



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES

ACATLÁN

SISTEMA DE INFORMACIÓN PARA EL CONTROL EN TIEMPO REAL Y
REPORTE VÍA INTERNET DE VENTAS DE COMBUSTIBLE EN
ESTACIONES DE SERVICIO.

CASO PRÁCTICO: SISTEMA DLT CARD DEL GRUPO DLT GASOLINERAS.

BAJO LA OPCIÓN DE
MEMORIAS DE DESEMPEÑO PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA:
MARCOS ACOSTA ZARCO

ASESOR: LIC. MAYRA OLGUIN ROSAS

OCTUBRE, 2005

m1349028



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.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: MARCOS ACOSTA
ZARCO

FECHA: 13 OCTUBRE 2003

FIRMA: [Firma]

Agradecimientos.

A mis padres de quienes recibo apoyo, comprensión y sobre todo amor. A mi madre Rebeca Zarco García quien me ha enseñado a valorar las cosas importantes de la vida, a quien agradeceré siempre el desgaste a mi lado y a quien quiero infinitamente. A mi padre José Acosta Ruiz, quien siempre me ha guiado para ser una persona de bien, responsable y honrada, con su ejemplo de dedicación al trabajo.

A mis hermanos Georgina, Marcela y Andrés por su cariño incondicional y las alegrías que me han dado. Isaac gracias por alentarme siempre para llegar al final de este objetivo.

A quienes tengo la bendición de llamar amigos, que forman parte de mí, que siempre han estado ahí, que me alientan a seguir por más, que han sido mis compañeros en este camino y que sin su ayuda mucho de esto no hubiera sido posible: Roberto Guzmán, Oscar Betanzos, Ramón Acosta, Miguel Acosta, Carlos Alemán, David Ortega, Ubaldo García, Israel García, Olga Silva, Antonio Argueta.

Lic. Rubén Nava González, Director General de Grupo Brago S.A. de C.V. por su amistad, confianza y apoyo recibidos durante la realización de este sistema.

TÍTULO.

SISTEMA DE INFORMACIÓN PARA EL CONTROL EN TIEMPO REAL Y REPORTE VÍA INTERNET DE VENTAS DE COMBUSTIBLE EN ESTACIONES DE SERVICIO.

CASO PRACTICO: SISTEMA DLT CARD DEL GRUPO DLT GASOLINERAS.

OBJETIVO.

Desarrollar un sistema informático útil que facilite la administración y el control en tiempo real de las ventas de combustible utilizando las tecnologías de información e Internet.

ÍNDICE.

Página.

INTRODUCCIÓN.

CAPÍTULO I. CONTEXTUALIZACION

Introducción.....	1
Antecedentes.....	1

CAPÍTULO II. PROBLEMA. NECESIDADES DENTRO DE LAS GASOLINERAS (ESTACIONES DE SERVICIO)

Introducción.....	5
2.1 Servicio al cliente.....	5
2.2 Calidad y servicio dentro de las estaciones de servicio.....	7
2.3 Franquicia PEMEX, realidad y futuro.....	10
2.4 Control de despachos.....	11

CAPÍTULO III. CICLO DE VIDA DE UN SISTEMA Y SUS CONCEPTOS BÁSICOS.

Introducción.....	13
3.1 Identificación de problemas, oportunidades y objetivos.....	16
3.2 Determinación de los requerimientos de información.....	17
3.3 Análisis de las necesidades del sistema.....	18
3.4 Diseño del sistema recomendado.....	20
3.5 Desarrollo y documentación del software.....	22

3.6	Pruebas y mantenimiento del sistema	22
3.7	Implementación y evaluación del sistema.....	23
3.8	Evaluación.....	27

**CAPÍTULO IV. ANÁLISIS DE REQUERIMIENTOS DEL DISEÑO.
METODOLOGÍA DE TRABAJO.**

	Introducción.....	29
4.1	Características del modelado de sistemas en la actualidad.....	29
4.2	Modelado de sistemas basado en UML.....	36
4.3	Análisis de requerimientos.....	43

CAPÍTULO V. PROPUESTA DE SOLUCIÓN.

5.1	P.O.S. (Point Of Sale) Terminales de punto de venta Modelos y características de las terminales Verifone.....	57
5.2	Lenguaje de programación WML, implementación de aplicación en servidor. Diagramas y pantallas en terminales.....	61
5.3	Sistema administrativo DLT Card.....	72
	CONCLUSIONES.....	78
	ANEXOS.....	81
	BIBLIOGRAFÍA.....	98

INTRODUCCIÓN.

Tradicionalmente las operaciones en una estación de servicio se llevan a cabo con dinero en efectivo, la operación no va más allá de cargar gasolina y pagar por este producto, sin obtener algún valor agregado a esta operación.

Los avances tecnológicos en el área de la computación permiten visualizar aplicaciones en las que el cliente se vea beneficiado de alguna manera, ya sea una atención más rápida, un servicio más confiable o el uso de medios de pago electrónico, diversificando los servicios que se pueden prestar en una gasolinera, convirtiendo a esta, en una verdadera estación de servicio.

Es así como el sistema DLT Card desarrollado para el grupo DLT Gasolineras, cubre una de estas necesidades, un servicio más confiable en el control de despachos de combustible de la estación de servicio, dejando abierta la posibilidad de aumentar más servicios.

La finalidad de este trabajo es mostrar el como, desde las bases teóricas que son importantes para cualquier desarrollo, hasta la implantación del sistema, el sistema DLT Card cumplió con su objetivo fijado, el control de despachos.

A lo largo del desarrollo de este trabajo, se presenta un panorama general del giro del corporativo al cual pertenece el grupo gasolinero para quien se implementó el sistema DLT Card, en donde

fue diseñado, la organización del área así como los involucrados en el sistema.

Se plantea la creciente necesidad en la que se ven, tanto las gasolineras como los clientes consumidores de la estación de servicio, de llevar el control de los despachos de combustible realizados.

Además tiene por objetivo mostrar la tendencia de los grupos gasolineros a ampliar el esquema de servicios que se pueden prestar en una gasolinera, aumentado así la diversidad de servicios y captación de más clientes.

Se describen las acciones para cubrir las necesidades de las estaciones de servicio, así como las áreas involucradas en cada una de estas.

Se hace una reseña sobre el ciclo de vida de un sistema con el objetivo de mostrar la metodología clásica recomendada en el desarrollo de los sistemas de información, para tener un punto de comparación con la metodología utilizada durante el desarrollo del sistema administrativo DLT Card del grupo gasolinera "DLT Gasolineras" y así formar un criterio válido del por qué la elección de dicha metodología.

Aunado a esto, se muestra, de manera general, la metodología de trabajo sugerida para llevar a cabo el proyecto, siendo esta, el Lenguaje de Modelado Unificado (UML Unified Modeling Language) en el cual fue basado el modelaje del sistema, así como el análisis de

requerimientos que fue necesario para llevar a cabo el diseño del sistema.

Se muestra como el sistema administrativo de la estación de servicio (DLT Card) junto con la Terminal de punto de venta (P.O.S.) y los reportes de consumo de combustible para los clientes de la estación de servicio vía Internet, constituyen las principales herramientas ofrecidas por el sistema implantado para dar respuesta a las necesidades que conllevan la venta de combustible en una gasolinera.

No se profundiza en el estudio de las técnicas o herramientas involucradas para el desarrollo, sino que se enfoca en los conceptos y teoría relacionados con el desarrollo de la aplicación.



CAPÍTULO I
CONTEXTUALIZACION.



Introducción.

En el presente capítulo se hace una descripción general del corporativo Grupo Brago, específicamente de la empresa Super Servicio DLT, para la cual fue desarrollado el sistema de información para el control en tiempo real y de reportes vía Internet de ventas de combustible en estaciones de servicio, denominado DLT Card, también se hace mención de los objetivos de la empresa, su visión y misión, del recurso humano con que se contó y las circunstancias que llevaron a buscar la solución del control de despachos.

Antecedentes.

Grupo Brago esta integrado por la empresa pionera llamada Diafa Diesel 2000 S.A. de C.V. constituida el 30 de marzo de 1981, la cual tiene como objetivo la comercialización de diversos derivados del petróleo, teniendo como proveedor único a PEMEX Refinación. La segunda empresa del grupo en aparecer es Combustibles B. González la cual se puso en marcha el día 6 de Julio de 1988, ambas empresas distribuidoras mayoritarias de PEMEX Refinación. Transportes Brago S.A. de C.V. se constituyó el 21 de Noviembre de 1997 siendo la tercera empresa del corporativo. El cuarto miembro del grupo es Super Servicio DLT S.A de C.V. comúnmente llamado DLT Gasolineras, la cual cuenta con cuatro estaciones de servicio ubicadas en los municipios de Atizapan, Toluca y Tlalnepantla.

Grupo DLT Gasolineras inició operaciones el 10 de noviembre de 1998 en la ciudad de Toluca. Desde la fecha y hasta el día de hoy, se ha ampliado la gama de servicios incorporando las estaciones de servicio Atizapan, Sor Juana y Pedregal.

El primer fruto del gran esfuerzo fue obtener el Reconocimiento NIVEL ORO en el año de 2003. reconocimiento que otorga PEMEX a estaciones de servicio por su excelente desempeño y cumplimiento de los estándares establecidos por la franquicia PEMEX.

Grupo DLT Gasolineras tiene como misión ser reconocidos en el mercado de combustibles por la calidad y servicio; en lo cual se realiza el mejor esfuerzo en todo lo que se hace, para constituirse como la mejor alternativa al consumidor, buscando el crecimiento de la empresa, la satisfacción de los clientes y el desarrollo de los colaboradores a través de la mejora continua.

Su visión es convertirse en la mejor opción en el mercado de los combustibles, ofreciendo el mejor servicio, productos y calidad. Cumpliendo con la normatividad del medio ambiente y la comunidad en un marco de respeto y desarrollo.

Dentro del corporativo se cuenta con un área de sistemas dedicados tanto al apoyo a la infraestructura tecnológica de cada una de las empresas del grupo, como al desarrollo e implementación de aplicaciones dentro de áreas de oportunidad, tanto administrativos como de control de las distintas empresas. Un área de oportunidad se encontró en DLT Gasolineras, debido a su creciente demanda de control de despachos, de captación de nuevos clientes así como de

mantenerse a la vanguardia en cuanto a desarrollos tecnológicos se refiere.

Esta área de oportunidad, que específicamente se refiere al control de despachos principalmente para clientes que cuentan con flotillas de autos, se había intentado darle solución desde tiempo atrás, con un sistema implantado a nivel nacional y por mandato directo de PEMEX, dicho sistema llamado GASOPLUS que carecía de soluciones reales para los administradores de las estaciones de servicio, no contaba con un control de dispensarios, no había reportes en línea además de que los malos manejos se hacían presentes, llevando a dicho sistema al fracaso como solución para la administración de gasolineras, debido a esto, esta área de oportunidad quedó vacante desde la salida de dicho sistema, tema que los administradores de las estaciones de servicio, no han dejado de apuntar como indispensable para cubrir dicha necesidad.

Es por esta razón que en Grupo Brago, se tomó la decisión de que el área de sistemas se comprometiera a desarrollar un sistema informático que cubriera las necesidades de los administradores, y no las de terceros, por lo que la decisión de desarrollarlo de manera interna no demoró demasiado en ser autorizada.

Dentro del área de sistemas del corporativo se cuenta con la siguiente estructura:

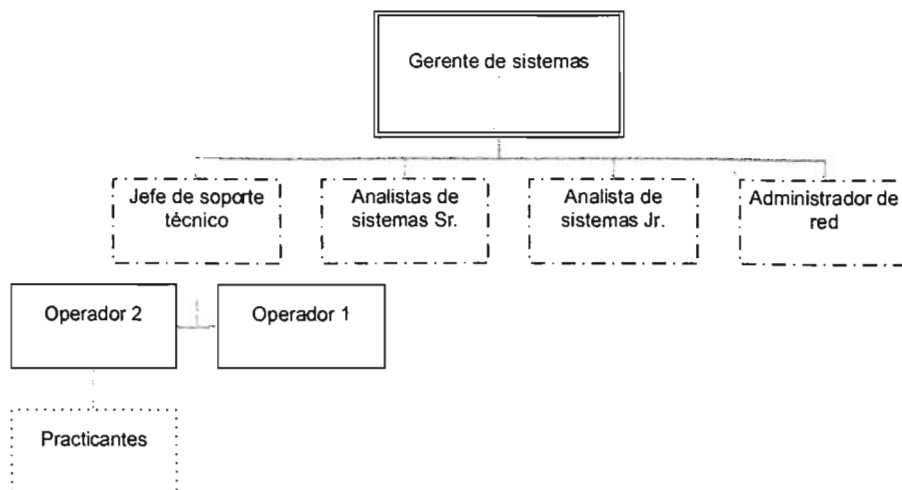


Fig. 1.1 Organigrama del área de sistemas

Para el desarrollo del sistema del cual es caso de estudio en el presente trabajo, se vieron involucrados los analistas de sistemas en el modelaje, programación, puesta a punto del sistema, administración y operación del mismo, pasando paulatinamente tareas asignadas a los operadores y practicantes del área como lo es la maquinación de las propias tarjetas, proceso que se explicará posteriormente.

Cabe mencionar que el desarrollo del sistema se llevó a cabo en las instalaciones del corporativo, realizándose la fase de implantación en la estación de servicio de Atizapan, habiéndose seleccionado esta, por el alto flujo de despachos que se realizan en esta, y así poder probar el sistema en tiempo real, con información generada en ese momento, además de que es considerada dentro del grupo, como la más importante de todas debido, como se mencionó con anterioridad, por el flujo importante de venta.



**CAPITULO II.
PROBLEMA.
NECESIDADES DENTRO DE LAS GASOLINERAS
(ESTACIONES DE SERVICIO)**



Introducción.

Durante el presente capítulo se plantea la creciente necesidad en la que se ven, tanto las gasolineras como los clientes de flotillas consumidores de la estación, de llevar el control de los despachos realizados, así como la búsqueda de nuevas estrategias de servicio dirigidas hacia el cliente.

Además, tiene por objeto mostrar la tendencia de los grupos gasolineros a ampliar el esquema de productos que se pueden prestar en una gasolinera, como el pago de luz, teléfono, etc. Compra de tiempo aire para celulares, pago de remesas, etc. aumentado así la diversidad de servicios y la captación de más clientes.

2.1 Servicio al Cliente

En la actualidad, vivimos en un mundo competitivo; los consumidores son cada vez más exigentes en cuanto a lo que desean recibir; si se trata de un producto o servicio esperan que sus expectativas o necesidades sean cubiertas al máximo, por lo que están en la búsqueda constante de establecimientos y empresas que satisfagan sus necesidades.

El éxito de un negocio consiste en mantener una buena relación con los clientes, proporcionando atención esmerada en todas las circunstancias que se presenten, ya sean las cotidianas o durante imprevistos. La actitud de excelencia (gran disponibilidad, atención esmerada, eficiencia y calidad) en el servicio brindado al cliente debe

convertirse en un hábito para cada empleado de nuestras Estaciones de Servicio.

Los clientes consumidores son el elemento principal de las Estaciones de Servicio, ya que sin ellos la Estación y todos los servicios complementarios que se ofrecen, sencillamente no tendrían razón de ser. Si se fracasa en el intento de allegarse y retener cada día a mayor número de clientes, se habrá fracasado en el objetivo principal.

Por ello, es de vital importancia brindarles un servicio que cubra sus expectativas de calidad y confianza. Proporcionar una atención esmerada y eficiente debe ser la principal preocupación del gasolinero y de su personal, principalmente de aquellos que mantienen el contacto directo con los clientes.

Si todos los empleados demuestran deseos, disposición, entusiasmo y preocupación por atender debidamente a todos y cada uno de los clientes que acudan a la Estación a cargar combustible para sus vehículos o para cualquier servicio adicional, seguramente el cliente regresará y se convertirá en un promotor del negocio, ya que lo recomendará a su familia y amigos; además será un cliente constante.

Proporcionar un servicio de calidad es una situación que implica gran esfuerzo para convertir en hábito constante actitudes como la amabilidad, honestidad y disposición para ofrecer un buen servicio y requiere dedicación permanente de todos los empleados de las Estaciones.

El desarrollo tecnológico y el crecimiento de la competencia que estamos enfrentando actualmente, se dan, por ahora, sólo a nivel interno entre gasolineros, que se dedican a la comercialización de

combustibles, lubricantes y productos relacionados al ramo. Dicha competencia es sana y positiva para la gran red de franquiciatarios Pemex, como una forma de mejorar los productos y los servicios que se prestan a la sociedad.

2.2 Calidad y servicio dentro de las Estaciones de Servicio.

Es definitivo, negocio que no implemente estrategias de calidad y servicio para una mayor y mejor satisfacción de sus clientes, tendrá graves riesgos de desaparecer.

Las exigencias de los consumidores están creciendo, intentando recibir valores agregados a cambio de los altos precios que actualmente se pagan debido a los problemas financieros por los que está atravesando el país, (como pueden ser la difusión de conceptos novedosos, o limpieza en el servicio), además de apreciaciones un tanto intangibles como son la cordialidad, amabilidad e interés por el cliente y mayores estándares de calidad en los productos y/o servicios que se comercializan.

La diferencia competitiva, se debe dar en estos mismos rubros, puesto que la percepción de todos los comerciantes está propiciando la estandarización de los servicios, con la única opción para los comerciantes de obligarse a facilitar al cliente la identificación de su concepto de negocio repitiendo características como son sus localizaciones e imagen corporativa.

Si algo es palpable actualmente en la preferencia de los consumidores, es que la infidelidad a los negocios ya no se da como antes por cuestiones de precio, ahora es por el servicio, de lo que deriva que la satisfacción de los clientes se debe dar enfocando todos los esfuerzos hacia el mejor servicio como objetivo.

No se puede aventurar a estructurar estrategias de algo que no se conoce, por lo cual se puede cuestionar lo siguiente:

¿Es todo lo bueno que el cliente desea y espera recibir, en lo referente a la excelencia del producto, marca o concepto?

¿Es la satisfacción plena de los deseos y /o necesidades de los consumidores?

Por lo tanto, el servicio tiene que ver con:

Rapidez, calidad, presentación, cortesía, seguridad, ambiente agradable, limpieza, iniciativa y dinamismo.

El esquema operativo de los sistemas de Franquicias ofrece Cinco Reglas de Oro para desarrollar un negocio, cuyas tendencias se dirijan hacia la satisfacción del cliente:

Conocimiento: El empleado debe conocer perfectamente su trabajo para la mejor atención de los clientes. Asimismo, la empresa franquiciante debe contar con estándares de medición para cada aspecto del negocio.

Constancia: Se debe ser constante en repetir el mismo modelo operativo en todas las unidades franquiciatarias, calidad constante, servicios iguales.

Supervisión: Toda actividad humana puede generar errores. Para evitarlos es conveniente y necesaria una función permanente de supervisión y apoyo.

Mantenimiento: Es vital para conformar un esquema de calidad. En algunas empresas sus gastos en este rubro, aplicado a inmueble, instalaciones y equipo, alcanzan hasta un 8% de sus ventas.

Servicio: Debe medirse permanentemente para conocer su calidad. Debe ser integral; desde la recepción del producto del proveedor hasta el cobro al cliente en la caja del establecimiento; inclusive los servicios de postventa.

Como se puede apreciar, las características del Sistema de Franquicias conllevan tendencias de calidad en el servicio. Entonces, y con mayores elementos, se puede asegurar que para alcanzar niveles óptimos en la calidad del servicio es necesario cumplir de manera eficiente y efectiva los requerimientos de un determinado cliente, de acuerdo a sus necesidades y expectativas.

Para mejorar constantemente el servicio, se deben analizar cuidadosamente los aspectos más relevantes de la operación de las unidades franquiciadas y los posibles errores que se estén cometiendo al respecto, buscando su solución inmediata además de establecer estrategias de mayor alcance para perfeccionar el propio Sistema de Franquicias.

2.3 Franquicia PEMEX, realidad y futuro.

El número de estaciones de servicio registra desde 1992, un crecimiento acumulado de 101 por ciento. En ese mismo año, las ventas de combustibles automotriz registran un crecimiento acumulado de 22.74 por ciento, lo cual es muy inferior al crecimiento de estaciones de servicio y lleva a una disminución significativa de la venta promedio por estación. También se promediaba una venta con una mezcla de los diferentes combustibles de 1 millón 40 mil litros mensuales. En 2004 se promedia una venta de 641 mil litros mensuales, lo cual nos muestra una caída del 38.36 por ciento de la venta promedio mensual, llevando esto a una rentabilidad generalizada. (1)

Un análisis cuidadoso de la historia reciente de la franquicia, hace pensar que PEMEX Refinación ha descuidado la planeación del crecimiento de su red y no ha considerado la rentabilidad necesaria de cada estación de servicio para lograr una red sólida, sana y logísticamente rentable.

Sin embargo, el sentir de una baja generalizada en rentabilidad de las estaciones de servicio no solamente es el reflejo de la baja en ventas, sino también de un alza en el gasto de operación.

En los últimos tres o cuatro años, el precio de los combustibles ha ido muy de la mano con la inflación, reflejando en teoría un aumento congruente del margen comercial, ya que es un porcentaje del precio de venta.

Esta aunado a que en el mercado gasolinero no se compite en precio, sino en los servicios que se puedan ofrecer en una estación, para así hacer uso de su nombre, estación de servicio, idea que han adoptado muchos propietarios de negocios como este, aumentando en sus estaciones diversos servicios como son pagos de otros servicios a su vez, bancos, tiendas de conveniencia, etc., haciendo esto que los franquiciatarios den un giro a su negocio, evitando su debilitamiento en el mercado y su posible desaparición.

2.4 Control de despachos.

Dentro del ramo gasolinero, el flujo de información es enorme, la cantidad de despachos que se realizan diariamente rebasa los 3500, la cantidad de litros vendida en un día puede llegar hasta los 70000 litros, haciendo esto que mensualmente una estación de servicio pueda llegar a vender hasta 2 millones de litros mensuales, por esto la cantidad de dinero que se maneja en una gasolinera se multiplica en cantidades mucho mayores.

El manejo de dinero, el control de despachos y la administración de un negocio como este requería de mucho personal, gente en la cual se pudiera depositar confianza tal como para administrar los recursos del negocio, esto no siempre era posible, por esto, la automatización de estos procesos da lugar a una gran área de oportunidad para el desarrollo de un sistema que controle dichos procesos.

Para el cliente consumidor de la gasolinera, el controlar el rendimiento de sus unidades es imprescindible, los malos manejos que

entre los chóferes de sus unidades y los despachadores de las estaciones de servicio se presentaban día con día, reflejándole al cliente perdidas para su negocio, es por eso que los clientes asiduos de las estaciones de servicio se ven beneficiados con un sistema de control, que en tiempo real y con restricciones de carga, puedan controlar las asignaciones de sus unidades.

Esto para el cliente le significaría el ahorro de mucho dinero en cuestiones que no debería preocuparle, pero que son cuestiones que por administración propia de su negocio son imperantes controlar.

Es así, que tanto estaciones de servicio como clientes con flotillas de dichas estaciones, conjuntamente se ven inmersos en la misma necesidad, que es el control de sus respectivos negocios.

Por lo cual, el presente trabajo pretende mostrar en que necesidades se encontraban las estaciones de servicio, y como, mediante un sistema de información, controlar en tiempo real los despachos, presentar reportes de estos despachos vía internet para los clientes que cuentan con flotillas de automóviles que necesitan controlar las cargas de estos, como para la estación de servicio y lograr con esto la captación de más clientes, o de mantener cautivos a clientes asiduos a ellos.



**CAPITULO III.
CICLO DE VIDA DE UN SISTEMA Y SUS CONCEPTOS
BÁSICOS.**



Introducción.

En el presente capítulo se hace una reseña sobre el ciclo de vida de un sistema informático con el objetivo de mostrar la metodología clásica recomendada en el desarrollo de los sistemas de información, esto, para tener un punto de comparación con la metodología que al final se utilizó durante el desarrollo del sistema administrativo DLT Card, y así formar un criterio válido del por qué el uso de dicha metodología.

En la actualidad, para muchas organizaciones los sistemas de información basados en computadoras son el corazón de las actividades cotidianas y objeto de gran consideración en la toma de decisiones. Las empresas consideran con mucho cuidado las capacidades de sus sistemas de información cuando deciden ingresar o no en nuevos mercados o cuando planean la respuesta que darán a la competencia.

Las organizaciones han reconocido, desde hace mucho, la importancia de administrar recursos principales tales como la mano de obra y las materias primas. La información se ha colocado en un lugar adecuado como recurso principal. Los tomadores de decisiones están comenzando a comprender que la información no es sólo un subproducto de la conducción, sino que a la vez alimenta a los negocios y puede ser el factor crítico para la determinación del éxito o fracaso de éstos.

Para maximizar la utilidad de la información, un negocio la debe manejar correctamente tal como maneja los demás recursos. Los administradores necesitan comprender que hay costos asociados con

la producción, distribución, seguridad, almacenamiento y recuperación de toda información. Aunque la información se encuentra a nuestro alrededor ésta no es gratis, y su uso es estratégico para posicionar la competitividad de un negocio.

A medida que las computadoras son empleadas cada vez más por personas que no son especialistas en computación, el rostro del desarrollo de sistemas adquiere una nueva magnitud. Los propios usuarios emprenden ya el desarrollo de algunos de los sistemas que ellos emplean. Por ello, es necesario comprender que un proyecto de desarrollo de un sistema computacional consta de varios pasos, los cuales ayudan a traducir las necesidades del cliente en un modelo de sistema.

Es por esto, que el ciclo de vida del desarrollo de sistemas es un enfoque por fases del análisis y diseño que sostiene que los sistemas son desarrollados de mejor manera mediante el uso de un ciclo específico de actividades del analista y del usuario, manteniendo dicha interacción desde el inicio del sistema hasta su implantación, conclusión y mejora.

Los sistemas de información se diseñan para lograr la sinergia, la acción simultánea de las partes separadas, aunque recíprocamente relacionadas.

Durante el desarrollo del presente capítulo, se busca obtener una base sólida sobre el ciclo de vida de un sistema, no buscando solo presentar definiciones técnicas, sino sustentar sobre la investigación

bibliográfica realizada, aportaciones personales obtenidas durante el desarrollo del sistema.

El análisis de sistemas, es el proceso de estudiar su situación con la finalidad de observar cómo trabajar y decidir si es necesario realizar una mejora sobre este.

El método del ciclo de vida para el desarrollo de sistemas es el conjunto de actividades que los analistas, programadores y usuarios realizan para desarrollar e implantar un sistema de información.

Aun no se esta de acuerdo en cuantas fases exactamente hay que dividir el ciclo de vida del desarrollo de sistemas, pero por lo general, es dividido en siete etapas, las cuales se explican durante el presente capítulo, aunque cada etapa es presentada en forma discreta, nunca se lleva a cabo como un paso aparte. En vez de ello, varias actividades pueden suceder simultáneamente, y las actividades pueden ser repetidas, según el modelo clásico y cuyas actividades han estado y estarán presentes en cualquier lugar donde exista un sistema.

3.1 Identificación de problemas, oportunidades y objetivos.

En la primera fase del ciclo de vida del desarrollo de sistemas el analista tiene que ver con la identificación de problemas, oportunidades y objetivos. Esta etapa es crítica para el éxito del resto del proyecto, debido a que nadie quiere desperdiciar el tiempo subsecuente resolviendo el problema equivocado.

La primera fase requiere que el analista observe honestamente lo que está sucediendo en un negocio. Luego, junto con los demás miembros de la organización, el analista hace resaltar los problemas.

Las oportunidades son situaciones que el analista considera que pueden ser mejoradas por medio del uso de sistemas de información computarizados.

La identificación de objetivos es también un componente importante de la primera fase. En primer lugar, el analista debe descubrir lo que está tratando de hacer el negocio. Luego será capaz de ver si algún aspecto de la aplicación de sistemas de información puede ayudar para que el negocio alcance sus objetivos atacando problemas específicos u oportunidades.

Las personas involucradas en la primera fase son los usuarios, analistas y administradores de sistemas que coordinan el proyecto. Las actividades de esta fase consisten en entrevistas a los administradores de los usuarios, estimación del alcance del proyecto y documentación de los resultados. La salida de esta fase es un estudio de factibilidad

que contienen una definición del problema y la sumarización de los objetivos. Luego los administradores deben tomar una decisión para ver si continúan con el proyecto propuesto.

3.2 Determinación de los requerimientos de información.

La siguiente fase a la que entra el analista es la de la determinación de los requerimientos de información para los usuarios particulares involucrados. Entre las herramientas utilizadas para definir los requerimientos de información en el negocio se encuentran: muestreo e investigación de los datos relevantes, entrevistas, cuestionarios, el comportamiento de los tomadores de decisiones y su ambiente de oficina y hasta la elaboración de prototipos.

En esta fase, el analista está esforzándose por comprender qué información necesitan los usuarios para realizar su trabajo. Se puede ver que varios de los métodos para determinar los requerimientos de información involucran la interacción directa con los usuarios. Esta fase sirve para formar la imagen que el analista tiene de la organización y sus objetivos.

Las personas involucradas en esta fase son los analistas y los usuarios, típicamente los administradores de las operaciones y los trabajadores de las mismas. El analista de sistemas necesita saber los detalles de las funciones actuales del sistema:

Quién	(las personas que están involucradas)
Qué	(la actividad del negocio)
Dónde	(el ambiente donde se lleva a cabo el trabajo)

Cuándo	(en qué momento)
Cómo	(de qué manera se desarrollan los procedimientos actuales)

El analista debe preguntar porqué el negocio usa el sistema actual. Puede haber muy buenas razones para desarrollar el negocio usando los métodos actuales, y deben ser considerados cuando se diseña cualquier sistema nuevo.

3.3 Análisis de las necesidades del sistema.

La siguiente fase que realiza el analista de sistemas involucra el análisis de las necesidades del sistema. Nuevamente, herramientas y técnicas especiales ayudan para que el analista haga las determinaciones de los requerimientos. Una herramienta de éstas es el uso de diagramas de flujo de datos para diagramar la entrada, proceso y salida de las funciones del negocio en forma gráfica estructurada. A partir de los diagramas de flujo de datos se desarrolla un diccionario de datos, que lista todos los conceptos de datos usados en el sistema, así como sus especificaciones, si son alfanuméricos y qué tanto espacio ocupan cuando se imprimen.

Durante esta fase el analista de sistemas también realiza un estudio de las decisiones estructuradas que se hacen. Las decisiones estructuradas son aquellas para las que pueden ser determinadas las condiciones como alternativas de condición, acciones y reglas de acción. Hay tres métodos principales para el análisis de decisiones estructurales: lenguaje estructurado, tablas de decisión y árboles de decisión.

No todas las decisiones de la organización son estructuradas, pero todavía es importante que el analista de sistemas las comprenda. Las decisiones semiestructuradas son sustentadas frecuentemente por los sistemas de apoyo a decisiones. Cuando se analizan decisiones semiestructuradas, el analista examina las decisiones con base en el grado de habilidad para la toma de decisiones requerida, el grado de complejidad del problema y la cantidad de criterios considerados cuando se toma la decisión.

En este punto del ciclo de vida del desarrollo de sistemas, el analista prepara una propuesta de sistema que resume lo que ha sido encontrado, proporciona análisis de costo/beneficio de las alternativas y hace recomendaciones sobre lo que debe ser hecho.

Un análisis de sistemas se lleva a cabo teniendo en cuenta los siguientes objetos en mente:

- Identificar las necesidades del cliente.
- Evaluar qué conceptos tiene el cliente del sistema para establecer su viabilidad.
- Realizar un análisis técnico y económico.
- Asignar funciones al hardware, software, personal, base de datos y otros elementos del sistema.
- Establecer las restricciones del presupuesto y planificación temporal

El aspecto fundamental es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. Los analistas al trabajar con los empleados y administradores, deben

estudiar los procesos de una empresa para reunir detalles relacionados con sus procesos, sus opiniones sobre por qué ocurren las cosas, las soluciones que proponen y sus ideas para cambiar el proceso.

Conforme se reúnen los detalles, los analistas estudian los datos sobre los requerimientos, con la finalidad de identificar las características que debe tener el nuevo sistema, incluyendo la información que debe producir, junto con características operacionales tales como controles de procesamiento, tiempos de respuesta y métodos de entrada y salida.

Cada problema de sistema es único y nunca hay una sola solución correcta. La manera en que se formula una solución o recomendación depende de la capacidad y entrenamiento profesional individual de cada analista.

3.4 Diseño del sistema recomendado.

El diseño de sistemas es el proceso de aplicar ciertas técnicas y principios con el propósito de definir un sistema, con suficiente detalle como para permitir su interpretación y su realización física.

Dentro del diseño es en donde se fomenta la calidad del desarrollo del proyecto, en base a este, los resultados del sistema se obtienen de mejor manera, y en cierta forma, hasta con mayor facilidad para el analista y para el usuario final.

En esta fase del ciclo de vida del desarrollo de sistemas, el analista usa la información recolectada anteriormente para realizar el diseño lógico del sistema de información. El analista diseña procedimientos precisos para la captura de datos, a fin de que los datos que van a entrar al sistema de información sean correctos. Además, el analista también proporciona entradas efectivas para el sistema de información mediante el uso de técnicas para el buen diseño de formas y pantallas.

Parte del diseño lógico del sistema de información es diseñar la interfaz de usuario. La interfaz conecta al usuario con el sistema y es, por lo tanto, extremadamente importante.

La fase de diseño también incluye el diseño de archivos o bases de datos que guardarán la mayor parte de los datos necesarios para los tomadores de decisiones de la organización. Una base de datos bien organizada es la base para todos los sistemas de información. En esta fase el analista también trabaja con los usuarios para diseñar la salida (ya sea en pantalla o impresa) que satisfaga sus necesidades de información.

Por último el analista debe diseñar procedimientos de control y respaldo para proteger al sistema y a los datos y producir paquetes de especificaciones de programa para los programadores. Cada paquete debe contener diseños de entrada y salida, especificaciones de archivos y detalles de procedimiento, y también puede incluir árboles o tablas de decisión, diagramas de flujo de datos, un diagrama de flujo del sistema y los nombres y funciones de cualquiera de las rutinas de código que ya hayan sido escritas.

3.5 Desarrollo y documentación del software.

En esta fase del ciclo de vida del desarrollo de sistemas el analista trabaja con los programadores para desarrollar cualquier software original que se necesite. Algunas de las técnicas estructuradas para el diseño y documentación de software incluyen diagramas estructurados, diagramas de flujo, y pseudocódigo. El analista de sistemas usa uno o más de estos dispositivos para comunicar al programador lo que necesita ser programado.

Durante esta fase, el analista también trabaja con los usuarios para desarrollar documentación efectiva para el software, incluyendo manuales de procedimientos. La documentación le dice al usuario la manera de usar el software y también qué hacer si suceden problemas con el software.

Los programadores tienen un papel principal en esta fase conforme diseñan, codifican y eliminan errores de sintaxis de los programas de computadora, se debe crear el lenguaje de control de trabajos. Para asegurar la calidad, un programador puede realizar ya sea un diseño o un ensayo del código, explicando las partes complejas del programa a un equipo de otros programadores.

3.6 Pruebas y mantenimiento del sistema.

Antes de que pueda ser usado, el sistema de información debe ser probado. Es mucho menos costoso encontrar problemas antes de que el sistema sea entregado a los usuarios. Algunas de las pruebas son realizadas por los programadores solos, y otras por los analistas

de sistemas junto con los programadores. Primero se ejecuta una serie de pruebas para que destaquen los problemas con datos de ejemplo y eventualmente con datos reales del sistema actual.

Durante la fase de pruebas, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir, que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga. Se alimentan como entradas conjuntos de datos de prueba para su procesamiento y después se examinan los resultados. En ocasiones se permite que varios usuarios utilicen el sistema para que los analistas observen si tratan de emplearlo en formas no previstas.

El mantenimiento del sistema y de su documentación comienzan en esta fase y es efectuado rutinariamente a lo largo de la vida del sistema de información. Mucho del trabajo rutinario del programador consiste en el mantenimiento. Muchos de los procedimientos sistemáticos que emplea el analista a lo largo del ciclo de vida del desarrollo del sistema pueden ayudar a asegurar que el mantenimiento se mantenga al mínimo.

3.7 Implementación y evaluación del sistema.

En esta fase del desarrollo del sistema el analista ayuda a implementar el sistema de información. Esto incluye el entrenamiento de los usuarios para que manejen el sistema. Algún entrenamiento es hecho por los proveedores, pero la supervisión del entrenamiento es responsabilidad del analista de sistemas. Adicionalmente, el analista

necesita un plan para una conversión suave del sistema antiguo al nuevo. Este proceso incluye la conversión de archivos de formatos antiguos a nuevos o la construcción de una base de datos, la instalación de equipo y la puesta del nuevo sistema en producción.

Dependiendo del tamaño de la organización que empleará la aplicación y el riesgo asociado con su uso, puede elegirse comenzar la operación del sistema sólo en un área de la empresa (prueba piloto). Algunas veces se deja que los dos sistemas, el viejo y el nuevo, si es que el primero existe, trabajen en forma paralela con la finalidad de comparar los resultados. En otras circunstancias, el viejo sistema deja de utilizarse determinado día para comenzar a emplear el nuevo al día siguiente. Cada estrategia de implantación tiene sus méritos de acuerdo con la situación que se considere dentro de la empresa. Sin importar cuál sea la estrategia utilizada, los encargados de desarrollar el sistema procuran que el uso inicial del sistema se encuentre libre de problemas.

Cuando es realizada una conversión de un sistema antiguo a uno nuevo, puede llevarse acabo mediante alguno de los procesos siguientes:

Proceso encadenado.

En este caso el sistema actual sigue funcionando y produciendo la información establecida, el nuevo sistema toma las transacciones ya procesadas por el sistema actual y produce sus propios resultados en un período de tiempo más adelante.

La información producida por el sistema actual se le distribuye al usuario y la producida por el nuevo sistema queda en poder del equipo de proyecto para su evaluación.

Este método se usa cuando el nuevo sistema es muy complejo o vulnerable y se le exige un alto grado de confiabilidad.

Permite la comparación con los resultados del sistema actual y facilita el análisis de los resultados del nuevo sistema fuera de línea.

Proceso directo.

En este caso se inactiva al sistema actual y las transacciones se procesan en el nuevo sistema.

Este método se usa cuando el sistema de información no es complejo y si las dificultades iniciales esperadas se estiman inferiores a las consecuencias de no comenzar en tiempo el nuevo sistema.

Proceso en paralelo.

Las transacciones se procesan en el sistema actual y en el nuevo sistema simultáneamente.

Por razones de costo, se puede inactivar el sistema actual tan pronto como los resultados del nuevo sistema de información sean confiables.

Este método se usa cuando la gerencia, exige una alta confiabilidad del nuevo sistema o cuando los sistemas son muy diferentes.

Tiene la ventaja de minimizar el riesgo de comenzar con un sistema defectuoso.

Proceso por subsistemas.

El sistema actual continúa procesando una parte de las transacciones y el nuevo sistema el resto.

La secuencia en la que los subsistemas comenzarán su operación debe ser cuidadosamente planificada.

Reduce costos y riesgos.

En cualquier caso deben ser modificados los programas, y el analista puede tener que cambiar algunos de los materiales del diseño del sistema. A su vez, esto puede necesitar que el analista se reúna con el usuario y vuelva a investigar cómo funciona una actividad específica del negocio.

Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo, las organizaciones y los usuarios cambian con el paso del tiempo, incluso el ambiente es diferente con el paso de las semanas y los meses. Por consiguiente, es indudable que debe darse mantenimiento a las aplicaciones, realizar cambios y modificaciones en el software, archivos o procedimientos para satisfacer las nuevas necesidades de los usuarios.

La evaluación se muestra como parte de esta fase final de ciclo de vida del desarrollo del sistema, principalmente para efectos de discusión. De hecho, la evaluación se realiza durante cada fase. Un criterio principal que debe ser satisfecho es si los usuarios pretendidos ya están usando el sistema.

Debe hacerse notar que a veces los sistemas trabajan en forma cíclica. Cuando un analista termina una fase del desarrollo del sistema y pasa a la siguiente, el descubrimiento de un problema puede obligar a que el analista regrese a la fase anterior y modifique el trabajo que se hizo con anterioridad.

3.8 Evaluación.

La evaluación ocurre a lo largo de las siguientes dimensiones:

Evaluación operacional. Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.

Impacto organizacional. Identificación y medición de los beneficios para la organización, en áreas tales como finanzas (costos, ingresos y ganancias), eficiencia operacional e impacto competitivo. También se incluye en impacto sobre el flujo de información interno y externo.

Opinión de los administradores. Evaluación de las actitudes de directivos y administradores dentro de la organización, así como de los usuarios finales.

Desempeño del desarrollo. La evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También se incluye al valoración de los métodos y herramientas utilizadas en el desarrollo.

Después de que el sistema esta instalado se le debe dar mantenimiento, esto significa que los programas de computadora deben ser modificados y mantenidos actualizados.



CAPÍTULO IV
ANÁLISIS DE REQUERIMIENTOS DEL DISEÑO.
DIAGNÓSTICO

METODOLOGÍAS DE TRABAJO Y EL ANÁLISIS DE
REQUERIMIENTOS.



Introducción.

En el presente capítulo, se explica la metodología de trabajo utilizada para el modelaje del sistema de información DLT Card, siendo está, el Lenguaje de Modelado Unificado (UML) por sus siglas en inglés, haciendo referencia a la necesidad que se tiene en la actualidad de hacer sistemas más estructurados, mucho mejor organizados, modularizados, es decir, hechos por partes independientes y mejor realizados. Cabe mencionar que en el presente capítulo solo se hace una revisión general de las principales características de dicha metodología, y cuales de éstas características fueron útiles y aplicables para el caso en estudio, ya que el profundizar en un tema tan amplio como lo es el UML no es objetivo del presente capítulo. Aunado a esto se hacen mención en este capítulo del análisis de requerimientos que se tuvo que levantar para el modelado del sistema en estudio.

Actualmente, para soportar la puesta en operación de nuevos productos y ofrecer un mejor servicio a los clientes, se requiere que toda organización cuente con una metodología de trabajo que permita una adecuada interacción de las áreas que intervienen en los procesos para el desarrollo y mantenimiento de aplicaciones.

4.1 Características del modelado de sistemas en la actualidad.

En un principio el desarrollo de software se hacía sin usar metodologías o técnicas, esto debido a que los sistemas eran pequeños y de muy escasa complejidad, pero a medida que avanzó la tecnología, las necesidades crecieron provocando que los sistemas de software no fueran de suficiente calidad.

Fundamentalmente, un buen sistema (o de alta calidad) es aquel que cumple las necesidades del usuario. Es decir, tiene que ser:

- Útil y aprovechable. Un buen software hace la vida de los usuarios más fácil o mejor.
- Fiable. Un buen software tiene pocos errores.
- Flexible. Las necesidades de los usuarios cambian a lo largo del tiempo, incluso mientras el software se está desarrollando, de manera que es importante poder realizar cambios en el mismo más tarde.
- Accesible. Tanto para la compra como para el mantenimiento.
- Disponible. De otro modo no importa lo bueno que sea, aquí se consideran dos aspectos:
 - El software tiene que poder ejecutarse en el hardware disponible, con el sistema operativo disponible, etc.
 - El software debe ser lo primero que exista de forma que un proyecto de software debe completarse con éxito y entregar el software prometido.

A lo largo de las últimas décadas se ha alcanzado un conocimiento cada vez mas profundo de que sistemas son los más propensos a ser correctos. El problema fundamental es que hay un límite de cuanto pueda manejar del sistema, un solo analista.

Las metodologías de trabajo que se seguían cuando los sistemas eran pequeños, podían construirse mediante técnicas heroicas de programación donde una única persona intenta tener en mente todos los aspectos relevantes del sistema, para un desarrollador o encargado

de mantenimiento, comprender todo lo relacionado con el sistema en general es imposible. De hecho, no existían metodologías establecidas para el desarrollo de sistemas de información, en muchas ocasiones se recurría a conocimientos empíricos de estructuración de líneas de código, sin tener la menor idea de que después, era una labor titánica tratar de hacer correcciones a dicho sistema, ya que no se tenía conocimiento de cómo detectar errores o como depurar procesos lógicos.

Esto quiere decir que es fundamental, emprender una tarea de desarrollo o mantenimiento sin conocer todo el sistema.

El siguiente paso es pensar en un sistema como conjunto de módulos e identificar dependencias entre módulos. En el más amplio sentido de la palabra, un módulo podría ser cualquier elemento identificable del sistema y que tiene sentido por sí mismo. Por ejemplo, las áreas que engloban a una gasolinera:

- Clientes.
- Chóferes.
- Automóviles.
- Combustibles.
- Facturación.
- Reportes.

Por mencionar algunos.

Por supuesto no todos los módulos son iguales, tomar un programa monolítico y separarlo de forma aleatoria en archivos no es

óptimo, es necesario considerar que módulos característicos se debería tener para llevar a cabo el desarrollo y el mantenimiento del sistema de la manera más sencilla, barata y fiable posible. Se deben tener en cuenta los conceptos asociados de **dependencia, acoplamiento, cohesión, interfaz, encapsulación y abstracción**. Una vez identificado lo que es un buen módulo, se puede contemplar la reutilización de éste como componente.

Dependencia y acoplamiento: El módulo A depende del módulo B si cualquier cambio en el módulo B implica que el módulo A también tenga que ser modificado. A veces se dice que el módulo A es un cliente del módulo B, o que el módulo B actúa como servidor del módulo A. Sin embargo los términos cliente y servidor se utilizan también de manera más específica en una arquitectura cliente-servidor por ejemplo, el cliente y el servidor son generalmente procesos separados en máquinas diferentes. En general es normal que un mismo módulo sea tanto cliente como servidor. Esto significa, que depende de algunos módulos, mientras que otros módulos dependen de él. Incluso es posible que un par de módulos se tengan el uno al otro de cliente.

La dependencia a veces se conoce como acoplamiento. Un sistema con muchas dependencias tiene fuerte acoplamiento. Los buenos sistemas tienen débil acoplamiento, porque en ese caso los cambios en una parte del sistema son menos probables de propagarse a través del sistema.

Entre otras cosas se pretende minimizar el número de casos en los que un cambio en un módulo necesite un cambio en otro módulo.

Esto significa que hay que saber qué cambios dentro de un módulo pueden afectar al resto del sistema. Sin embargo, para poder aprovecharse del débil acoplamiento de un sistema es muy importante poder identificar qué módulos están acoplados; por otra parte existe la posibilidad que gastar esfuerzo comprobando si se necesitan los cambios en un módulo, lo cual es caro incluso si la conclusión es que no se necesitan dichos cambios. Lo ideal es saber con certeza qué módulos de un sistema podrían verse afectados por un cambio en un módulo dado. En resumen, una vez que se han establecido los límites entre los módulos del sistema, hay dos tipos de información que puede ser útil:

1. Qué servicios va a proporcionar el módulo, saber qué tipo de cambios en un módulo pueden ser peligrosos.
2. Qué módulos se tienen que cambiar, si se realiza un cambio peligroso en un módulo.

Interfaz: Una interfaz de un módulo, define algunas características de dicho módulo de las que sus clientes pueden depender. El resto del sistema sólo puede utilizar el módulo de las maneras permitidas por la interfaz; esto es, una interfaz encapsula conocimiento sobre el módulo. Las conexiones entre módulos son las suposiciones que los módulos hacen unos de otros. Cualquier suposición que un cliente hace sobre un servidor corre el riesgo de ser incorrecta; por lo que se debería documentar tales suposiciones en las interfaces y comprobar su corrección. Si se documentan todas las suposiciones en la interfaz con éxito se podrá decir: si un módulo cambia internamente sin cambiar su interfaz, dicho cambio no

necesitará que se realice ningún otro cambio en ninguna otra parte del sistema.

Hay varias razones para querer saber no sólo las dependencias que podrían existir, esto es, qué características están documentadas en las interfaces de los módulos del sistema, sino qué dependencias existen realmente. Se ha mencionado la situación en la que se realiza un cambio a un módulo que podría afectar a sus clientes; sus clientes son por definición los módulos que podrían cambiar, por lo que es importante poder decir cuáles son. Después supóngase que se considera reutilizar un módulo. Se necesita saber no sólo los servicios que van a proporcionar (cual es la interfaz) sino qué servicios requiere para funcionar. Los servicios que requiere un módulo a veces se denominan dependencias de su contexto. Se pueden expresar en función de interfaces; el módulo podría garantizar que si tiene interfaces seguras, entonces por turnos, se proporcionará su propia interfaz. Entre ellas, las dependencias de contexto de un módulo y la interfaz propia del módulo constituyen un contrato que describe las responsabilidades del módulo. Si el contexto proporciona lo que el módulo necesita, entonces éste garantiza proporcionar los servicios descritos en su interfaz. Un buen sistema está formado por módulos encapsulados.

Los módulos correctos a menudo tienen la propiedad de que sus interfaces proporcionan una abstracción de algún elemento conocido de manera intuitiva que puede, no obstante, ser difícil de implementar. Este tipo de módulos se dice que tienen una fuerte cohesión.

La interfaz se abstrae de las cosas que el desarrollador no tiene que conocer para utilizar el módulo, dejando una fotografía ordenada y coherente de lo que el usuario del módulo quiere conocer. El módulo realiza un conjunto coherente de cosas, pero dentro de lo posible el desarrollador del cliente está protegido de la información irrelevante relativa a cómo el módulo hace lo que hace. Este asunto de permitir al desarrollador concentrarse en lo esencial es sutilmente diferente del asunto de encapsulación para alcanzar el débil acoplamiento, que está implicado en la prevención del desarrollador de utilizar información oculta.

Abstracción: Es cuando un cliente de un módulo no necesita saber más de lo que hay en la interfaz.

Encapsulación: Es cuando un cliente de un módulo no es capaz de saber más de lo que hay en la interfaz.

Si un módulo, de cualquier tamaño y complejidad, es una buena abstracción (tiene fuerte cohesión y débil acoplamiento) puede ser factible reutilizarlo en sistemas posteriores, o sustituirlo en el sistema existente. Esto es, puede ser posible considerarlo como un componente conectable. Sin embargo, aunque esto es posible, también depende de la arquitectura en la que se desarrolla el componente y en la que vaya a ser utilizada. Esto lleva a pensar en el desarrollo basado en componentes centrado en la arquitectura.

Componente: Es una unidad de reutilización y sustitución. Un componente puede ser un módulo con propiedades que lo hacen reutilizable y sustituible. Aunque existen diferentes definiciones de que

es un componente, y son caso de atención en varios capítulos de libros consultados para el desarrollo de la presente investigación, el asunto es que es más fácil reemplazar un módulo si se puede hacer sin necesidad de recompilar el sistema.

Todo esto, es necesario conocerlo, ya que la finalidad es que se tengan sistemas grandes, con módulos o componentes tan independientes pero a la vez reutilizables dentro del sistema.

Cohesión: En un módulo en el cual sus interfaces proporcionan una abstracción de algún elemento conocido de manera intuitiva que puede, no obstante, ser difícil de implementar, se dice que el módulo tiene una fuerte cohesión.

4.2 Modelado de sistemas basado en UML.

Bajo este contexto no es de sorprender que un módulo reutilizable sea bueno si tiene fuerte cohesión, débil acoplamiento con el resto del sistema, una interfaz bien definida y es una abstracción encapsulada de un elemento bien comprendido. Lo que puede ser más sorprendente es que un módulo es reutilizable dependiendo del contexto en el que sea desarrollado y en el que se vaya a utilizar.

La manera ideal de construir un nuevo sistema es tomar algunos componentes existentes y conectarlos unos con otros. Por supuesto la conectividad es la propiedad que permite hacer esto. La metáfora sugiere correctamente que esto es sólo en parte una propiedad de los elementos conectados. Los componentes tienen que ser elementos que completan las necesidades del sistema en su totalidad. Más que esto,

tienen que ser compatibles unos con otros, y esto depende de la presencia de una arquitectura adecuada.

El desarrollo está basado en la arquitectura y en componentes si se le da máxima prioridad al seguimiento (y toma) de decisiones de arquitectura correctas, y a la utilización y desarrollo de componentes correctos.

Después de la recolección de varios autores, la construcción de un buen sistema está basado en analogías de artefactos usados en ramas como la ingeniería, una aproximación técnica:

- Utiliza un proceso definido con fases claras, cada una de las cuales tienen un producto final.
- Esta relacionado con encontrar un claro conjunto de requisitos.
- Considera los formularios de verificación y validación.
- Utiliza un almacenamiento de conocimientos, arquitecturas y componentes relevantes.
- Hace un uso coherente de las herramientas.

Se ha identificado ya que los sistemas de software utilizan una arquitectura modular reutilizando componentes, tal y como hacen los artefactos de ingeniería en la construcción de un puente o de un motor.

Es posible imaginarse que en proyectos pequeños, simplemente se sienta el programador y realiza el trabajo. Sin embargo, para sistemas más complejos es necesario dirigir el proceso de desarrollo.

Es posible planificar el desarrollo y decir hasta dónde se ha llegado y si se va con retraso. Se debe tener un juego de herramientas de técnicas apropiadas para desarrollar un sistema. Que debe tener una manera comprensible de documentar y controlar lo que se hace.

Un método o metodología normalmente especifica qué lenguaje de modelado debería utilizarse en la descripción del trabajo de análisis y de diseño. También indica cómo se deberían producir estas cosas, por ejemplo, dando un conjunto de pasos a seguir para capturar los requisitos de los usuarios (analizado posteriormente).

Un modelo es una representación abstracta de una especificación, un diseño o un sistema, desde un punto de vista particular. A menudo se representa gráficamente mediante uno o más diagramas. Tiene como objetivo expresar la esencia de algunos aspectos de lo que se está haciendo, sin especificar detalles innecesarios. Su propósito es permitir a la gente implicada en el desarrollo pensar y discutir sobre problemas y soluciones sin desviarse de los objetivos. Si esto es útil, un modelo tiene que tener significado preciso y bien entendido.

Un lenguaje modelado es una manera de expresar los distintos modelos que se producen en el proceso de desarrollo. Un lenguaje de modelado define una colección de elementos del modelo, que son aproximadamente análogos a la pronunciación en el lenguaje hablado; un modelo está formado por elementos del modelo, tal y como una sentencia está formada por palabras. UML, Lenguaje Unificado del Modelado, es, por supuesto, un ejemplo de lenguaje de modelado. Un

lenguaje de modelado normalmente se centra en diagramas, pero puede utilizar texto. Tiene:

- Sintaxis, en un lenguaje de modelado basado en diagramas, las reglas que determinan qué diagramas son legales;
- Semántica, las normas que determinan qué significa un diagrama correcto.

Tanto la sintaxis como la semántica puede ser más o menos formales. La mayoría de lenguajes de modelado expresan tanto la sintaxis como la semántica de manera informal con el español coloquial.

Además de utilizarlo para apoyar el desarrollo de un sistema particular, un lenguaje de modelado puede utilizarse para documentar un artefacto reutilizable como es un componente o un marco de trabajo.

Dado que los desarrolladores necesitan un lenguaje de modelado para ayudarles a discutir los problemas y soluciones implicados en la construcción del sistema, el lenguaje elegido deberá ser suficientemente expresivo, de manera que sea posible expresar los aspectos del diseño que será necesario tratar, y que reflejen de forma que tengan sentido, los cambios en el diseño que se lleven a cabo durante el desarrollo como cambios en el modelo; suficientemente fácil de utilizar, de forma que el lenguaje de modelado ayude a tener un conocimiento claro en vez de proporcionar el camino para tener dicho conocimiento claro; inequívoco, para que el lenguaje de modelado

ayude a resolver malos entendidos en vez de presentar más; soportado por herramientas adecuadas, de manera que el esfuerzo de los desarrolladores, pueda utilizarse en un trabajo que requiera su habilidad, no en un trabajo rutinario como crear un diagrama con una herramienta de dibujo; Generalmente utilizado, por gran variedad de razones.

Por supuesto cuanto más general sea la utilización de un lenguaje es más probable que se cumplan los cuatro puntos anteriores; cuando se incorpora gente nueva en el proyecto, es una ventaja si ya conocen el lenguaje de modelado en vez de tener que aprenderlo; para hacer diseño basado en componentes hay que ser capaz de leer las descripciones de los componentes, y cuanto más rápido y fácil se pueda hacer es más barato tener en cuenta un componente. Cuanto más general sea la utilización de su lenguaje de modelado, mayor es la posibilidad de que sea el mismo que el escritor del componente decidió utilizar.

Existe un fuerte acoplamiento significativo entre los asuntos del proceso de desarrollo y los de un sistema de gestión de calidad. El objetivo último de ambos es asegurar que el proceso (y desde aquí, se espera, el producto) tenga alta calidad. En general, el proceso de desarrollo especificará aspectos más técnicos del proceso y del sistema de gestión de calidad, los más administrativos.

Cabe recalcar que ningún proceso o metodología, aunque sean buenos, puede asegurar el éxito de un proyecto. Lo más que puede hacer es apoyar a la gente implicada en el proyecto para producir buenos resultados.

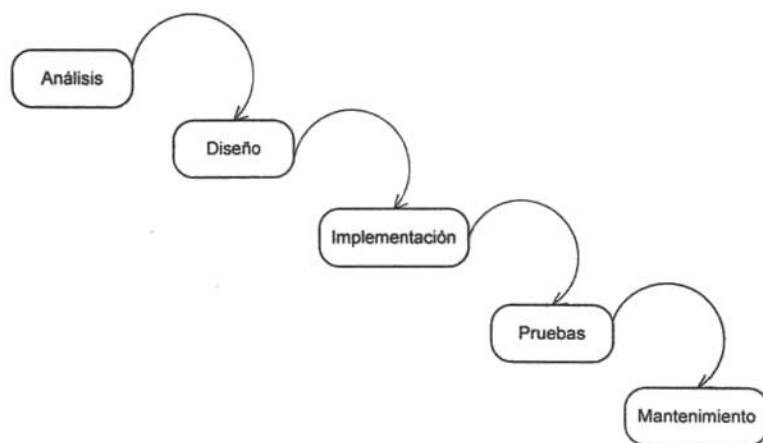


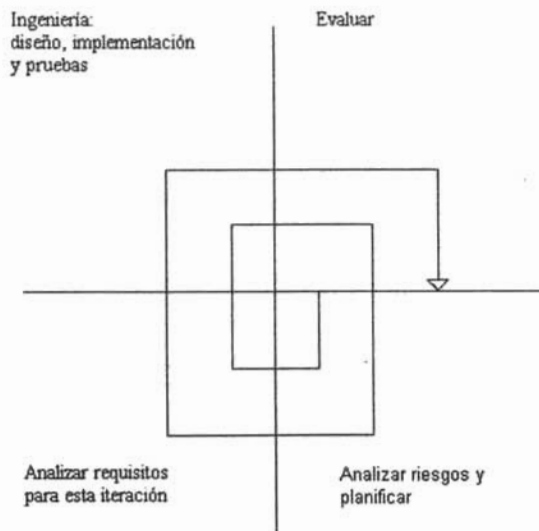
Fig. 4.1 Proceso en cascada

El famoso pero desacreditado, proceso en cascada, mostrado en la figura anterior, tiene un número relativamente pequeño de partes identificables, en este caso las cinco fases del ciclo de vida. Una división en partes es algo fundamental de cualquier proceso, debido a que el proceso pretende ayudar a los hombres a entender una cosa. En este caso el proceso se divide en partes en función de las actividades que se están ejecutando; sin embargo, hay también una consecuencia y es que las actividades se realizan una detrás de otra, transcurriendo el tiempo hacia abajo.

Este proceso incorpora una suposición (restricción) en la que una vez que se supone que ha terminado una fase, nunca se vuelve a ella. Si fuera posible tomar decisiones perfectas todo el tiempo, las cosas funcionarían. Desafortunadamente es imposible, por ejemplo, debido a que constantemente se añade nueva información. En la práctica, es

necesario revisar las primeras decisiones en función de la experiencia: por ejemplo, revisar la especificación de los requisitos cuando se descubre que al cliente no le gusta el sistema implementado. El rechazar la revisión de las decisiones pueden terminar fácilmente en un abandono del proyecto.

Hoy en día, sin embargo, se reconoce que para casi todos los sistemas es correcto y necesario tener algún tipo de proceso iterativo. Los procesos de desarrollo modernos tienen como elemento fundamental las iteraciones, y tratan de proporcionar maneras de gestionar, en vez de ignorar, los riesgos.



4.2 Proceso Boehm

El proceso en espiral de Boehm incorpora estas dos ideas, y desde entonces han aparecido muchas variantes del proceso en espiral. Una variante sencilla se ilustra en la figura anterior. Empezando por el centro de la espiral, un proyecto que siga el proceso

atraviesa sucesivos análisis y planificaciones de riesgos, análisis de requisitos, fases de ingeniería y evaluación. La fase de ingeniería del proceso en espiral implica diseño, implementación y pruebas. El número de iteraciones es arbitrario.

4.3 Análisis de requerimientos.

La parte más difícil de cualquier proyecto de diseño es comprender la tarea que se está tratando, en un principio, el principal requerimiento hacia el área de sistemas por parte de los gerentes de las estaciones de servicio fue: Elaborar un sistema para controlar cargas de combustible y evitar despachos ficticios.

La especificación del problema dado antes es muy incierta, pero es la típica solicitud que se recibe al principio, cuando se plantea por primera vez un proyecto. Antes incluso de ponerse de acuerdo en abordar el diseño, es necesario un análisis detallado de los requerimientos de los usuarios. Esta tarea de ingeniería de requisitos es compleja por una gran variedad de razones:

- Usuarios diferentes tendrán distintas prioridades, a veces en conflicto.
- Los usuarios no suelen tener una visión clara y fácilmente expresable de lo que quieren: por ejemplo, les resulta difícil distinguir entre lo que hace un sistema existente y lo que tiene que hacer un sistema adecuado.
- Es difícil imaginar un sistema del cual sólo se ha visto su descripción, por lo que los usuarios pueden pensar que la descripción es correcta cuando en realidad falta algo vital.

- Los responsables que se encargan de hablar con la mayoría de los desarrolladores pueden no tener experiencia directa en hacer el trabajo que hacen los usuarios del sistema.

Después de algunas investigaciones esmeradas, surgieron los siguientes hechos sobre los requisitos que debería satisfacer un buen sistema de control de despachos para una estación de servicio.

- Sistema para controlar las ventas de combustible.
- Manejo de reportes vía Internet. Acceso a la información en línea.
- Automatización de manejo de clientes de crédito.
- Eliminar malos manejos en los despachos a unidades de clientes de crédito
- Verificación de que a la unidad que se despache combustible sea realmente la asignada (identificación de unidades)
- Personalización de unidades para el despacho (asignación de una tarjeta por unidad)
- Ofrecimiento a clientes de control de sus flotillas de unidades para el despacho de combustible.
- Surtir combustible a la unidad especificada cantidades, producto, días y horarios autorizados o asignados por los dueños de las flotillas, evitando malos manejos del despacho, como lo pudiesen ser, despachos a unidades no miembros de una flotilla.

- Sistema ajustable a las necesidades del cliente (reportes por día, unidad, centro de costos, combustibles autorizados, etc)
- Control de odómetros (rendimiento) para evitar que los chóferes de las unidades autorizadas, robaran el combustible de las unidades sin haber recorrido ninguna ruta.
- Identificación de los vehículos.
- Histórico de despachos realizados para auditorias internas de los propios clientes.
- Sistema seguro, que solo el cliente pudiera revisar sus historiales en línea con un usuario y contraseña únicos.
- Despacho solamente de la cantidad autorizada al chofer, esto ligando el autorizador al dispensario de gasolina (evitando despachos mayores).
- Posibilidad de realizar despachos en cualquier estación de servicio miembro del grupo, centralizando saldos y autorizaciones en un lugar central de autorización.
- Seguridad en tarjetas, evitando la fácil clonación y hacer mal uso de este recurso.

Para que un sistema sea de alta calidad, tiene que cumplir las necesidades de sus usuarios. De manera que se realizará una aproximación al análisis de los sistemas orientada al usuario. Hay que identificar los usuarios del sistema y las tareas que deben acometer con el sistema. También se debe de buscar información sobre qué tareas son las más importantes, de manera que se pueda planificar el desarrollo de acuerdo a esto.

Los usuarios y tareas en UML, en realidad se utilizan como términos técnicos actores y casos de uso

Un caso de uso es una tarea que un actor necesita ejecutar con la ayuda del sistema, como lo es Pasar Datos del Despacho o Mandar Despacho a estación.

Un caso de uso se representa con una elipse, denota un requerimiento solucionado por el sistema. Cada caso de uso es una operación completa desarrollada por los actores y por el sistema en un diálogo. El conjunto de casos de uso representa en su totalidad de operaciones desarrolladas por el sistema. Va acompañado de un nombre significativo. En el caso de sistema DLT Card se tienen como casos de uso despachar, emitir comprobante, verificar saldo y la autorización manual del despacho, lo que comúnmente se denomina dentro del sistema asignación.

. Un actor es un usuario del sistema que tiene un rol particular. En realidad un actor puede ser un sistema externo que es como un usuario desde el punto de vista de nuestro sistema: el punto crucial es que se está diseñando alguien o algo externo al sistema, que interactúa con nuestro sistema y que puede hacerle peticiones. Por ejemplo, el sistema que aquí interesa tendrá un actor llamado despachador de gasolina representando a la persona que interactúa con el sistema para realizar un despacho o para una asignación de una carga. Se representa mediante un



Despachador de Gasolina

acompañado de un nombre significativo, si es necesario, como en este caso Despachador de Gasolina, o en su caso, si es un sistema externo que interactua con el sistema mismo, se le denomina con el nombre del sistema como lo es en el ejemplo anterior.

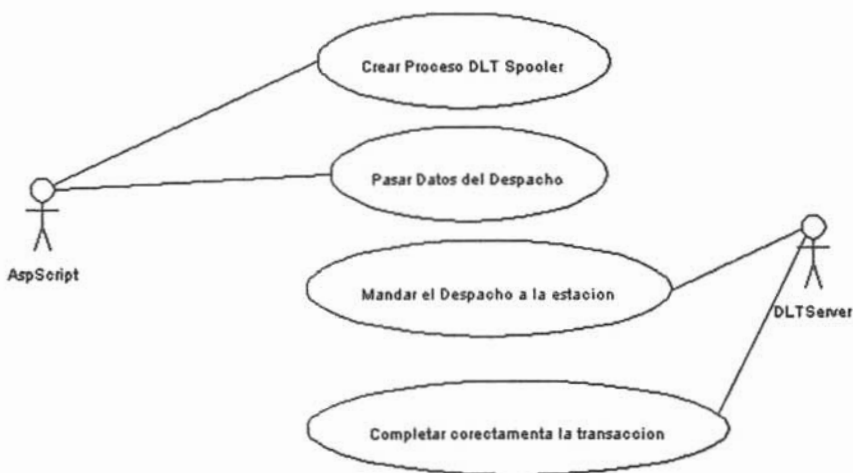


Fig. 4.3 Casos de uso modelados para el sistema DLT Card.

Entre los elementos de un diagrama de casos de uso se pueden presentar tres tipos de relaciones, representadas por líneas dirigidas entre ellos (del elemento dependiente al independiente) comunica

(comunicantes). Relación entre un actor y un caso de uso, denota la participación del actor en el caso de uso determinado. En la figura 4.3 todas las líneas que salen del actor denotan este tipo de relación.

Usa (uses). Relación entre dos casos de uso, denota la inclusión del comportamiento de un escenario en otro. En el caso del ejemplo el caso de uso Cancelar incluye en su comportamiento DarVueltas y PedirProducto incluye también DarVueltas.

Extiende (extends). Relación entre dos casos de uso, denota cuando un caso de uso es una especificación de otro. Por ejemplo, podría tenerse un caso de uso que extienda la forma de pedir azúcar, se parte de que permita escoger el tipo de azúcar (normal, dietético, etc) y además la cantidad en las unidades adecuadas para cada caso (cucharadas, bolsas, etc).

Cualquier proceso de desarrollo pretende producir, probablemente después de varias iteraciones, la implementación de un sistema. Esto es un programa o un conjunto de programas que funcionan en un entorno apropiado para cumplir las necesidades de los usuarios, incluyendo necesidades implícitas tales como el mantenimiento. El diseño y la arquitectura del sistema incorporan las decisiones importantes sobre cómo construir el sistema, abstrayéndose de muchos detalles. Un lenguaje para describir un diseño debería, naturalmente, estar basado en diagramas. Es inconcebible que un único diagrama pueda capturar todo sobre nuestro diseño, y está claro que no debería ser deseable, ya que en cada momento interesa diferentes aspectos del diseño.

Un modelo de casos de uso describe el sistema requerido desde el punto de vista del usuario, muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras operaciones).

El diagrama no es el diseño: el diagrama es una representación de un modelo de diseño, que captura un aspecto del diseño de una manera que pueda ser discutido.

En UML, los diagramas de clases se utilizan para documentar la estructura estática del sistema; esto es, qué clases hay y cómo están relacionadas, pero no cómo interactúan para alcanzar comportamientos particulares. Un diagrama de clases puede también mostrar otros aspectos de la estructura estática, el conjunto de clases y objetos importantes que hacen parte de un sistema. Muestra de una manera estática la estructura de información del sistema y la visibilidad que tiene cada una de las clases, dada por sus relaciones con las demás en el modelo.

Es importante recordar que los objetos son realmente cosas dentro de un programa de computadora; que cuando se habla sobre libros y copias, por ejemplo, realmente nos referimos a la representación de estas cosas dentro de nuestro sistema.

Para el sistema en análisis, las clases que se pueden generar de los requerimientos solicitados por los usuarios, pueden ser, por enumerar algunas, los siguientes ejemplos:

- Estación
- Transacción
- Despacho
- Sesión
- Trama

Etc.

Una clase describe un conjunto de objetos con un rol o roles equivalentes en un sistema.

Una clase es representada por un rectángulo con tres divisiones internas, son los elementos fundamentales del diagrama. Una clase describe un conjunto de objetos con características y comportamiento idénticos.

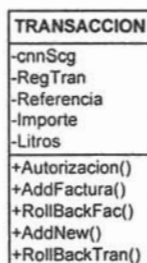


Fig. 4.4 Clase transacción del sistema DLT Card.

Los tres compartimientos estándares alojan el nombre de la clase, sus atributos y sus mensajes, respectivamente.

Los atributos identifican las características propias de cada clase. Generalmente son de tipos simples, ya que los atributos de tipos compuestos se representan mediante asociaciones de composición con

otras clases. El conjunto de operaciones que describen el comportamiento de los objetos de una clase. Una asociación en general es una línea que une dos puntos o más símbolos. Pueden tener varios tipos de adornos, que definen su semántica y características.

En la figura que aparece a continuación, se diagraman algunas clases utilizadas en el desarrollo del sistema DLT Card, llevándose a cabo una descripción más detallada en los capítulos posteriores.

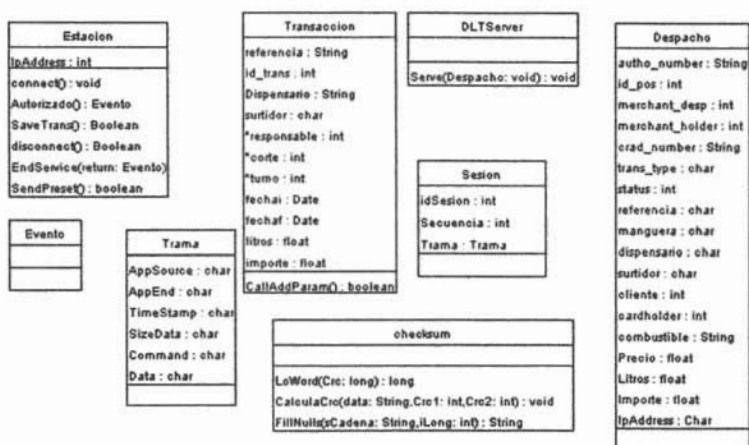


Fig. 4.5 Diagrama de clases: estructura estática de algunas clases utilizadas en el sistema DLT Card.

En esta figura se puede observar la representación de algunos requerimientos que el usuario hizo con anterioridad, ya diagramados, incluyendo las características que tiene cada clase, por ejemplo, existe una clase llamada Sesión la cual como propiedades cuenta con un idSesion, secuencia y trama, características que tiene que tener

cualquier inicio de sesión en un sistema de cómputo seguro, o que por lo menos sea diseñado de esa manera, una clase llamada Despacho, la cual como características o propiedades tiene un número de tarjeta, un dispensario, surtidor, chofer, etc, características enumeradas en el análisis de requerimientos hecho a los usuarios del sistema.

Un diagrama de secuencia muestra los objetos y actores que participan en una colaboración encima de las líneas de punto. La línea representa el tiempo visto por el objeto: es la línea de vida del objeto. Se supone que el tiempo pasa según nos movemos de arriba abajo en el diagrama. Un mensaje aparece como un flecha desde la línea de vida del emisor hasta la línea de vida del receptor.

No importa el orden en que aparecen los objetos, sin embargo el diagrama será más legible si se colocan los objetos que participan antes lo más a la izquierda posible, de manera que la mayoría de los mensajes fluyen de izquierda a derecha.

Cuando un objeto tiene una activación existente, se muestra un rectángulo estrecho que cubre su línea de vida. Opcionalmente se pueden sombrear las partes de la activación en las que el objeto está calculando realmente. De nuevo opcionalmente se puede indicar cuándo tienen lugar las respuestas a los mensajes.

A continuación se muestra un ejemplo de diagrama de secuencia, que da detalle al caso de uso asignación del sistema DLT Card, dicho diagrama será explicado a detalle en el siguiente capítulo.

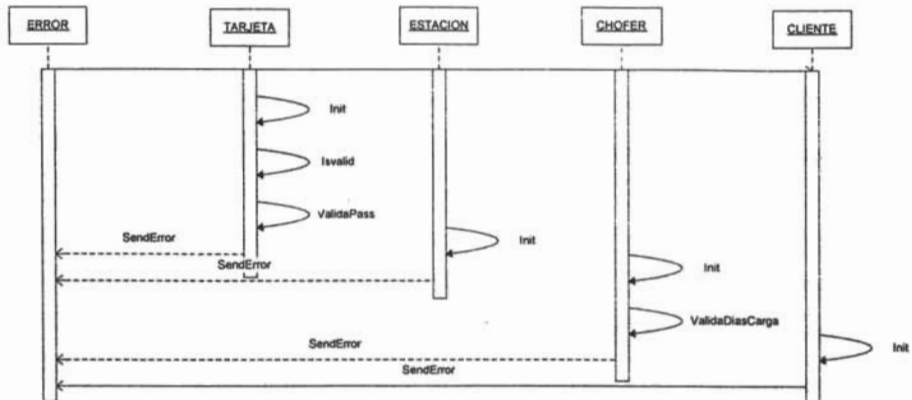


Fig. 4.6 Diagrama de secuencia

Un objeto se representa como una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal que denotan la ejecución de métodos. El rectángulo de encabezado contiene el nombre del objeto y el de su clase. La activación muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por sí mismo o por medio de delegación a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto. El envío de mensajes entre los objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta.

Es de esta manera como, de una lista que pudiese ser interminable de requerimientos hechos por los futuros usuarios del sistema, se comienza a hacer el estudio de las necesidades reales y la interrelación entre cada caso de estudio y como por medio de un diagrama, se puede visualizar los diferentes requisitos como un todo, esquematizado en partes, de manera ordenada y sencilla.

De ahí la importancia del UML, los lenguajes de modelado vienen a aclarar la visión de los diseñadores, analistas, programadores y todo aquel individuo relacionado con un sistema grande, así por medio de módulos y esquemas dibujados, no es necesario, como se apuntaba en un principio del presente, que una sola persona conozca todo el sistema o que todos los involucrados conozcan en su totalidad el sistema, es suficiente que la persona asignada a la elaboración de un módulo tenga el conocimiento de sus propiedades, sus relaciones y su resultado final que debe arrojar al sistema informático.

El Lenguaje Unificado de Modelado (UML) es un tema mucho más profundo, que no es caso de estudio para la presente memoria, pero si es indispensable contar con las bases teóricas para poder entender posteriormente en el desarrollo del sistema, que clases y casos de uso son utilizados y que actores intervinieron en el modelaje del sistema.



CAPÍTULO V
PROPUESTA DE SOLUCIÓN.



En el presente capítulo se explican, los módulos del sistema DLT Card, la plataforma sobre la cual se desarrolló el sistema, lenguajes utilizados y el modelaje que se realizó, para poder llegar al desarrollo.

De esta manera, la propuesta de solución que se presentó para resolver las necesidades de las estaciones de servicios, fue la creación de un sistema de tarjetas inteligentes que contarán con protección mediante un NIP para cada cliente, protección de la misma tarjeta mediante algoritmos de encriptación de un solo paso para evitar la clonación, tarjetas personalizadas, ticket de consumo indicando sucursal, titular de la tarjeta, saldo disponible, litros consumidos, tipo de combustible despachado y hora de despacho, conexiones seguras vía Internet así como dispensarios de gasolina asignados a este servicio, garantizando así que la cantidad marcada para despachar a cierta tarjeta, sea realmente la despachada a la unidad indicada, y toda esta información disponible en línea.

Con dicho sistema se elimina el uso de efectivo, se puede despachar en cualquier gasolinera del grupo, existe la garantía de que los litros cobrados sean realmente los litros despachados, se realiza la consulta de información vía Internet, las restricciones de consumo por día, el número máximo de despachos al día, por combustible y por litros, la captura de kilometraje y la ruta de la unidad así como la agrupación de los chóferes por centros de costos, ayudando a la administración de dichos choferes.

En la figura 5.1 se muestran los casos de uso generales para el sistema, los actores que intervienen en el sistema y las acciones que realizan con este.

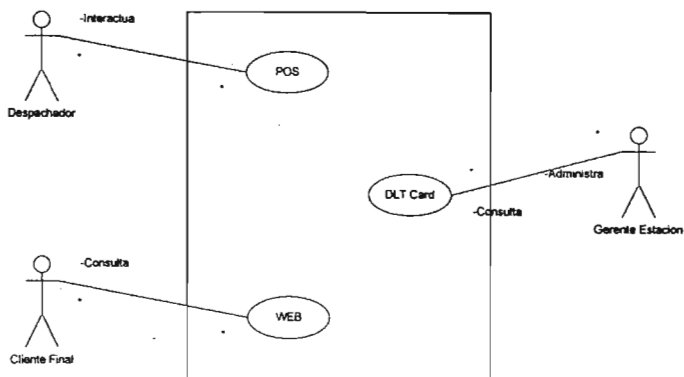
Caso de uso general para el sistema DLT
Card

Fig. 5.1 Casos de uso general para el sistema DLT Card

El sistema está compuesto de tres módulos, el actor despachador interactúa con el módulo denominado P.O.S. que es el encargado de hacer validaciones y de despachar. El actor cliente final, consulta en el módulo denominado WEB los movimientos realizados por sus choferes y por último el actor Gerente de estación, es quien administra el sistema desde la estación de servicio, dando de alta clientes, asignando choferes y tarjetas, entre otras actividades

En los subtemas siguientes basándose en la teoría expuesta en capítulos anteriores se realiza la explicación de diseño, programación, implantación y administración de cada uno de los módulos de los que cuenta el sistema DLT Card.

5.1 POS (Point of Sale) Punto de venta

Las terminales disponibles en el piso de venta para el sistema DLT Card son de marca Verifone modelo 3750 y modelo 3600 las cuales son las últimas innovaciones del mercado, ambas terminales están basadas en programación WML, cuentan con navegador WML así como interfaces ethernet, siendo la terminal 3600 inalámbrica, permitiendo esto hacer el despacho de combustible desde la unidad que así lo solicite.



La Terminal 3750 es un equipo fijo que realiza transacciones a alta velocidad. Dependiendo de la velocidad de conexión, puede hacer una autorización bancaria o una transacción local, dependiendo el caso, hasta en 2 segundos.

Se utiliza en comercios con gran afluencia de clientes o comercios que realicen muchas ventas con tarjetas bancarias o de vales y sus importes sean pequeños o cuando el comercio haga muchas transacciones de aplicaciones de mercados verticales, como tarjetas de lealtad, de regalo, etc.

Para poder hacer uso de una Terminal de este tipo, es necesario contar con conexión a Internet permanente, esta conexión puede ser de banda ancha proveída por Telmex o Internet por cable.

Los equipos 3600 Wi-Fi es un equipo móvil con la más nueva tecnología inalámbrica, que trasmite a través de la red inalámbrica Wi-Fi en la frecuencia libre de 2.4 Ghz, es decir que no se tiene que pagar a ningún proveedor por el uso del aire para que la Terminal transmita desde ella hacia la base receptora conocida como puerto de acceso o Access Point. A partir de este punto de acceso la transmisión hacia Internet se realiza a través de una conexión de banda ancha.

Se utiliza en locales abiertos o cerrados como ferias, recintos de exposiciones o inclusive donde exista una alta concentración de comercios en una misma área como mercados populares o gasolineras, pero que requieran de movilidad, es decir que no quieran depender de que el equipo esté conectado a la corriente eléctrica ya que esta Terminal cuenta con su propia batería de respaldo, con lo cual NO es

necesario que se encuentre conectada a la corriente eléctrica permanentemente para poder estar funcionando como le sucede a los equipos fijos. La duración de la batería es de aproximadamente de 8 horas en stanby (encendida sin uso) o de 2 a 3 horas de uso continuo. Se pueden adquirir baterías adicionales para contar con más horas de uso. Un buen ejemplo de comercio para este tipo de terminales, es un restaurante que necesite mas de 3 o 4 terminales para cobrar en la mesa de sus clientes ya que esto le da mayor confianza al consumidor además de que el restaurante podrá desalojar la mesa en promedio en 1 minuto y medio, ya que con esta Terminal no es necesario que el mesero deslice la tarjeta bancaria y se lleve la Terminal a una base para cobrar a través de la línea telefónica, porque a través de la red celular de autorización se imprimirá frente al cliente en su propia mesa en aproximadamente 5 segundos, mas el tiempo que tarde el cliente en firmar el voucher y el mesero en imprimir las copias.

Esta es la única solución disponible hoy en día para ambientes específicos como ferias donde hoy en día ningún banco ha podido ofrecer una solución de cobro para todos los locatarios o inclusive para ferias móviles.

Cabe mencionar que existen otros modelos de terminales verifone, que igual que la anterior, cuentan con la mayor tecnología de punta disponible hoy en día, los equipos GPRS o CDMA, que al igual que el modelo 3600 son equipos móviles, que transmiten inalámbricamente a través de la red celular de datos de las compañías Iusacelll, Telcel o Movil, no es necesario contar con línea telefónica fija.

Su principal uso es cuando el comercio necesita realizar ventas fuera de su local o en lugares donde es muy costoso o imposible contar con una línea telefónica para realizar las transacciones de una Terminal tradicional o inclusive cuando el comercio desea reducir sus altos costos por el uso de la línea telefónica tradicional para realizar cobros con tarjeta bancaria, pero con la posibilidad de llevar la Terminal consigo hasta donde se encuentra el consumidor o tarjetahabiente, ya que también esta Terminal cuenta con batería propia.

Para operar esta Terminal es necesario contar con conexión a Internet de datos celular, el costo de la transacción depende del plan que se contrate con la compañía telefónica, es importante señalar que esta Terminal podrá transmitir siempre y cuando la compañía celular cuente con cobertura de datos en la población donde se va a instalar la Terminal, lo cual debe verificarse con dichas compañías.

Todos estos modelos de terminales se programan con lenguaje WML, las características de dicho lenguaje serán explicadas posteriormente.

5.2 Lenguaje de programación WML, implementación de aplicación en servidor. Diagramas, pantallas y programación.

Este capítulo presenta el lenguaje WML (Wireless Markup Language) por sus siglas en ingles, sus orígenes y similitudes con otros lenguajes de programación meramente como referencia para tener una idea generalizada de las herramientas utilizadas en el desarrollo del sistema DLT Card.

WML es un lenguaje de programación basado en etiquetas que se usa para describir la estructura de los documentos que se distribuirán a través de dispositivos sin cable lo mismo que HTML es a los ordenadores de trabajo. WML se creó para que se hiciese cargo de las limitaciones de pantalla, ancho de banda y memoria propias de los teléfonos móviles y de los dispositivos sin cable, como son los mini ordenadores handheld o como en este caso, en terminales punto de venta. Como se diseñó para que se pudiese ejecutar en varios dispositivos, WML asume muy pocas cosas sobre el dispositivo en el que se ejecuta la aplicación y proporciona mucho control sobre los formatos de salida que se han de utilizar con HTML.

A principio de la década de los 90, Unwired Planet creó HDML (Handheld Device Language) como estándar para el desarrollo de las aplicaciones sin cable. Y en junio de 1997, Unwired Planet cambio su nombre por el de phone.com y, junto con nokia, motorola y Ericsson, creó WAP Forum, una organización sin ánimo de lucro dedicada al desarrollo y proliferación de un protocolo estándar para el desarrollo de las aplicaciones sin cable. Usando el lenguaje HDML de phone.com

como base para su lenguaje de etiquetas, el grupo creó y distribuyó WML, un lenguaje independiente de HDML pero con ciertos aspectos similares.

Aunque los desarrolladores de WML pueden encontrar ciertas similitudes con HTML, la verdad es que los dos lenguajes son más primos que hermanos. Se puede considerar que el padre de ambos HTML y XML (Extensible Markup Language) es SGML (Standardized Generalized Markup Language). HTML se diseñó para trabajar con una gran cantidad de objetos, imágenes y multimedia. La limitación del ancho de banda de los dispositivos móviles hace que no se pueda utilizar para desarrollar sus contenidos. Basándose en HTML se desarrolló el WML, que necesita una sencilla arquitectura de datos para ayudarle a analizar el documento. La necesidad y el deseo de trabajar con un lenguaje que podría sobrevivir a las demandas y fluctuaciones de un proceso de estandarización, fue la razón por la que se basó WML en XML. Al basarse en XML, el diseño de XML permitió crear un protocolo ligero que respetase las limitaciones de ancho de banda propia de los dispositivos móviles existentes.

Si se está familiarizado con HTML, SGML o XML, seguro se reconocerá la sintaxis de WML que se basa en XML, pero tiene una definición más genérica que las aplicaciones escritas en este lenguaje. La diferencia entre trabajar en HTML y WML, quizá la más importante, es que la pantalla del dispositivo es muy pequeña y varía de unos modelos a otros. Los dispositivos que se encuentran en la actualidad en el mercado tienen pantallas que van desde las que pueden trabajar con gráficos super VGA a las que únicamente admiten texto, con cuatro líneas de 32 caracteres cada una. En segundo lugar, la gran

cantidad de dispositivos que pueden trabajar con estas aplicaciones hace que los desarrolladores tengan cada vez más complicado hacerse una idea del aspecto que tendrá su aplicación en la pantalla del usuario. El desarrollador de WML no se limita a trabajar con un par de exploradores. Se ha de tener en cuenta cientos de exploradores, cada uno con una pantalla y una interfaz única.

La forma mas sencilla de poner los contenidos WML disponibles es utilizar un servidor http. A menos que el sitio WAP se encuentre dentro de un portal que pertenezca a un distribuidor de servicios para móviles, los usuarios tendrán que escribir la URL a mano una tarea complicada cuando únicamente se cuenta con las teclas de un teléfono o una Terminal de punto de venta, en cualquier caso no basta con publicar nuestras aplicaciones en un servidor http estándar porque los usuarios no podrán ver nada a través de la pantalla de sus móviles. El servidor le ha de indicar al explorador WML que va a recibir una página WML en vez de una HTML o cualquier otro tipo de contenidos. Esta información se desarrolla utilizando las extensiones MIME que se han de agregar al servidor para que los exploradores WML puedan ver correctamente el contenido de las cartas en las pantallas de los dispositivos.

MIME es el acrónimo de Multipurpose Internet Mail Extensión, que es una parte de la información de la cabecera que se utilizaba originalmente en los mensajes de correo electrónico para permitir dar un formato adecuado a los mensajes no ASCII que se enviaban por Internet. Hay muchos tipos MIME predefinidos de uso común, como archivos gráficos JPG y HTML. Además de los programas de correo electrónico, los exploradores Web también pueden trabajar con una

gran variedad de tipos MIME. Esto permite que el explorador pueda mostrar el contenido de los archivos utilizando otros formatos distintos de HTML. Los tipos MIME que suelen incluir los servidores de Internet son: text/html para archivos HTML, image/jpg para archivos JPG, image/gif para archivos GIF.

Los tipos de archivos que fueron necesarios implantar en el servidor para el reconocimiento de la aplicación de la P.O.S. se describen a continuación.

Como estos tipos de archivos son muy comunes, la mayoría de los servidores Web están configurados para enviar los tipos correctos. Pero WAP requiere sus propios tipos MIME para reconocer el contenido de los distintos archivos.

La versión 1.1 de WAP necesita cinco tipos MIME que se muestran en la siguiente tabla, para servir contenidos WML, WMLScript y wireles Bitmap.

Wml	Codigo fuente WML	Text/vnd.wap.wml
Wmls	Codigo fuente WMLScript	Text/vnd.wap.wmlscript
Wbmp	Mapas de bits sin cable	Image/vnd.wap.wbmp
Wmlc	Wml compilado	Application/vnd.wap.wmlc
Wmlsc	Wmlscript compilado	Application/vnd.wap.wmlscriptc

Todos estos tipos de archivos, fueron agregados en el servidor IIS de Windows 2000 Server, plataforma sobre la cual fueron implantados los ambientes de desarrollo y de producción de la aplicación P.O.S.

El tamaño reducido de las pantallas puede solucionar varios problemas, pero presenta uno nuevo: el límite de las dimensiones de la pantalla de cara a la aplicación. Las aplicaciones WML importantes solucionan este problema suministrando los datos más importantes en la parte superior de la pantalla. Se ha de tener en cuenta que las aplicaciones que ofrecen una gran cantidad de contenidos pueden llegar a cansar a los usuarios que han de desplazarse constantemente por menús y pantallas hasta que encuentran la información que les interesa. Este tipo de aplicaciones pueden llegar a ser poco atractivas y prácticas.

A la hora de desarrollar una aplicación para un dispositivo, hay que formularse cuatro preguntas:

¿Necesita el usuario acceder en tiempo real a los datos importantes?

¿Se pueden presentar los datos de una forma intuitiva en un dispositivo que tiene limitaciones de pantalla?

¿Se pueden cubrir las necesidades de los usuarios transmitiendo pequeñas cantidades de datos?

¿Contará el usuario con una buena cobertura?

Cada una de las preguntas anteriores ayudará a asignar la aplicación a dispositivo móvil.

Basándose en las características de las aplicaciones para dispositivos móviles, la navegación fue desarrollada para la mínima intervención del usuario, guiándolo a solo presionar opciones predefinidas en la pantalla, reduciendo los márgenes de error para el despachador, como se ilustra en el diagrama 5.2.

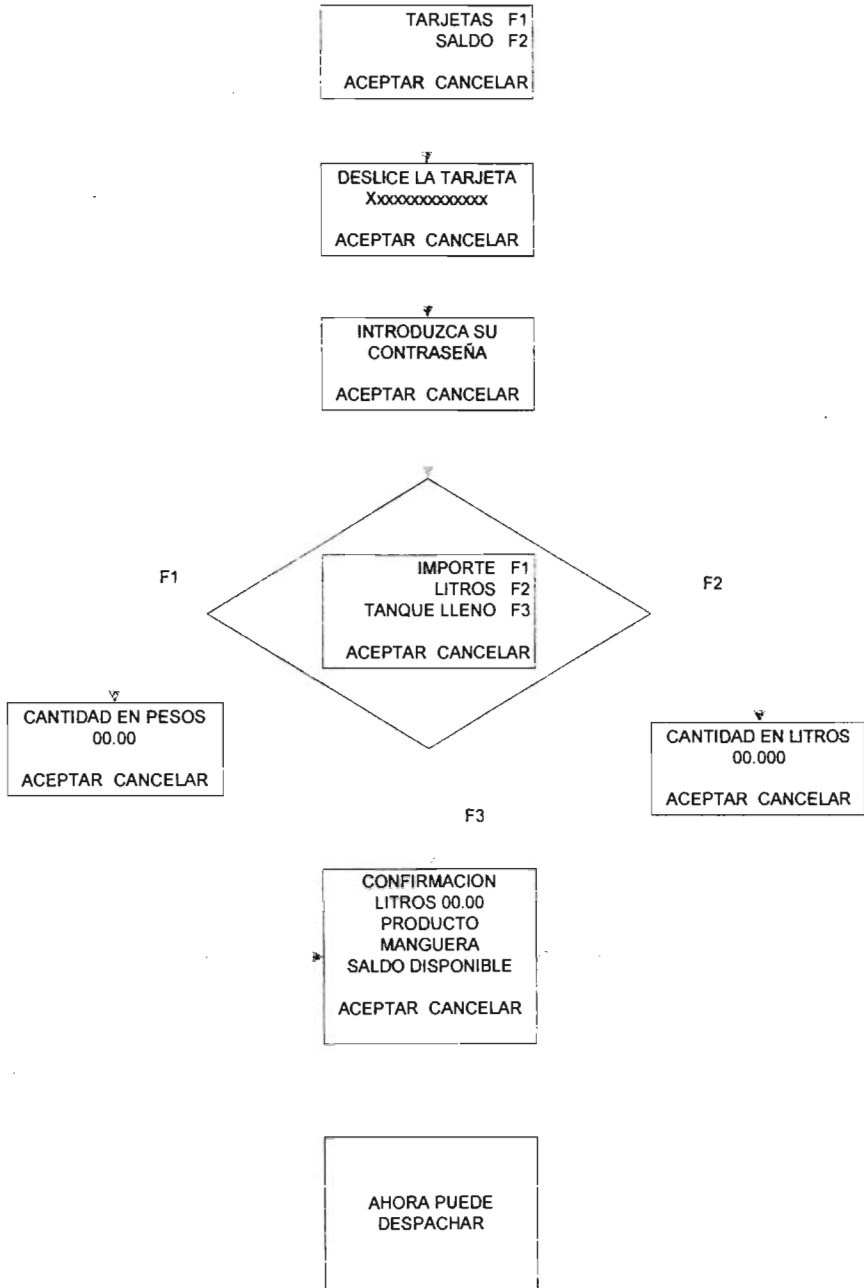


Fig. 5.2 Flujo de pantallas en Terminal P.O.S.

Dicho diagrama muestra las pantallas que aparecen en la Terminal instalada en la estación de servicio, como se observa, la intervención del despachador es mínima, para el caso de uso de despacho el flujo comienza eligiendo con la función F1 la opción de tarjetas, se pide al despachador que deslice la tarjeta, se pide indique el importe, litros o tanque lleno a despachar, con las funciones F1, F2 o F3 respectivamente, posteriormente se pide introduzca el número de manguera en la cual se va a despachar, se muestra una confirmación de las opciones digitadas anteriormente, y por ultimo se muestra una pantalla indicando que es momento de realizar el despacho.

En el diagrama 5.3 se describen los casos de uso válidos para el módulo del sistema P.O.S.

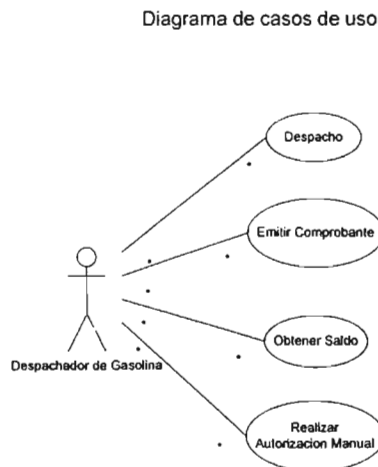


Fig. 5.3 Diagramas de casos de uso para el módulo de P.O.S.

El actor descrito como despachador de gasolina debe realizar un despacho al cliente, emitir un comprobante de un despacho realizado, obtener un saldo de algún chofer y en caso de falla técnica del sistema, realizar una autorización manual, falla técnica refiriéndose a la ausencia de red o baja del servicio de Internet en el servidor de producción de la aplicación, el cual es necesario para que la aplicación opere.

Una vez instalado el servicio de web en el servidor producción, agregadas las extensiones necesarias para el reconocimiento del lenguaje WML de la aplicación en el mismo, los archivos del sistema como despacho.asp, cliente.asp, chofer.asp entre otros, se colocaron en el directorio raíz de dicho sitio web.

Estos archivos representan en programación las clases diseñadas para el módulo P.O.S. para cada uno de sus casos de uso, así como demás archivos necesarios para la configuración de la aplicación, dichas clases se muestran en la figura 5.4.

Diagrama de estructura estatica caso de uso Despacho

La clase error tiene que ver con la solución del problema mas no así con el problema en si, por lo que no se relaciona con las demás clases en este momento

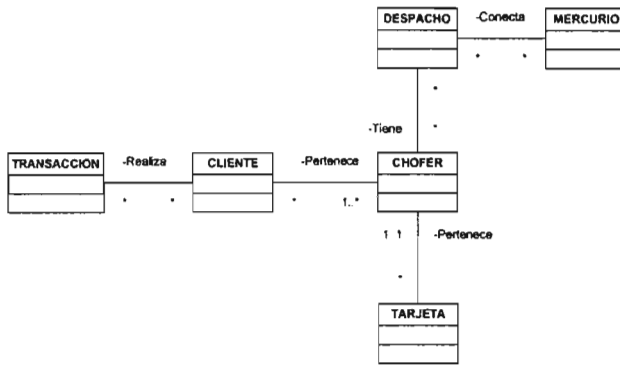


Fig. 5.4 Diagrama de clases del caso de uso despacho.

Dicho diagrama contiene un ejemplo de un caso de uso, despacho, el cual contienen las clases y como interactúan entre ellas, describiendo las tareas que hace cada clase y que espera de la clase con la que esta relacionada.

El chofer pertenece a un cliente, el cliente realiza una transacción, al chofer pertenece una tarjeta, el chofer realiza un despacho y el despacho hace una conexión a mercurio. Esta última clase es la encargada de realizar la conexión con el programa que controla los dispensarios de gasolina, y posterior a la conexión, autoriza el despacho o no, en caso de haber un error, la clase error envía el mensaje correspondiente, pero esta no esta relacionada entre las demás clases, pero ayuda al desarrollo de la aplicación.

Como estas clases, fueron diseñadas y programadas todas las demás involucradas en el sistema, dichos diagramas de estructura estática y los diagramas de secuencia se encuentran en la sección de anexos.

5.3 Sistema administrativo DLT Card.

Dicho software permite al personal administrativo de la estación de servicio supervisar la operación diaria, dar de alta cliente, asignar tarjetas a choferes así como asignar estos a clientes dados de alta.

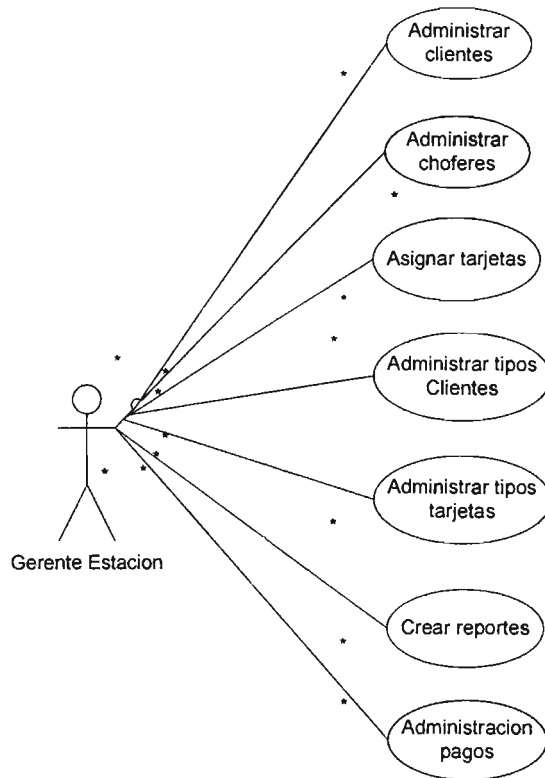


Este sistema se encuentra instalado en el equipo de cómputo de cada gerente de estación de cada una de las estaciones de servicio, además de estar instalado en el equipo de la asistente del gerente, ya que es este personal quien ayuda a la administración de la estación de servicio.

El sistema administrativo DLT Card, fue desarrollado con lenguaje Visual Basic 6.0, si no el mas actual, si el mas utilizado por desarrolladores para realizar sistemas bajo plataformas Microsoft.

DLT Card esta compuesto por un módulo principal en el cual se cambia de usuario, configuraciones de la barra de herramientas así como de las impresoras. Un módulo de tarjetas, en el cual se lleva acabo la administración de los clientes, choferes y la asignación de tarjetas a estos, la sección de catálogos, de tipos de cliente, tarjeta, bancos o documentos, la sección de reportes de clientes, estados de cuenta y reportes de la estación de servicio, sección de pagos en donde se puede agregar, cancelar y consultar los pagos. Un módulo de usuarios en donde se administraban los usuarios, agregar, consultar, modificar o eliminar un usuario y el módulo de ayuda, en donde se hacia referencia del sistema.

Cada módulo de los mencionados anteriormente, fue diseñado de la misma manera que fue diseñado el módulo de P.O.S., haciendo uso de UML.



La figura 5.5 muestra los casos de uso para el módulo DLT Card.

El actor descrito como gerente de estación de servicio o la asistente, deben dar de alta o baja a un cliente y a los choferes que les correspondan a cada cliente, asignar tarjetas a los chorefes

autorizados por el cliente, administrar tipos de tarjetas, si la tarjeta debe ser de débito o crédito, asignar restricciones a estas como días de carga, tipo de combustible asignado, límites de crédito, etc., generar reportes de venta de tarjetas para el área contable así como administrar pagos de los clientes, liberar crédito a estos mismos o cancelar servicio por falta de estos.

Para el caso de uso de administrar clientes, la estructura estática se presenta a continuación:

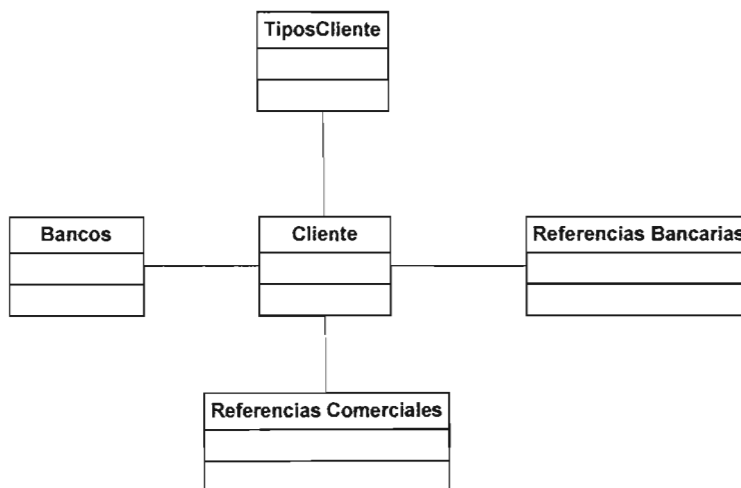


Fig. 5.6 Diagrama de clases del caso de uso Administración de clientes.

Como se muestra en la figura 5.6 un cliente cuenta con una o varias referencias bancarias, así como cuentas bancarias en distintos bancos, cuenta con una o varias referencias comerciales, así como un cliente puede ser de uno o varios tipos de clientes dentro del sistema,

pudiendo ser estos de crédito, que son aquellos clientes que realizan consumos en la estación de servicio y estos son pagados en el tiempo estipulado por ambos, de débito, quienes depositan la cantidad a consumir en un periodo determinado, o los clientes de facturación, quienes pueden realizar sus pagos de cualquier forma y solo cuentan con una tarjeta de facturación, para expedir dicho documento en el momento del pago.

Como la figura 5.6, existen otros diagramas que explican la lógica de los demás casos de uso del sistema, de los cuales, algunos se encuentran en la parte de anexos, siendo así que, un chofer solo puede estar asignado a un cliente y no a varios, así como puede pertenecer a uno o varios centros de costos, léase centros de costos como unidades administrativas dentro de la organización del cliente, por ejemplo, para uno de los principales clientes de las estaciones de servicio en estudio, el H. Ayuntamiento de Atizapan de Zaragoza, sus centros de costos son denominados conforme a sus áreas: bomberos, protección civil, seguridad pública, etc. Unido a esto, una tarjeta debe pertenecer a solo un chofer, el chofer a un cliente, y pueden existir varios tipos de tarjetas, que como se mencionó con anterioridad, pueden ser de acuerdo al tipo de cliente dado de alta en el sistema, facturación, débito o crédito.

Para poder llevar acabo la administración de los clientes, al momento de dar de alta un cliente deben de existir tipos de clientes registrados en el sistema, así como introducir las referencias tanto bancarias como comerciales del cliente. Para el caso de un chofer debe existir un cliente registrado, y tener registrados centros de costos para ese cliente, caso similar se presenta al asignar una tarjeta a un chofer,

la tarjeta pertenece a un chofer y el chofer a su vez a un cliente, pero para poder asignar una tarjeta debe de haber en el sistema tipos de tarjeta dados de alta. Estos procesos se muestran en los diagramas de secuencia de cada caso de uso, para ejemplificar un poco mas este proceso, en la figura 5.7 se muestra el diagrama de secuencia del caso de uso administrar clientes, para todos los demás casos de uso, el diseño fue realizado de la misma manera.

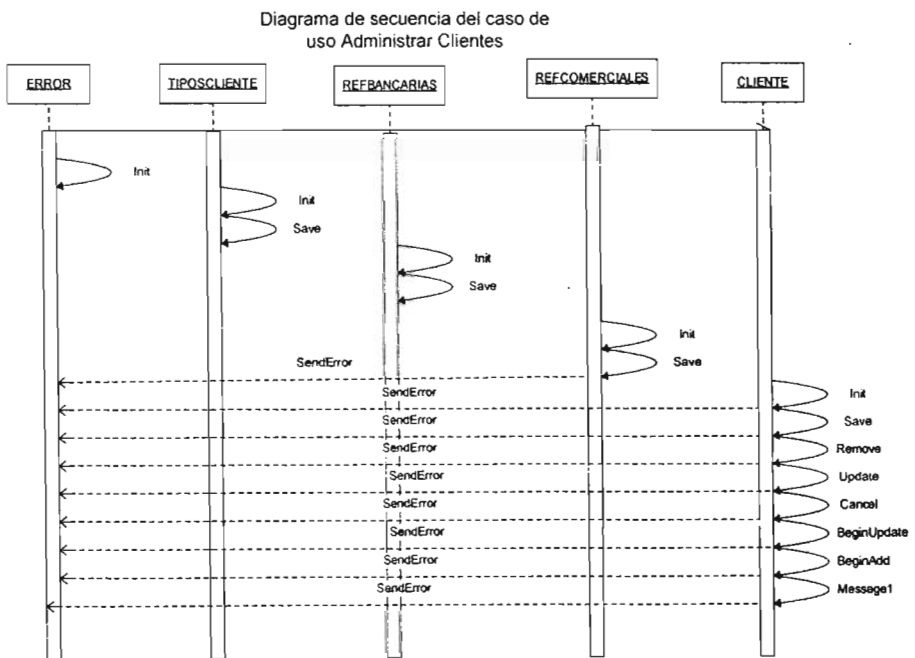


Fig. 5.7 Diagrama de secuencia caso de uso Administrar Clientes.



CONCLUSIONES.



CONCLUSIONES.

La instalación y el uso del sistema DLT Card en la estación de servicio 5734 Atizapan, ubicada en Blvd. Adolfo López Mateos 101 colonia el mosco, en el municipio de Atizapan de Zaragoza, Estado de México aporta beneficios tanto a la administración de la propia estación, como a los clientes que hacen uso del sistema por medio de sus tarjetas personalizadas.

Para la estación de servicio, la administración de los clientes fue automatizada mediante el módulo DLT Card, los clientes de crédito fueron ingresados al sistema, desde ahí eran cobrados los créditos, capturaros sus pagos, etc. Se eliminó el uso de vales propios de la estación (hoja de papel scribe sellado por la estación) y fueron reemplazados por tarjetas personalizadas para cada chofer de los clientes. Los reportes entregados al gerente de la estación, son entregados en cada corte que se realiza, recopilando información de todos los despachos de crédito realizados en un turno, dichos reportes son generados al momento, sin hacer tareas adicionales, no complicando con esto la operación de la propia estación.

Con la colocación de los puntos de venta en piso, se deja abierta la posibilidad de ofrecer mas servicios verticales dentro de la misma Terminal, debido esto a que solo direccionando la Terminal a otro sitio web, se pueden comenzar a hacer pagos de servicios, remesas, venta de boletos de autobús, entre otros, facilitando esto, ya que la infraestructura se encuentra instalada.

La instalación de este sistema, fue un punto a favor de la estación de servicio para la licitación de la prestación del servicio al H. Ayuntamiento Municipal de Atizapan, debido a que dicha institución buscaba todas las características ofrecidas por el sistema DLT Card, llevando a la firma de un contrato por 3 años entre ambas partes para el despacho de combustible, remunerando para la estación de servicio un cliente que consume mensualmente un millón de pesos cautivos, que realiza 9000 despachos mensuales en promedio y que deja a la gasolinera la imagen de la capacidad de despacho que cuenta, sin mencionar que el sistema se implementará en las tres gasolineras restantes, causa por la cual, el mismo servicio se está ofreciendo a administraciones similares como el H. Ayuntamiento Municipal de Tlalnepantla, abriendo de esta forma las expectativas de negocio de la gasolinera.

Para el cliente, específicamente el mas importante que es el H. Ayuntamiento de Atizapan de Zaragoza, representó el uso de este sistema, el ahorro de cerca de doscientos mil pesos que se fugaban por el mal manejo del control o mejor dicho de la ausencia de control en los despachos, dinero que o bien no se cargaba en las unidades especificadas, o se sacaba de las unidades una vez ya llenas de combustible. Aunado a esto, se reemplazó personal encargado del propio Ayuntamiento, que vigilaba las cargas de los choferes, disponiéndolo para otras actividades.

Si bien el sistema porta los beneficios mencionados, también, como todo sistema, es susceptible de ser adecuado a nuevos requerimientos.

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

Entre las mejoras que se pueden incorporar al sistema está la migración del modulo DLT Card, que actualmente esta instalado en cada equipo del personal administrativo de la gasolinera, a una aplicación 100% vía web, para así tener concentrado en un solo punto la información de todos los clientes de las 4 estaciones de servicio, además, de cómo se mencionó con anterioridad, la aplicación de mas servicios verticales en la misma Terminal, pudiendo también migrar todas las terminales fijas o terminales móviles modelo 3600, dando mayor agilidad al despacho.

En general, se satisfacen los requerimientos planeados por la gerencia de las estaciones de servicio y se cumple con el objetivo establecido de desarrollar un sistema informático útil que facilite la administración y el control en tiempo real de las ventas de combustible utilizando las tecnologías de información e Internet.



ANEXOS.

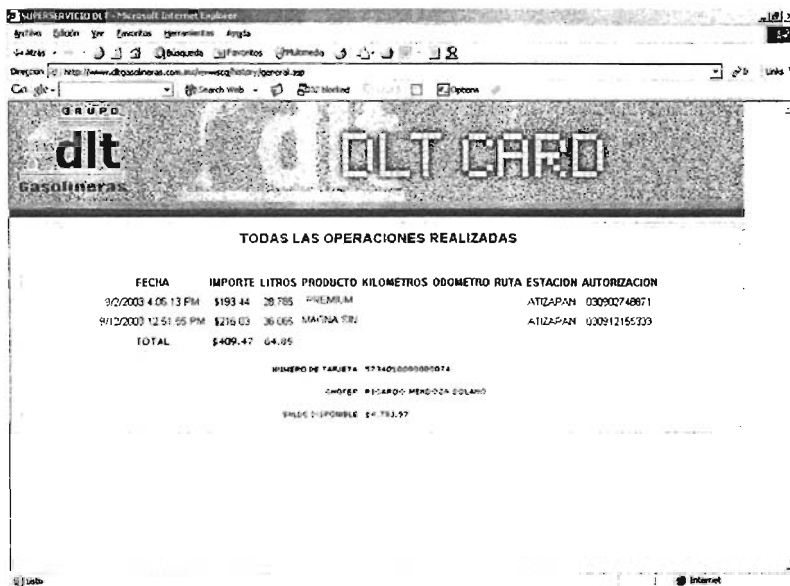


ANEXOS.

Página de internet.

www.dltgasolineras.com.mx

Permite la consulta de los despachos y saldos de cualquier cliente en tiempo real.



The screenshot shows a web browser window displaying the DLT Card website. The page title is "SUPER SERVICIO DLT" and the URL is "http://www.dltgasolineras.com.mx/consultahistoria/genera.asp". The main content area features the DLT logo and the text "DLT CARD". Below this, there is a section titled "TODAS LAS OPERACIONES REALIZADAS" which contains a table of transactions. The table has columns for FECHA, IMPORTE, LITROS, PRODUCTO, KILOMETROS, ODOMETRO, RUTA, ESTACION, and AUTORIZACION. Two transactions are listed, with a total amount of \$409.47 and 64.85 liters. Below the table, the card number is displayed as "NÚMERO DE TARJETA: 4734010000000074", the cardholder's name as "CHOFER: RICARDO MENDOZA COLARDO", and the balance as "SALDO DISPONIBLE: \$ 793.97".

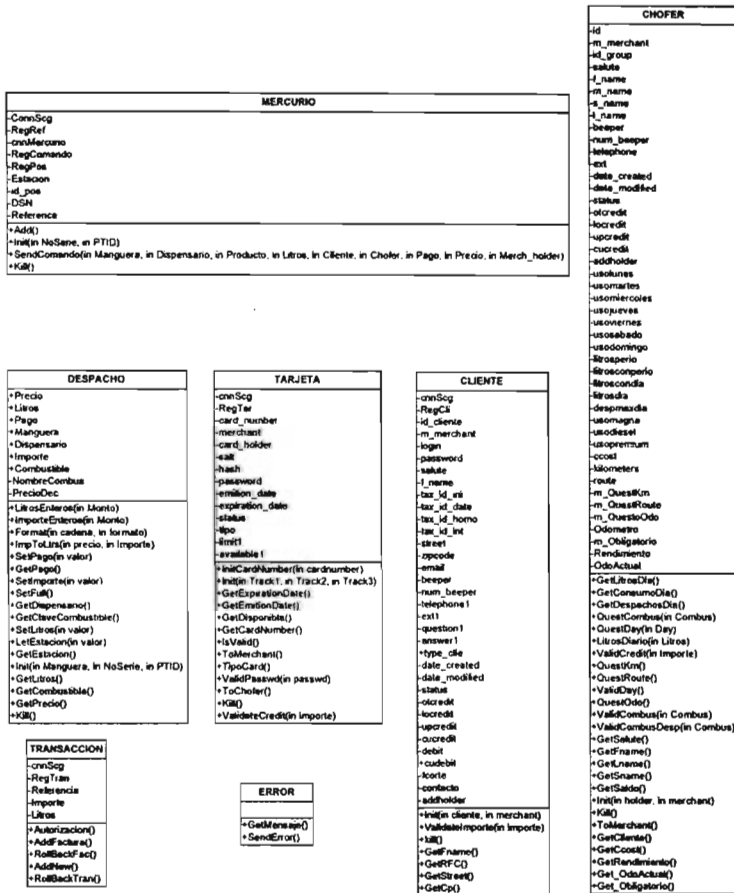
FECHA	IMPORTE	LITROS	PRODUCTO	KILOMETROS	ODOMETRO	RUTA	ESTACION	AUTORIZACION
9/2/2003 4:06:13 PM	\$193.44	28.785	PREMIUM				ATIZAPAN	030902748871
9/12/2003 12:51:05 PM	\$216.03	36.065	MAYONA SIN				ATIZAPAN	030912155339
TOTAL	\$409.47	64.85						

NÚMERO DE TARJETA: 4734010000000074
CHOFER: RICARDO MENDOZA COLARDO
SALDO DISPONIBLE: \$ 793.97

La página de consultas fue desarrollada mediante asp, básicamente presenta información por centros de costos, por chofer, por turno, los despachos que realizaron, su importe, litros consumidos, el producto, autorización, entre otra información requerida por el cliente, dicha información se puede exportar a un archivo de Excel para poder así entregar la información impresa al cliente.

Diagramas del sistema.

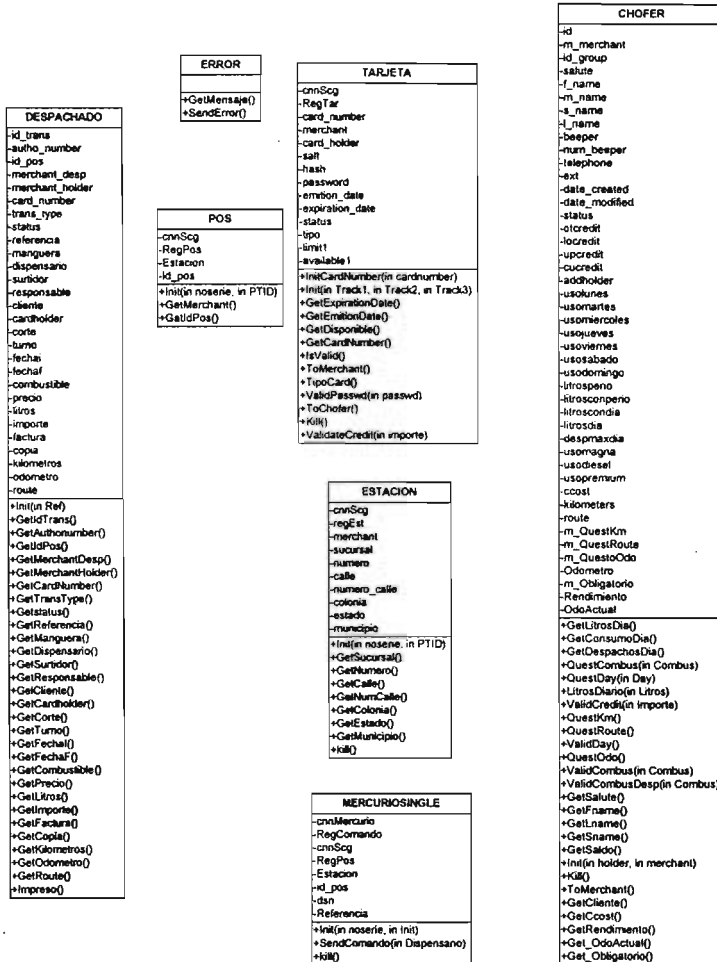
Clases de caso de uso Despacho



La clase mercurio es la encargada de conectarse al dispensario de gasolina y autorizar o no los despachos, las clases despacho, tarjeta, cliente y chofer, contienen las propiedades y métodos de esas clases, la clase transacción es la encargada de completar con una

referencia, importe, y otros datos la transacción una vez que esta haya terminado.

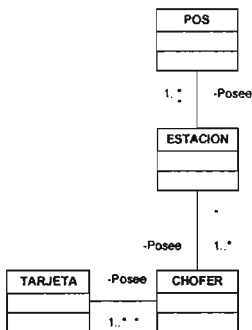
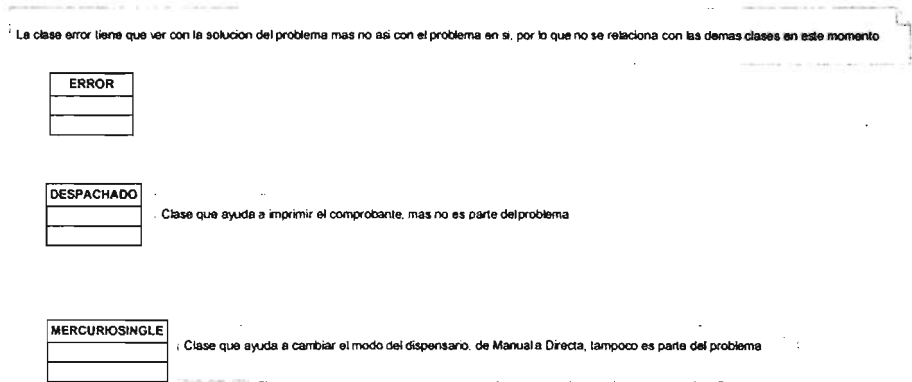
Clases de caso de uso Emitir Comprobante



La clase mercuriosingle pasa de modo manual a modo automático el dispensario de gasolina, una vez que finalizo la transacción y esta está marcada como realizada, despacho, tarjeta,

estación y chofer reúnen propiedades y métodos de las clases, la clase pos identifica, de que Terminal y estación salio el despacho.

Diagrama de estructura estatica de caso de uso
Emitir Comprobante



Las clases error, despachado y mercuriosingle intervienen en la solución del caso de uso, mas no en el flujo del propio caso.

Aquí una estación posee una o más terminales, a una estación pertenecen uno o mas choferes, y al chofer pertenece una sola tarjeta. Es así como se cumplen las restricciones del flujo del sistema.

Clases de caso de uso Verificar Saldo

ERROR
+GetMensaje()
+SendError()

TARJETA
-cnnScg
-RegTar
-card_number
-merchant
-card_holder
-salt
-hash
-password
-emtion_date
-expiration_date
-status
-tipo
-limit1
-available1
+Ini(CardNumber(in cardnumber)
+Ini(in Track1, in Track2, in Track3)
+GetExpirationDate()
+GetEmtionDate()
+GetDisponible()
+GetCardNumber()
+IsValid()
+ToMerchant()
+TipoCard()
+ValidPasswd(in passwd)
+ToChofer()
+ValidateCredit(in importe)

CLIENTE
-cnnScg
-RegCfi
-id_cliente
-m_merchant
-login
-password
-sakute
-f_name
-tax_id_ini
-tax_id_date
-tax_id_homo
-tax_id_int
-street
-zipcode
-email
-beeper
-num_beeper
-telephone1
-ext1
-question1
-answer1
+type_cle
-date_created
-date_modified
-status
-olcredit
-locredit
-upcredit
-cucredit
-debit
-cudebit
-fcorite
-contacto
-addholder
+Ini(in cliente, in merchant)
+ValidateImporte(in importe)
+kill()
+GetFname()
+GetRFC()
+GetStreet()
+GetCp()

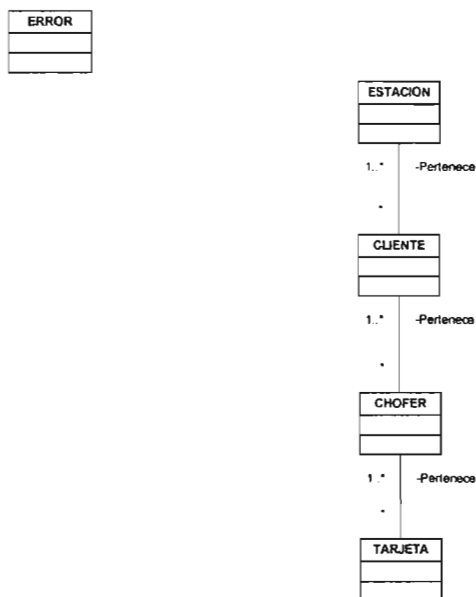
ESTACION
-cnnScg
-regEst
-merchant
-sucursal
-numero
-calle
-numero_calle
-colonia
-estado
-municipio
+Ini(in noserie, in PTID)
+GetSucursal()
+GetNumeros()
+GetCalle()
+GetNumCalle()
+GetColonial()
+GetEstado()
+GetMunicipio()
+kill()

CHOFER
-id
-m_merchant
-id_group
-sakute
-f_name
-m_name
-s_name
-l_name
-beeper
-num_beeper
-telephone
-ext
-date_created
-date_modified
-status
-olcredit
-locredit
-upcredit
-cucredit
-addholder
-usolunes
-usomarfes
-usomercules
-usosueves
-usoviermes
-usosabado
-usodomingo
-litrosperno
-litrosconpeno
-litroscondia
-litrosdia
-despmaxdia
-usomagna
-usodiesel
-usopremium
-ccost
-kilometers
-route
-m_QUESTkm
-m_QUESTroute
-m_QUESTodo
-Odometro
-m_Obligatorio
-Rendimiento
-OdoActual
+GetLitrosDia()
+GetConsumoDia()
+GetDespachosDia()
+QUESTCombust(in Combust)
+QUESTDay(in Day)
+LitrosDiano(in Litros)
+ValidCredit(in Importe)
+QUESTkm()
+QUESTroute()
+ValidDay()
+QUESTodo()
+ValidCombust(in Combust)
+ValidCombustDesp(in Combust)
+GetSalute()
+GetFname()
+GetLname()
+GetSname()
+GetSaldo()
+Ini(in holder, in merchant)
+Kill()
+ToMerchant()
+GetCliente()
+GetCcost()
+GetRendimiento()
+Get_OdoActual()
+Get_Obligatorio()

En este caso de uso, las clases que intervienen contienen solo las propiedades y métodos propios de las clases, debido a que se realiza una consulta a la base de datos del saldo que tiene ese chofer.

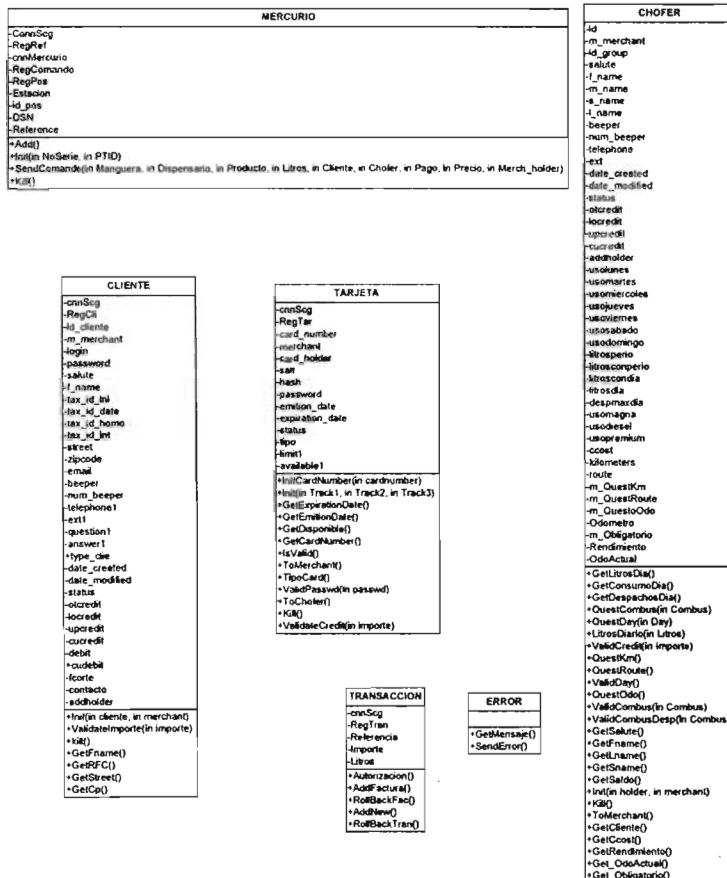
Diagrama de estructura estatica del caso de uso Consulta Saldo

La clase error tiene que ver con la solución del problema mas no así con el problema en sí, por lo que no se relaciona con las demas clases en este momento

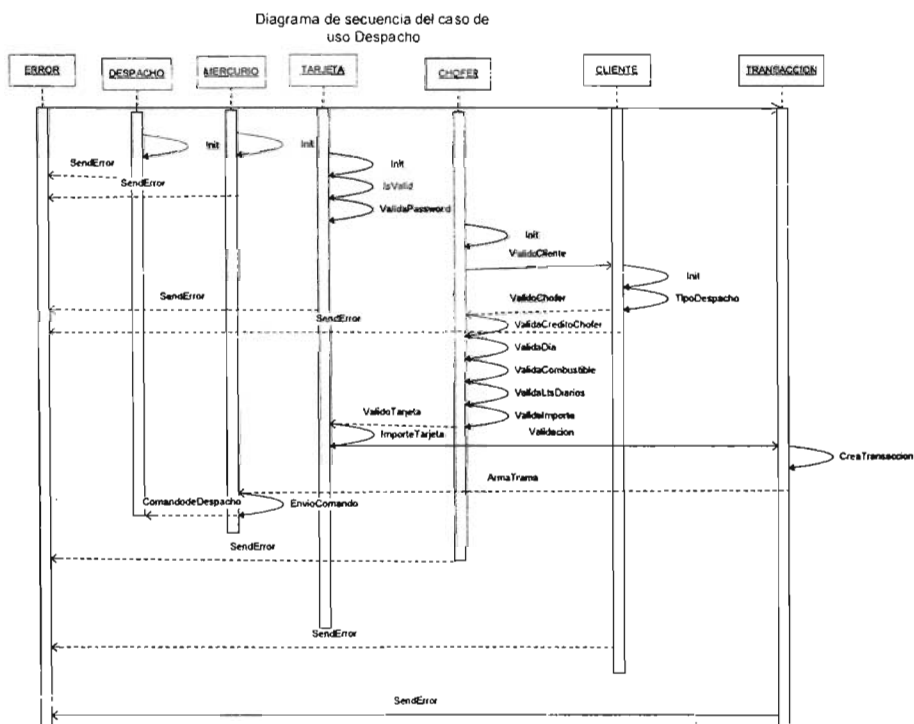


Es en este caso, en donde se genera el ticket del saldo del chofer, una estación posee a los clientes, los choferes pertenecen al cliente, y la tarjeta al chofer, es así como se construye la consulta de un saldo.

Clases de caso de uso Autorización
Manual

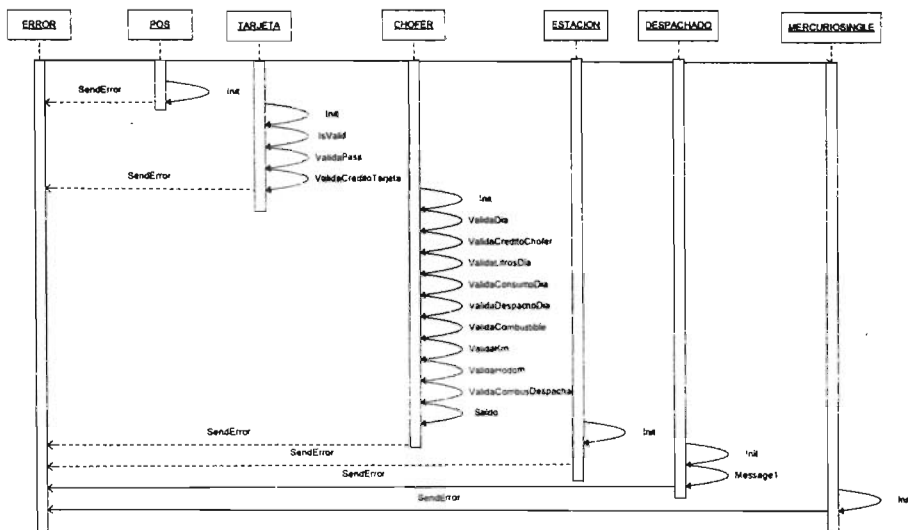


En caso de falla del sistema, sea a causas de red, energía eléctrica, o cuestiones ajenas al sistema, el supervisor por medio de una clave, puede asignar un despacho normal a una transacción del sistema, siendo descontada esta del saldo del chofer, por lo que se incluyen de la misma manera clases con las propiedades y métodos de cada una de ellas, siendo la clase transacción quien asigna el despacho a un chofer, de igual manera la clase mercurio es la encargada de realizar la conexión al dispensario y en su momento, despachar, si es que el despacho no se había realizado.



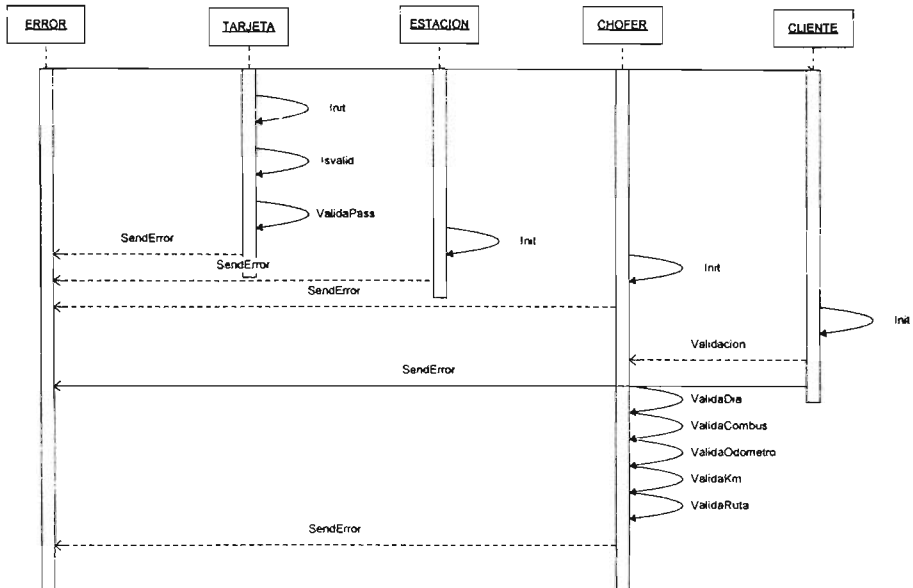
Se muestra la secuencia que debe de pasar por todas las clases en el caso de uso de despacho, en cada clase existe la posibilidad de enviar error, y volver a comenzar el proceso, en este caso en particular, hasta que no pasa todas las restricciones de chofer, tarjeta y cliente, es cuando realiza una transacción.

Diagrama de secuencia del caso de uso Emitir Comprobante

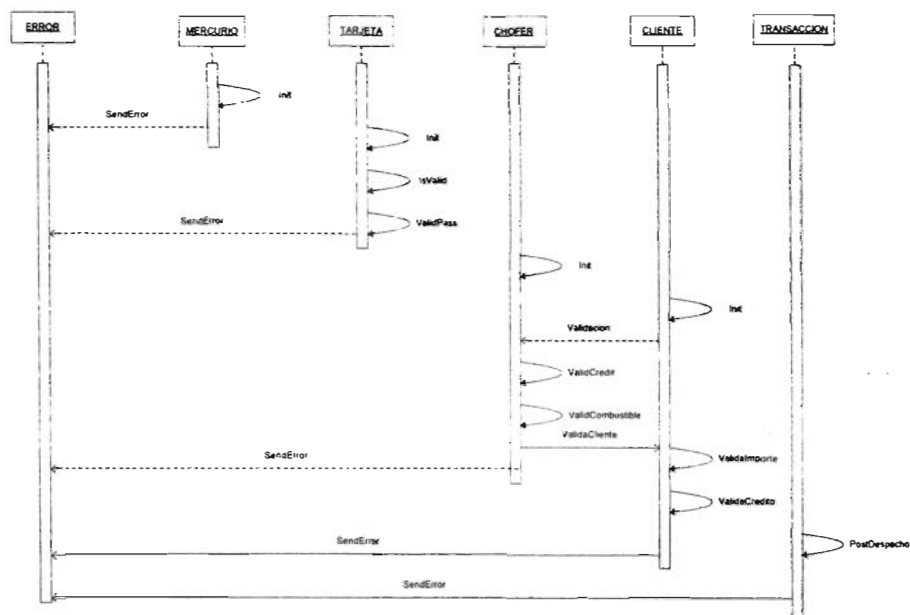


En el caso de uso de emitir un comprobante, las restricciones van desde verificar que la Terminal este dada de alta en el sistema, para evitar la emisión falsa de comprobantes, pasando por las validaciones de tarjeta, chofer y estación hasta llegar a la clase despachado y mercuriosingle que son quienes marcan la transacción como finalizada.

Diagrama de secuencia del caso de uso Consultar Saldo

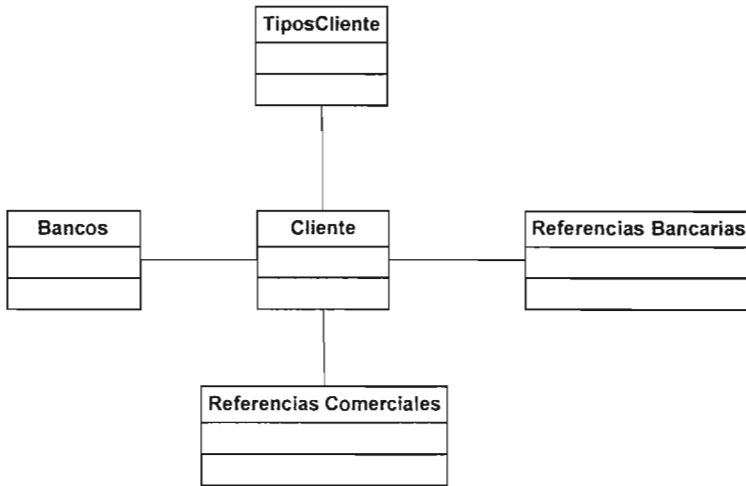


Para consultar un saldo, la secuencia a seguir es como se presenta en el diagrama anterior, se valida hasta la clase cliente, para verificar que exista el cliente y ahí imprimir un saldo.

Diagrama de secuencia del caso de uso
Asignación

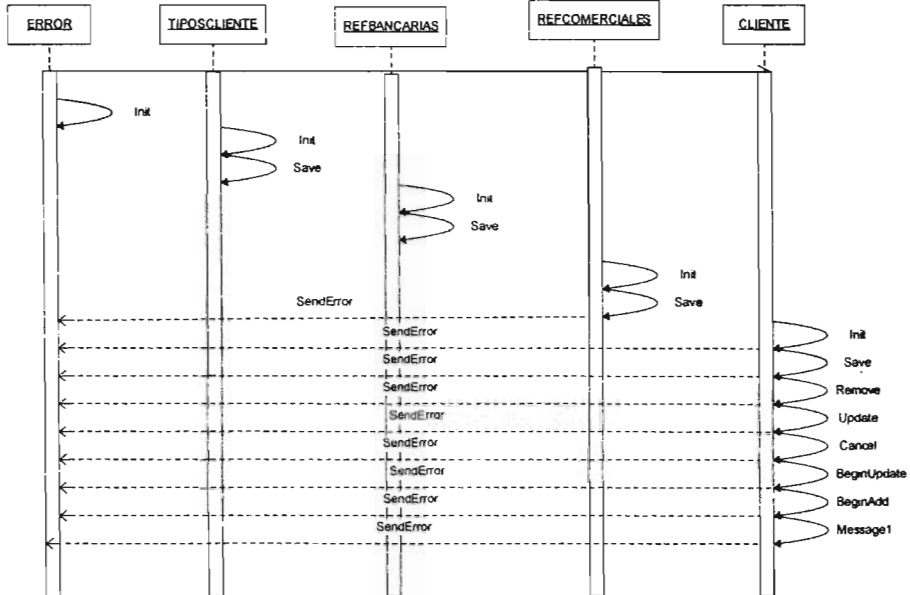
Para la asignación de despachos manuales, las restricciones son verificadas en las clases involucradas, mercurio, tarjeta, chofer y cliente, la clase transacción aquí es quien cuenta con un método PostDespacho, quien es el encargado de indicar que posteriormente a un despacho normal, habrá una asignación manual.

Diagrama de estructura estatica caso de uso administrar clientes



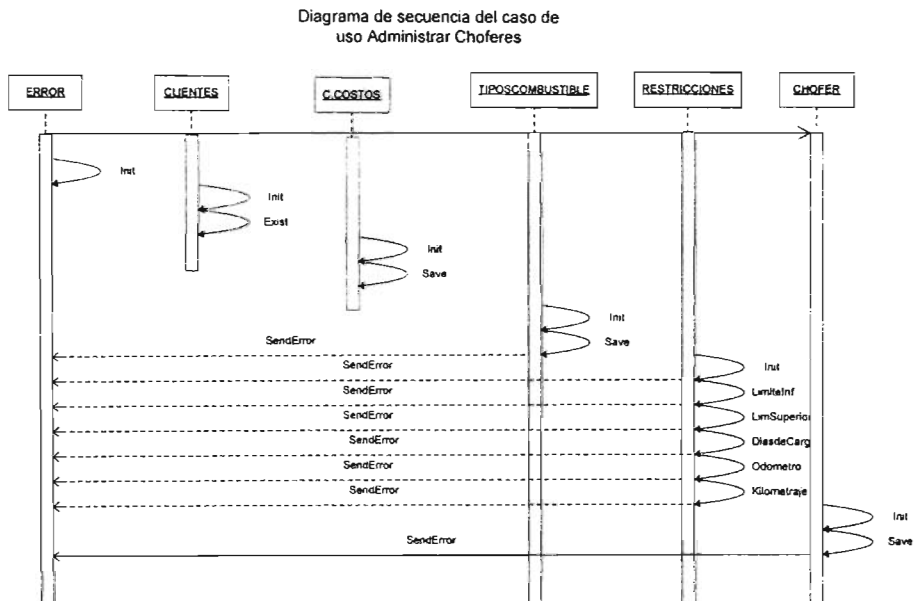
En lo que se refiere al módulo DLT Card, la estructura es similar para todos los casos de uso involucrados en este módulo, un chofer se asigna a un cliente de la misma manera, verificando que el cliente exista, que existan centros de costo asignados a los clientes, entre otras muchas restricciones, pero la lógica de programación es la misma, aquí el concepto de interfaz fue de gran utilidad, ya que de la misma manera que se da de alta un cliente, es exactamente igual a la manera de dar de alta un centro de costo o un tipo de tarjeta en el sistema. Las relaciones entre clases no cambian, solo cambian los nombres de estas y siempre son de una clase de nivel superior por llamarlo de una manera, hacia las dependientes de esta, como se podrá observar en el siguiente diagrama.

Diagrama de secuencia del caso de uso Administrar Clientes



Para agregar a un cliente, las clases dependientes de cliente deben inicializarse, como es en este caso, un tipo de cliente, una vez realizado esto, debe inicializar y existir una referencia bancaria y una referencia comercial, para así llegar a la inicialización de la clase cliente, en donde ella posee los métodos necesarios para administrar un cliente, un método save, remove, update, cancel, etc., cumpliendo con la lógica propia del sistema, de que no puede ser agregado, removido o modificado un cliente si no cuenta antes con un tipo de cliente y sus referencias respectivas.

En el siguiente diagrama se muestra como la administración de un chofer se realiza de la misma manera.



Para agregar un chofer, se realiza de la misma manera que un cliente, existiendo primero un cliente en el sistema, asignados a este cliente, centros de costo o unidades administrativas para el cliente, deben de existir de la misma manera, tipos de combustible válidos en el sistema, restricciones para el chofer, como lo son limites superior e inferior a asignar al chofer, días de la semana que esta autorizado el chofer a cargar combustible, si se va a verificar el odómetro o el kilometraje de la unidad del chofer entre otras restricciones, para así poder dar de alta un chofer en el sistema.

La interfaz utilizada para la administración de todos los casos de uso de este módulo fue la misma, en dicha interfaz se encapsulan las características que van a tener todos los catálogos a realizarse en el sistema, siendo así, un cliente se da de alta en el sistema de la misma manera que un chofer, una tarjeta, un centro de costo, etc. Y siendo así también, una baja o una modificación.



BIBLIOGRAFÍA.



BIBLIOGRAFÍA.

- Senn, Jame A.; "Análisis y Diseño de Sistemas de Información; 2ª edición; Mc Graw Hill; 2000.
- Stevens P., Pooley P.; "Utilización de UML en ingeniería de software con objetos y componentes"; 3ª edición; Pearson Education; 2002.
- Fowler, Martin; "UML Gota a Gota"; 1ª edición; Addison WesleyM 1999.
- Ross, Joel. Murdick, Robert; "Sistemas de información basados en computadoras"; 2ª edición; Editorial Diana; 1991.
- Mora, Jose Luis,; "Introducción a la informática"; 4ª edición; Trillas; 1985.
- Kendall, Kendall; "Análisis y Diseño de Sistemas"; 3ª edición; Prentice Hall; 1997.
- Desarrollo de WAP con WML y WML Script; Ben Forta; 1ª edición; Anaya multimedia.
- Revista ONEXPO, Octubre 2004.

REFERENCIAS ELECTRÓNICAS.

<http://www.uml.org/>

<http://www.lania.mx/biblioteca/manuales/umlrefman.pdf>