar siempre en soluciones lo mas simples posibles.
- Es recomendable tener gente especializada en cada puesto, pero es más importante tener gente que conozca todo el proceso para apuntalar en un momento dado al departamento que se encuentre en problemas. Es valiosa la gente polivalente.
- Muchas veces se no se hace el trabajo sucio no es posible continuar, por tal razón a veces es necesario arremangarse la camisa para revisar un nodo de conexión, capturar alguna lista de datos. Se necesita una amplia disposición para hacer cosas que no necesariamente tengan que ver con el puesto ya que no se puede detener el proceso.
- Es necesario dominar diferentes aplicaciones, lenguajes y sistemas operativos. Si el Windows funciona para una agencia es posible que no funcione para otra y mejor meter Linux. Es recomendable manejar un amplio espectro de herramientas.

A manera de **compartir la experiencia de cómo solucionar los problemas enfrentados** tenemos la siguiente Lista de Experiencias sobre cómo solucionar Problemas:

- No hay continuidad en proyecto - Cada cliente debe tener un expediente de operación.
- Se piden cambios dos veces - Cada decisión tomada debe registrarse en el expediente.
- Perdida de procedimientos - Debe documentarse cada proceso operativo.
- Piden cosas contradictorias - No todos los usuarios pueden reportar errores.
- Retraso de Planes de Trabajo - Base de datos de Desarrollo y Producción.

Con los resultados obtenidos podemos establecer los tiempos para el ciclo de vida del desarrollo de sistemas Automotrices. De esta manera es posible establecer que el trabajo ha cumplido con el 100% de los objetivos, por lo cual la **evaluación global resulta positiva**. Este ciclo de vida es posible de cumplirlo en seis meses,. Cabe destacar que el ciclo de creación de sistemas se renueva cíclicamente, ya que por cada MARCA de autos se genera una versión diferente, debido a los requerimientos totalmente distintos que se tienen por planta ensambladora, y el equipo una vez que ha terminado con una marca, continua su ciclo con otra. La vida útil del software es de entre 5 y 10 años, después se creará una reingeniería³.

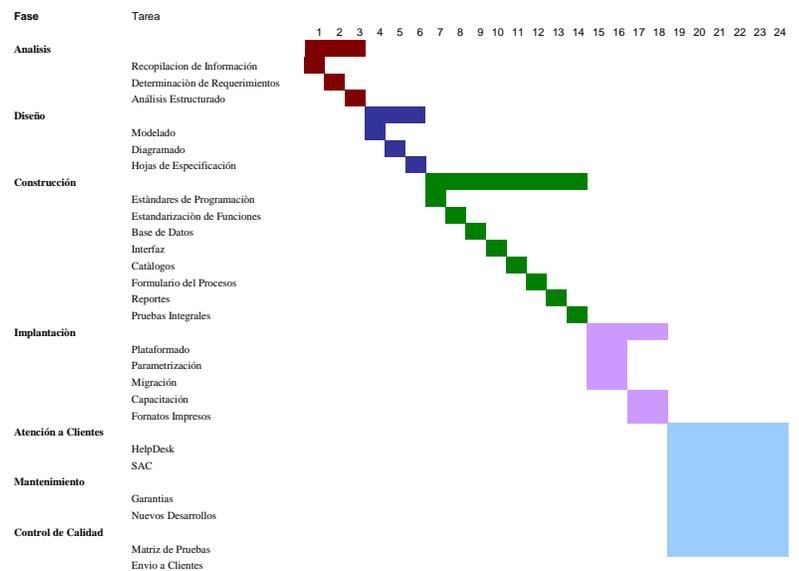


Imagen 1. Diagrama de Gantt del Ciclo de Creación de Sistemas.

³ La reingeniería de Negocios está definida por Hammer y Champy como “El repensamiento fundamental y el rediseño radical de los procesos del negocio para alcanzar mejoramientos dramáticos en medidas críticas y contemporáneas de desempeño como costo, calidad, servicio y rapidez.

CONCLUSIONES

¿Se cumplieron los objetivos?

En base a la evaluación *de los resultados obtenidos podemos concluir* que se cumplieron los objetivos trazados al principio del informe, ya que en cada capítulo y en cada fase se hicieron contribuciones interesantes que pueden tomarse como herramientas aplicables en la técnica de programación de sistemas. Los ejemplos cubren la metodología, que puede encontrar una aplicación general y no solo dentro del ámbito automotriz.

Resumen de la Aventura.

A través del presente informe hemos recorrido la aventura del ingeniero al crear software. Un camino sinuoso donde a cada momento se presentan disyuntivas y debemos elegir rápidamente la mejor alternativa. De no contar con la pericia para saber discernir podríamos obtener resultados inesperados. Imaginemos que el desarrollo de software es como un viaje largo en auto. Si planeamos el viaje con anticipación, reunimos los recursos necesario y tenemos precaución en el trayecto tenemos muy altas posibilidad de llegar a uno destino de manera exitosa. Si por el contrario. No revisamos el auto antes de viajar, contamos con recursos contados, y no tomamos las debidas precauciones corremos el riesgo de que nuestro proyecto no llegue a buen puerto.

La metodología comienza desde que se concibe el software. Desde que nos damos cuenta de la necesidad de una herramienta de esta magnitud. Se comienza por conocer nuestro negocio y tener la inquietud por hacerlo cada vez más eficiente. Durante el desarrollo del caso práctico vimos que los pasos a seguir son el Análisis, es decir determinar qué es exactamente lo que necesitamos, posteriormente sigue el Diseño, donde se establece cómo le vamos a hacer para obtener lo que necesitamos; seguido del Desarrollo, donde se construye la herramienta ara así poder pasar a la Implementación que es donde presentamos al sistema en sociedad y finalmente ayudarlo a dar sus primeros pasos a través de la atención a clientes; el mantenimiento y desarrollo y el control de calidad.

No es una labor fácil, ya que los responsables del proyecto enfrentaran diversos problemas que nunca serán iguales. De aquí que sea necesario utilizar toda la habilidad en sortear los imponderables que se presentan.

Puntos clave de la metodología.

Vamos a enlistar a manera de conclusión la lista de todas las decisiones importantes que se tomaron en forma de “tips”, y se va a explicar el porqué se tomaron de esa manera. De esta forma vamos a demostrar un criterio profesional para escoger la alternativa más apropiada en los puntos cruciales. Esto significa que dentro de una metodología que existen puntos finos sobre los que debemos hacer énfasis:

(a) Análisis:

- Siempre será más conveniente partir un problema grande en varios módulos.
- Es más prácticos dividir los módulos en diferentes Productos o Pantallas, donde cada producto realizará una etapa del proceso operativo del módulo.
- Cada análisis se realizará por “Producto” y se deberá dejar debidamente documentado en “Hojas de Especificación”.
- Las hojas de especificación contienen información de: Apariencia de la Pantalla; Detalle de controles; Explicación del proceso operativo; Detalle de las políticas de operación.
- Las hojas de operación deberán ser revisadas, validadas y firmadas por los usuarios, lo cual es una manera efectiva de evitar discrepancias futuras.

(b) Diseño:

- La recopilación de las Hojas de Especificación de Cada uno de los productos será un catálogo en el que se apoyará la construcción del proceso.
- Se deberá pensar siempre a partir de una estructura sólida o backbone.
- La principal preocupación del diseñador será tener una estructura sólida.
- Generalmente entendemos por una estructura sólida a tener un mecanismo uniforme para procesar y almacenar el objetivo principal para el que fue creado (Ej. Facturas).
- Así por ejemplo el objetivo principal de un sistema para agencias de autos será automatizar el proceso de compra-venta, por lo cual hay que garantizar primero la facturación de refacciones, de autos y de servicio, así como la entrada de dinero por la caja, creando un solo mecanismo adaptable para cada uno de estos elementos, así el mecanismo creará la facturación y la grabará, así como su correspondiente información contable y sus cuentas por pagar o por cobrar según sea el caso.
- Así lo que no nos debemos permitir que falle es la parte donde el dinero cambia de manos, por ejemplo en el proceso de autos los puntos clave serán: cuando se genera la entrada del auto, se registra la unidad en el inventario y se crea la póliza contable de entrada y a su vez la cuenta por pagar; cuando se factura la unidad y se registra la póliza contable de salida de la unidad y se crea la cuenta por cobrar; cuando el cliente paga la unidad y se ingresa el dinero al banco; cuando se liquida la unidad en la planta; Cuando se pagan los impuestos correspondientes de ISAN, IVA.
- Los elementos estructurales deberán tener el mecanismo para grabar el proceso, generar su póliza contable, afectar a las cuentas por cobrar, emitir la información a planta por Webservice ¹ y contemplar la cancelación en caso de que sea necesario.

(c) Construcción:

- La construcción del nuevo sistema o de la nueva versión comenzará al integrarse el equipo desarrolladores dirigidos por un líder de proyecto.
- El equipo de desarrollo acordará los “estándares de programación” referentes al conjunto de normas que rigen los lineamientos para programar para que de esta manera se garanticen tanto una estructura como una apariencia uniforme
- Establecerá los tiempos de entrega de los productos mediante un diagrama de Gantt.
- Cuando se entreguen los productos se creará un sistema de rotación entre el equipo de programadores para realizar pruebas en primera instancia revisando que cada uno de los productos (pantallas) cumplan con cada uno de los estándares de programación. Completada la revisión en primera instancia por parte del grupo de desarrolladores, se pasaran los productos terminados y probados al departamento de *control de calidad*.

(d) Implementación:

Se recomienda hacer un equipo de implementación de 3 personas: un consultor experto en el proceso operativo de la agencia y un desarrollador o con sólidos conocimientos de programación y un parametrizador con conocimientos contables para establecer el plan de cuentas y los catálogos iniciales, se cubre esta fase en las siguientes etapas:

- Plataformado: El desarrollador prepara el servidor, instala las terminales, etc.
- Parametrización: Se integran los catálogos principales de la aplicación
- Migración: Se pasa la información histórica al nuevo sistema
- Capacitación: Capacitar a la gente. Se recomienda hacer exámenes a los usuarios y en caso de que no tengan la calificación aprobatoria emitir las recomendaciones correspondientes.
- Pruebas e Impresos: El consultor recabará información respecto formatos, consecutivos de folios de facturas, etc. Ajustar los formatos y reprogramará cualquier cambio de última hora.

¹ Un servicio web (en inglés Web service) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Al final de la implantación los usuarios deberán liberar el sistema mediante documentos firmados donde aceptan el sistema que esta de acuerdo a lo analizado y recibieron la capacitación adecuada.

(e) Atención a Clientes:

- No siempre el cliente tiene la razón, y deberá estudiarse lo que pide para el sistema.

(f) Mantenimiento:

- En la ingeniería civil, como en la de software, el 60% de trabajo es de mantenimiento

(g) Control de Calidad:

- Se necesita un programa controlador de versiones como el Visual SourceSafe.
- Cuando el requerimiento es declarado como terminado por el área de desarrollo, automáticamente pasará al departamento de control de calidad.
- Este departamento es el encargado de realizar las pruebas necesarias para que no falle la solución, así como tendrá que verificar que no falle lo demás del sistema.
- Para su trabajo el departamento deberá diseñar matrices de pruebas donde valide que las correcciones y los nuevos productos estén aplicados correctamente.
- Si no cumplen con lo establecido se regresaran para su reprogramación y ajuste.
- Si cumple se envía la solución al cliente, con atención del administrador del sistema para que tome la DLL, verifique que la solución es la adecuada y enseñe a usar al usuario final.

4.3 MENSAJE FINAL

Hemos llegado al final del camino de la aventura digital que implica la creación de programas. En particular abordamos la vereda de sistemas para agencias de autos, pero el camino es muy similar en cualquier otra aplicación de la industria, sea para un hospital, un centro comercial, etc. La metodología del desarrollo de sistemas es una herramienta ampliamente usada en la industria del software y su correcta aplicación es la clave del éxito en la implementación de sistemas complejos requeridos por operaciones competitivas de un mundo globalizado.

No pretendo con este informe establecer un método estricto y lleno de normas, simplemente me gustaría que a éste trabajo se le diera una óptica de una “bitácora de viaje” que sirva de inspiración a los ingenieros que enfrenten una situación similar. El mundo de afuera es dinámico y cambiante, por lo cual sería imposible establecer un método científico, pero hacer una exploración preliminar a través de este texto para conocer las regla básicas que rigen nuestra profesión seguramente nos dará ventaja frente a nuestros competidores. En el viaje se enfrentan situaciones complejas e inesperadas, pero nunca se toman decisiones a la ligera, la parte medular del informe consiste en exponer con claridad el problema, desglosarlo y desmenuzarlos; después se plantean diversas soluciones; finalmente se aplica el criterio profesional para tomar la solución más óptima a los intereses de la empresa. Aunque cada uno pueda llegar a conclusiones diferentes, la actualidad de este problema ampliará la perspectiva de quien lo lea ya que ante todo esto es a manera sugerente.

Me alegrará mucho que esta herramienta los guíe por el camino del éxito, ya que le he puesto cariño, ganas e ilusión. Solo les pido considerar que las cosas no son absolutas y lo que fue real para mi puede llegar a ser absurdo para otros. Esto es solo mi óptica. Espero que la habéis pasado bien leyendo esto así como yo la pasé bien haciendo el reporte, sería muy bonito usa este reporte como guía para generar una empresa y por ende fuentes de empleo y bienestar. Muchas Gracias. Adiós.

REFERENCIAS BIBLIOGRÁFICAS

-JAMES A. SENN, Análisis y Diseño de Sistemas de Información, Segunda Edición, Mc Graw Hill, Abril 2000.

-ROGER S. PRESSMAN, Ingeniería de Software, Sexta Edición, Mc Graw Hill.

REFERENCIAS ELECTRÓNICAS

1.1.3 Impacto Económico y Social del Automóvil

"Efectos del Automóvil en las Sociedades"

<http://en.wikipedia.org/wiki/Automobile>

1.2.3 México dentro del contexto Mundial

World Motor Vehicle Production by Country and Type: Cars 2004 - 2005. OICA.

http://www.oica.net/htdocs/statistics/tableaux2005/worldproduction_cars2005.pdf

World Motor Vehicle Production by Country and Type: Light Commercial Vehicles 2004 - 2005. OICA.

http://www.oica.net/htdocs/statistics/tableaux2005/worldproduction_lightCV2005.pdf

1.3.1 Producción Nacional de Automóviles

"Asciende a 8 mil mdd inversión automotriz en los últimos seis años"

Notimex - El Universal - Viernes 05 de enero de 2007

"Otras consideraciones sobre la industria china"

<http://www.amda.org.mx> Gabriel Ortiz –Socio - KPMG en México

1.3.4 Panorama actual de las ventas de autos en México

"El Modelo de Experiencia del Cliente Automotriz"

<http://www.amda.org.mx> Alfredo Tsutsumi / A.T. Kearney

"Empresas armadoras pretenden producir y vender en México"

<http://www.amda.org.mx> Gabriel Ortiz - Socio de KPMG en México

"Oportunidades de México en el mercado automotriz actual"

<http://www.amda.org.mx> Gerardo Aguilar Cañada - Especialista en la Industria Automotriz PWC

"La industria automotriz mexicana y su perspectiva para el 2007"

<http://www.amda.org.mx> Gabriel Ortiz - Socio de KPMG en México

1.3.6 La Agencia Automotriz

"Retos y tendencias de los distribuidores en México". 04 de septiembre 2006

<http://www.amda.org.mx> Enrique Ochoa Báez - Socio coordinador de la industria automotriz en PWC México

1.4.1 Retos actuales de la agencia automotriz después de TLC

Automotriz: una industria de cambios y retos

<http://www.amda.org.mx>

Mercado automotriz: intensa competencia

<http://www.amda.org.mx>

Cada vez más modelos

<http://www.amda.org.mx>