



# INSTITUTO POLITÉCNICO NACIONAL

---

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE  
INGENIERÍA Y CIENCIAS SOCIALES Y  
ADMINISTRATIVAS

“SISTEMA DE ADMINISTRACIÓN DE CLIENTES ON-DEMAND  
PARA EL MEJORAMIENTO DE LA ATENCIÓN AL CLIENTE”

T E S I N A

QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN INFORMÁTICA

P R E S E N T A :  
EDGAR GUSTAVO ÁLVAREZ VÉLEZ

QUE PARA OBTENER EL TÍTULO DE:  
LICENCIADO EN CIENCIAS DE LA INFORMÁTICA

P R E S E N T A N :  
CARLOS DARÍO CASTAÑEDA MENDOZA  
GILBERTO ROSAS ACEVES

# ÍNDICE

<b>Resumen</b> .....	<b>i</b>
<b>Introducción</b> .....	<b>ii</b>
<b>Capítulo I Marco metodológico</b> .....	<b>1</b>
1.1 Planteamiento del problema .....	1
1.2 Objetivos .....	2
1.2.1 Objetivo general .....	2
1.2.2 Objetivos específicos .....	2
1.3 Justificación .....	2
1.4 Tipos de investigación .....	3
1.5 Técnicas de investigación a emplear.....	3
<b>Capítulo II Marco Teórico</b> .....	<b>4</b>
2.1 El valor de las buenas relaciones con los clientes.....	4
2.2 CRM.....	5
2.3 Uso e Implementación de Tecnología en el CRM.....	6
2.4 Tipos de CRM.....	9
2.5 Clasificación de CRM.....	11
<b>Capítulo III Metodología y Tecnología</b> .....	<b>16</b>
3.1 Introducción al RUP.....	16
3.2 Características del RUP.....	16
3.3 Fases RUP.....	17
3.4 Disciplinas RUP.....	17
3.5 Introducción al UML.....	19
3.6 Descripción del Lenguaje UML.....	20
3.7 Descripción de los Diagramas UML.....	21
3.8 Introducción al J2SE.....	22
3.9 Tecnología J2SE.....	22
3.10 Ventajas J2SE.....	24
3.11 Desventajas J2SE.....	25
3.12 Integración con otros sistemas.....	25
3.13 Arquitectura de múltiples capas.....	26
<b>Capítulo IV Análisis y Diseño del Sistema</b> .....	<b>27</b>
4.1 Identificación de requerimientos.....	27
4.2 Análisis de requerimientos.....	27
4.3 Stakeholder y descripciones de usuarios.....	30
4.4 Ambiente a utilizar.....	31
4.5 Casos de uso.....	31

4.6 Diseño de la interfaz.....	47
4.7 Diagramas de secuencia.....	48
<b>Capítulo V Desarrollo y Pruebas del Sistema.....</b>	<b>57</b>
5.1 Diseño de la realización del caso de uso.....	57
5.2 Diagrama de clases.....	58
5.3 Diccionario de clases.....	61
5.4 Diagrama Entidad Relación.....	63
5.5 Diccionario de Datos.....	65
5.6 Pruebas.....	70
5.7 Ejercicio Práctico.....	71
5.8 Vista Despliegue.....	77
5.9 Matriz de Pruebas.....	77
<b>Conclusiones.....</b>	<b>80</b>
<b>Bibliografía.....</b>	<b>81</b>

## RESUMEN

En el presente trabajo se llevó a cabo el análisis, diseño y desarrollo de un sistema; el cual nos permitirá realizar las actividades correspondientes a la metodología de trabajo CRM.

En el capítulo 2 empezaremos por revisar los orígenes del CRM y el por qué es conveniente mantener una buena relación con los clientes. Trataremos de definir en esencia lo que es CRM, partiendo de un enfoque de proceso y como la tecnología lo complementa como sistema, se clasificarán los tipos CRM que existen actualmente basándonos en el funcionamiento que llegan a desempeñar dentro de las empresas.

Daremos una breve explicación en el capítulo 3 de la metodología RUP la cual es la más empleada para el desarrollo de software describiendo las fases y disciplinas que la componen así mismo describiremos las herramientas tecnológicas a emplear para la realización del sistema. Un complemento importante es la diagramación empleada para el desarrollo del sistema, daremos una breve introducción de la metodología UML, así como de cada uno de los diagramas que lo conforman.

Mostraremos algunas de las características del lenguaje J2SE empleado para la codificación del proyecto así como del diseño multicapas.

Durante los capítulos 4 y 5 se mostrará la aplicación de las metodologías y uso de las herramientas anteriormente descritas.

# INTRODUCCIÓN

Este trabajo tiene como objetivo principal el dar una solución tecnológica al problema que presentan principalmente las empresas de mantener una buena relación con sus clientes; ofreciendo una solución innovadora, económica de fácil implementación adecuándose 100% a sus necesidades. Para lograr el objetivo planeado se llevó a cabo un análisis de los CRM's mas comerciales y de las necesidades de las empresas en cuanto a su control de ventas, gestión de usuarios, valor de la información, marketing y seguimiento de ventas; generando requerimientos generales.

Con la implementación de una estrategia CRM, se pretende mejorar la relación con los clientes, desarrollando una lealtad y fidelidad hacia la misma. Esta herramienta nos proporcionara el conocimiento de los clientes, haciendo más estrecha la relación entre todos; además de aumentar la retención de éstos, mejorando así la operación de la empresa.

El CRM no es solo la aplicación de un software sofisticado con tecnología de punta, si no es el complemento de una actitud de servicio total, ya que facilita el manejo de la base de datos que se irá creando y esto nos ayudara a tener un mejor control de la información de cada uno de nuestros clientes.

La metodología y estándar utilizado en el desarrollo del software será RUP basada en UML lo cual nos proporcionara todas las bases para llevar al éxito la elaboración del software con lo cual se asegura la calidad del producto final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado.

Actualmente la tecnología Java brinda una amplia gama de soluciones para el desarrollo e implementación de sistemas; auxiliada con un manejador de base de datos como MySQL y una solución Hosteada tipo SaaS (Software as Service) se pretende poner en producción el sistema que se utilizará como solución a la problemática de gestión y mejoramiento de las relaciones con el cliente (CRM onDemmand).

Se Realizaran las pruebas necesarias de validación de datos y funcionalidad del sistema para poder liberar una versión integral la cual puede ser implementada bajo los requerimientos necesarios del sistema.

# CAPÍTULO I MARCO METODOLOGÍCO

## 1.1 Planteamiento del problema.

El éxito de una empresa depende fundamentalmente de la demanda de sus clientes. Ellos son los protagonistas principales y el factor más importante que interviene en el juego de los negocios. Si la empresa no satisface las necesidades y deseos de sus clientes tendrá una existencia muy corta. Todos los esfuerzos deben estar orientados hacia el cliente, porque él es el verdadero impulsor de todas las actividades de la empresa. De nada sirve que el producto o el servicio sean de buena calidad, a precio competitivo o esté bien presentado, si no existen compradores.

El mercado ya no se asemeja en nada al de los años pasados, que era tan previsible y entendible. La preocupación era producir más y mejor, porque había suficiente demanda para atender. Hoy la situación ha cambiado. La presión de la oferta de bienes y servicios y la saturación de los mercados obliga a las empresas de distintos sectores y tamaños a pensar y actuar con criterios distintos para captar y retener a esos "clientes" que no mantienen "lealtad" ni con las marcas ni con las empresas.

En la actualidad las empresas se han dado cuenta que es necesario identificar y administrar las relaciones con los clientes, trabajando diferentemente con cada uno de ellos de forma tal de poder mejorar la efectividad. En resumen ser más efectivos al momento de interactuar con los clientes, sin embargo, para el caso de algunas PYMES en México existe la falta de formalización de los procedimientos y la falta de interés de los empleados por la importancia de estos.

El CRM como herramienta de información es de gran ayuda ya que permite mantener una relación de lealtad con los clientes, es decir, tener una relación duradera con los clientes, identificando, comprendiendo y satisfaciendo sus necesidades.

Los principales puntos de por qué tener una herramienta CRM son los siguientes:

- Identificar nuevas oportunidades de negocio.
- Identificar los clientes potenciales que mayor beneficio generen para la empresa.
- La retención de clientes aumenta tras estudiar los datos de gestión de quejas.
- Almacenar en un único ambiente todas las actividades llevadas a cabo con un cliente.
- Mantener un cliente satisfecho requiere de un presupuesto 6 veces inferior al designado a atraer nuevos prospectos.

## **1.2 Objetivos**

### **1.2.1 Objetivo General.**

Analizar, diseñar y programar un sistema informático onDemand, que ayude a las empresas a la toma de decisiones estratégicas para el control de las relaciones con sus clientes.

### **1.2.2 Objetivos específicos.**

- Entender el valor y la gran importancia que es para las empresas lograr la satisfacción de sus clientes.
- Conocer las áreas y la estructura general de cualquier empresa.
- Llevar a cabo el análisis del sistema haciendo uso de la metodología RUP.
- Elaborar el diseño del sistema haciendo uso de la metodología RUP.
- Realizar el diseño de la base de datos haciendo uso de la metodología de normalización de bases de datos. (Empleando la herramienta MySQL WorkBench).
- Construir la base de datos haciendo uso del manejador de bases de datos MySQL.
- Crear un sistema en Java utilizando el Framework de desarrollo de aplicaciones Spring, y Hibernate.
- Desarrollo del módulo para la administración de usuarios (Empresas).
- Desarrollo del módulo para la administración de clientes, así como de sus historiales y tendencias de compra.
- Desarrollo del módulo para la generación de reportes gerenciales que apoyen a la toma de decisiones.
- Desarrollo del módulo de sugerencias y recomendaciones por parte de los clientes.
- Pruebas, correcciones y modificaciones necesarias al sistema.

## **1.3 Justificación.**

La carrera de Ingeniería en Informática aportara a la investigación de este anteproyecto el conocimiento adquirido durante el estudio de la misma en las áreas de lenguajes de programación, análisis y diseño de sistemas de información, así como el manejo de base de datos y las herramientas computacionales necesarias para el desarrollo del sistema.

La percepción y abstracción del problema se realizara mediante el uso de metodologías de investigación correspondientes empleados en la Ingeniería en Informática. Como ingenieros en

informática se analizara, diseñara e implementara un sistema de información estableciendo mecanismos de seguridad en los datos así como la automatización de los procesos organizacionales siguiendo estándares de calidad que permitan la competitividad del sistema.

La carrera de Licenciado en Informática aportara al anteproyecto la administración de los recursos informáticos para garantizar el correcto funcionamiento del sistema empleando herramientas informáticas de acuerdo al alcance planteado.

El licenciado en Informática aplicara los conocimientos interdisciplinarios adquiridos en el uso y manejo de las tecnologías de la información para el análisis, diseño y desarrollo del sistema apoyándose de los lenguajes de programación así como del diseño de base de datos.

Mediante el conocimiento adquirido en la carrera se mantendrán los procedimientos adecuados para estar dentro de la legislación en Informática.

#### **1.4 Tipos de Investigación.**

Para la realización del proyecto nos documentaremos con la información que existe actualmente acerca del desarrollo de un sistema CRM, debido a que partiremos del hecho de que ya están detectadas y definidas ciertas variables correspondientes a nuestro estudio podemos decir que el tipo de investigación es descriptiva.

Durante la investigación del proyecto relacionaremos los conceptos y variables detectados, para ello emplearemos la investigación correlativa.

Una vez que logremos la correlación de las variables detectadas en el estudio, llegaremos a encontrarnos con diferentes formas de abordar la solución del problema, por lo cual emplearemos el método de investigación explicativo.

#### **1.5 Técnicas de investigación a emplear.**

La documentación de la investigación la realizaremos mediante la consulta de libros, revistas, informes técnicos, manuales e información obtenida en la Internet. Se emplearan las entrevistas y cuestionarios, así como la observación directa que es parte de las técnicas del método de investigación descriptiva. Todo esto con la finalidad de obtener información acerca de la problemática.

## **CAPÍTULO II    MARCO TEÓRICO.**

### **2.1    El valor de las buenas relaciones con los clientes.**

Para toda empresa el cliente es una figura que siempre está presente, no es una cifra de ventas a final de mes, es quien demanda los productos y servicios que la empresa ofrece y dar respuesta a sus demandas se hace necesario e imprescindible; el cliente es el activo más importante de cualquier empresa, gracias a los clientes las empresas consiguen consolidarse en el mercado, obteniendo ingresos y rendimientos para posicionarse y sobrevivir. De nada sirve que el producto o el servicio sean de buena calidad, a precio competitivo o esté bien presentado, si no existen compradores.

Siempre preferiremos hacer negocios con aquellas personas con las cuales nos sintamos a gusto, compenetrados, o simplemente, con quienes hayamos logrado desarrollar una buena relación. Cuando establecemos esta conexión con nuestros clientes creamos una sensación de confianza, y es sabido que la gente generalmente gravita hacia aquellas personas en quienes confía.

El crear esta atmósfera es responsabilidad del vendedor, y éste es precisamente uno de los factores que separan al vendedor promedio del vendedor estelar. Uno de nuestros principales objetivos cuando nos encontramos frente a clientes potenciales, es precisamente lograr que ellos se sientan a gusto y que no duden en depositar su confianza en nosotros.

Una relación exitosa con un cliente se basa en que la empresa cambie su comportamiento con ese cliente. Este le indica lo que desea o necesita y la empresa adapta su servicio o personaliza su producto para satisfacer esa necesidad. De esa manera, con cada interacción, ese servicio se acerca cada vez más a las preferencias individuales de ese cliente.

El valor de un cliente, en relación con otros clientes, permite a la empresa priorizar esfuerzos y dedicar más recursos para conseguir que los clientes más valiosos sigan siendo leales y que aumenten aún más su valor, es por eso el esfuerzo de las empresas por mantener y reforzar las relaciones con los clientes actuales. La meta de cualquier empresa debe ser consolidar y mantener una base de clientes comprometidos quienes sean rentables para la organización, para ello se debe de tener en cuenta lo siguiente:

- Es más económico mantener a un cliente actual que atraer a uno nuevo.

- El costo de captar un nuevo cliente es 10 veces el costo de vender a un cliente actual.
- El 50%-60% de los clientes de una empresa no son rentables.
- Las compañías pierden la mitad de sus clientes cada 5 años.
- El 69% de los clientes se cambia por que recibe un servicio inadecuado. Sólo un 13% por insatisfacciones en el producto y un 9% por el precio.

El administrar las relaciones con el cliente, es una filosofía empresarial en la que se plantea entender y anticipar los requerimientos de los consumidores, tanto existentes como potenciales, por lo que puede entenderse como una estrategia de negocios enfocada en el cliente y en sus necesidades.

Debido a que el mercado nacional aún no tiene una oferta amplia para las PYMES, hoy se está presentando una oportunidad importante para que empresas locales ofrezcan software de CRM para ese segmento, el cual podría entregarse principalmente a través de Internet para disminuir el costo. De más está decir que debería ser robusto y fácil de instalar para asegurar su éxito.

## **2.2 CRM.**

El concepto de sistema CRM ha existido toda la vida, desde que el hombre dio inicio a las actividades comerciales o de trueque; la sigla como tal es reciente, pero el concepto detrás es tan antiguo como el comercio mismo. Siempre ha habido compradores y vendedores; siempre los vendedores han tratado de "cautivar" a sus compradores a través de diversos mecanismos. Tal como lo definió Ted Levitt "*la razón de ser de un negocio son los clientes*", sin clientes no hay negocio, sin clientes no hay generación de valor. Lo anterior nos indica que existe un elemento que es el centro de una estrategia CRM y es el cliente.

Customer Relationship Management, CRM por sus siglas en inglés, significa de manera literal la administración de la relación comercial con los clientes.

Un CRM es un proceso continuo que comprende, además del uso de la tecnología, una estrategia de negocios y un cambio de cultura dentro de la organización. Erróneamente en los años 90's las empresas se enfocaban mas en el uso de tecnología para alcanzar los objetivos de un CRM, dejando de lado su núcleo fundamental que es el componente estratégico y las personas. El concepto de CRM implica una serie de cambios en las organizaciones. Uno de éstos se refiere a la integración de áreas que solían trabajar de forma independiente en el modelo convencional, tales

como los sectores de atención, ventas y marketing. Como la nueva estrategia tiene como objetivo el establecimiento de una visión única del cliente, es necesario un entrenamiento interno, además de cambios en los procesos, para que haya sinergia entre las diferentes áreas de la empresa.

Para la pequeña empresa, el tiempo es dinero, tal vez más que para las grandes organizaciones. La mayoría de las pequeñas empresas no tienen el capital y los recursos para financiar los procesos y la interacción con la historia de los clientes, ni pueden permitirse el lujo de contratar a analistas o gastar dinero en grandes campañas de marketing. Cada transacción significa más para una pequeña empresa, y un CRM les permite sacar el máximo provecho de cada cliente y de cada decisión.

Con la implementación de un proceso CRM, principalmente se busca la reducción de costos en la obtención de nuevos clientes e incrementar la lealtad de los que ya se tienen, así como en tener un marketing más efectivo y abrir la posibilidad a nuevas oportunidades de los productos. Los ejecutivos podrán tener una visión de la actividad del grupo y la posibilidad de analizar las actividades de cada individuo o del equipo, así mismo los trabajadores tendrán la posibilidad de ver las actividades de sus colegas con los clientes, trabajando por tanto como un equipo coordinado proporcionando un mejor servicio y clientes más satisfechos.

Para entender al 100% un sistema CRM se deben tener en cuenta las siguientes variables:

1. Variables de recursos humanos: los recursos humanos están en contacto con nuestros clientes, son uno de los aspectos más relevantes que se deben considerar a la hora de mejorar nuestro CRM, especialmente si se trata de una empresa de servicios. Un CRM podría, por ejemplo, mejorar sus procedimientos de atención, sus modales, sus capacidades, entre otros.
2. Variables Tecnologías de Información y Comunicación: las TIC deben dar el sustento tecnológico necesario para que nuestros procesos que se vean involucrados en la mejor relación con clientes de manera que nos permita brindar el 100% de calidad esperada.
3. Procesos: los cuales se deben revisar y mejorar continuamente para lograr que se optimice la relación con los clientes.

### **2.3 Uso e Implementación de la Tecnología en el CRM.**

El concepto de "*Software CRM*" Surge en el año 2000, y rápidamente se convirtió en "concepto del año". Mediante la tecnología suficiente, se comenzó a almacenar los perfiles de cada cliente en bases de datos, para luego hacer un análisis detallado del sector al cual apuntaba la empresa. En

un principio se utilizaba con un propósito meramente de seguimiento, no muy desarrollado a causa de las limitaciones técnicas, pero siempre teniendo en cuenta su importancia.

Las PYMES sufren por grandes cantidades de documentos - debido a cosas como tarjetas de crédito, cuentas de crédito o nuevos pedidos de los clientes - Una PYME puede utilizar software de Gestión de Relaciones con el Cliente para realizar un seguimiento de las transacciones con sus clientes y relacionar esto con su inventario y/o servicios. Los centros de llamadas pueden usar Software CRM para identificar a un cliente al momento de tomar la llamada y rápidamente acceder a su historial de compras, quejas, status de su solicitud y/o pedido, así como las notas anteriores de otros ejecutivos ahorrando tiempo y dinero. Empresas de tamaño medio pueden integrar soluciones de CRM con sus libros contables, los informes trimestrales de los clientes automatizan y eliminan los servicios de terceros.

Un Software CRM puede reducir significativamente la cantidad de servicios proporcionados por terceros (outsourcing) en los aspectos de administración de la relación con los clientes y reducir considerablemente los gastos destinados a esta partida presupuestal.

El software CRM ha contribuido a incrementar el acceso y la confiabilidad de las empresas, el mercado y la información de los clientes. Un CRM permite hacer negocios en línea mucho más rápido y más fácil. Las PYMES han encontrado soluciones CRM para su beneficio - esencialmente - para su éxito, crecimiento, desarrollo y retorno de la inversión (ROI). Las aplicaciones de CRM automatizan las siguientes áreas de negocios:

- Ventas.
- Marketing.
- Servicio al Cliente y Soporte.
- Manejo de la Relación con los Socios.
- Escritorio de Ayuda Interno (Helpdesk).

Por la gran complejidad, no se implanta CRM de una única vez, ni de una forma estándar. Así como los clientes son diferentes, también cada empresa difiere de las demás.

El proceso de implementación de un sistema CRM es el siguiente:

- Determinar las funciones que se desean automatizar.
- Automatizar sólo lo que necesita ser automatizado
- Obtener el soporte y compromiso de los altos niveles de la compañía

- Emplear inteligentemente la tecnología
- Involucrar a los usuarios en la construcción del sistema.
- Realizar un prototipo del sistema.
- Capacitar a los usuarios.
- Motivar al personal que lo utilizará.
- Administra el sistema desde dentro.
- Mantener un comité administrativo del sistema para dudas o sugerencias.

Algunos factores claves a cubrir en una implementación CRM:

Compartir información con los proveedores: de esta manera nos estamos asegurando de que el producto, desde su materia prima, cuente con las características deseadas por el cliente

Determinación de las campañas de mercadeo: se debe identificar cuáles son las formas de mercadeo que realmente llegan a nuestros clientes y cuáles son las que permiten atraer nuevos. Hay que determinar si se pueden realizar campañas de correo directo, de email marketing, de contacto directo en puntos de venta, etc.

Gestión rápida y efectiva de peticiones de servicios y pedidos: a través de la operación de call centers o la automatización de la fuerza de ventas aprovechando la interne.

El manejo de la información para ejecutivos: Permite tener un control de toda la información y una visibilidad total en todo momento y en cualquier lugar para explotar la información.

La excelente sincronización de los datos hay que recordar que la información debe ser confiable, simple, veraz y oportuna

El e-commerce: En los últimos años se ha incrementado el número de personas que acceden a productos o servicios por medio de medios electrónicos, el e-commerce permite abarcar un mayor número de personas.

Los impulsores de CRM no son las utilidades o reducción de costos, sino los aumentos en el valor del cliente. Un proyecto de CRM es una iniciativa que involucra todas las áreas de la empresa. Aunque la tarea comience en un área específica (generalmente en el call center, marketing o ventas) y se desarrolle por etapas, se necesita que haya un planeamiento y una definición de las metas claras a ser alcanzadas. Actualmente el concepto está muy difundido en el mercado y

muchas empresas invierten en la mejora de la relación con el cliente. Sin embargo, algunas compañías que iniciaron proyectos de CRM se frustraron por no obtener los resultados esperados. Las razones que llevan a la falta de éxito son muy variadas. Uno de los errores principales es la falta de la definición de los objetivos y de una estrategia clara desde el comienzo.

Antes que nada es importante que la empresa tenga claro donde pretende llegar para, rápidamente, hacer un análisis sobre sus procesos internos, determinar lo que se necesita alterar para alcanzar las metas establecidas y de qué forma lo hará. Ese trabajo, muchas veces, requiere la ayuda de una consultoría externa que facilitará la realización de ese análisis y la definición de objetivos, para sólo después estipular las etapas del proyecto, lo que deberá alterarse y en caso que se concluya que deberán adoptarse otras tecnologías, cuáles deberán ser, cuáles serán los impactos en cada área, y cómo se medirán los resultados.

Otro error común es el hecho de que muchas compañías se dejan seducir por falsas expectativas y promesas realizadas por los proveedores de soluciones CRM. Muchas veces contratan las consultorías sólo para implementar la tecnología. En esos casos, consiguen obtener algunas ganancias con la automatización de los procesos, pero los resultados no corresponden a las expectativas porque no se realizó una transformación real y efectiva. Un proyecto de CRM requiere cambios en la forma de trabajar, teniendo como foco el cliente. Eso exige la modificación de la cultura interna y una estrategia clara. La mayoría de las veces, el empleo de la tecnología es fundamental, pero sólo esto no es suficiente.

## **2.4 Tipos de CRM.**

CRM OnDemand.

Es un modelo de distribución de software en donde la compañía de tecnologías de información y comunicación (IT) provee el servicio de mantenimiento, operación diaria, y soporte del software usado por el cliente. Regularmente el software puede ser consultado desde cualquier computadora conecta a Internet, esté presente en la empresa o no. Se deduce que la información, el procesamiento, los insumos y los resultados de la lógica de negocio del software están hospedados en la compañía de IT. El software CRM OnDemand a través de una módica cuota mensual por usuario pueden ser utilizadas y en algunos casos podría incluir la propiedad; por ejemplo: en algunos esquemas de hospedaje después de cierto periodo de renta el cliente puede ser el dueño total de la aplicación y si desea puede dejar de hospedar y hacer la instalación en sus equipos.

El software CRM onDemand es rápido de implementar; no hay nada que instalar en ninguna computadora ni servidor. Por lo tanto, se puede poner en marcha un proyecto sin tener que recurrir a un consultor de sistemas. No hay inversión inicial en hardware ni software. Los costos de hosting aumentan en función del crecimiento de su empresa y no se requiere un administrador de sistemas dedicado.

Otro beneficio fundamental es que la información sobre los clientes resulta fácilmente accesible para su personal móvil, con acceso seguro las 24 horas desde cualquier parte del mundo. Sin importar dónde se encuentren sus empleados, siempre tendrán acceso a la información que requieran. Con soporte para dispositivos inalámbricos la información se puede visualizar incluso en equipos PDA.

CRM StandAlone.

Es una versión en intranet en la que toda la información es almacenada por la misma empresa usted es dueño de la aplicación y ésta se encuentra físicamente en sus propias instalaciones. Y, gracias a que usted la tiene en sus manos, tiene acceso directo a los servidores en caso de que desee integrar la solución a otras aplicaciones. Sin embargo, necesita tener personal altamente capacitado y adquirir el hardware y el software necesarios para la implementación de la aplicación. Una vez que una empresa alcanza un tamaño determinado, resulta más lógico desde el punto de vista económico pagar una licencia que estar pagando cargos mensuales por cada usuario. La implementación interna (a cambio de un gasto que se hace una sola vez) también significa que usted puede adaptar el sistema de ventas, mercadotecnia y servicio para que se adapte.

Todas las empresas tienen sus propios sistemas "heredados": las bases de datos, aplicaciones y herramientas Web que han venido utilizando durante algún tiempo. La implementación interna es, sin lugar a dudas, la mejor manera de garantizar que estas aplicaciones puedan integrarse para formar un sistema coherente.

Los sistemas de TI son, en su expresión más evolucionada, entidades orgánicas que crecen y se transforman junto con la empresa, y la posibilidad de cambiar en forma imperceptible entre una solución de CRM interna y una externa ofrece grandes ventajas. Para la mayoría de las compañías en las etapas incipientes de implantación de CRM, una solución contratada externamente funciona bien. Muchas nunca se verán en la necesidad de cambiar a otro esquema. Sin embargo, otras, a fin de poder generar un ROI real, deberán agregar elementos que conviertan al Software CRM en su sistema nervioso central y, en este caso, no bastará con una solución externa.

## 2.5 Clasificación de CRM.

Existen 3 tipos de CRM en el mercado, los cuales son:

CRM Operacional: Involucra todo lo relacionado con el soporte de los procesos de negocios hacia el mundo exterior, que incluye el contacto con los clientes (ventas, marketing y servicios). Las tareas resultantes de estos procesos se asignan a los recursos responsables de las mismas y se les provee de la información necesaria para poder llevarlas a cabo con éxito, además de registrarse en el sistema para poder tener una referencia futura. El CRM Operacional implica los siguientes beneficios:

- Personaliza y aumenta la eficiencia de los procesos de marketing, ventas y servicios a través de la colaboración entre las distintas áreas de la empresa.
- Permite tener una visión de 360 grados de los clientes mientras se interactúa con ellos.
- El sector de ventas y el de atención al cliente pueden acceder al historial completo de las interacciones del cliente con la empresa, sin importar el punto de contacto.

El CRM Operacional involucra típicamente tres áreas generales de negocios:

- SFA - Automatización de Fuerza de Ventas La automatización de fuerza de ventas es una de las funciones críticas de ventas y administración de fuerza de ventas, por ejemplo, la administración de clientes potenciales, administración de contactos, administración de presupuestos, previsión de ingresos, llevar registro de las preferencias de los clientes y sus hábitos de compra. Mejorar la productividad en el área de ventas.
- CSS - Servicio y Soporte al Cliente El soporte y servicio al cliente automatiza los pedidos de servicio, quejas, devolución de productos y solicitudes de información. Las típicas mesas de ayuda y centros de llamados internos para el soporte al cliente ahora evolucionan hacia un CIC - Centro de Interacción con Clientes, utilizando múltiples canales (Web, teléfono / fax, en persona, etc.).
- EMA - Automatización de Marketing Empresarial La automatización del Marketing Empresarial provee información acerca del ambiente del negocio, incluyendo la competencia, tendencias de la industria y variables macro-ambientales. Es la fase de ejecución de la administración de las campañas y administración de los clientes potenciales. La intención de las aplicaciones de automatización de marketing empresarial es permitir aumentar la eficiencia de las campañas de publicidad. Las funciones incluyen el análisis demográfico, segmentación variable y modelos predictivos.

Por tanto, un CRM Operacional es aquel que permite realizar las tareas cotidianas de una empresa, a manera de enfocar nuestros datos desde el punto de vista del cliente.

Esto implica funcionalidades como:

1. Almacenamiento de datos de Contactos (personas individuales) y Cuentas (Grupos de Personas, Organizaciones, Empresas): Al contar con estos datos actualizados dentro de un sistema, puedo realizar tareas de marketing, segmentación de datos, rápida identificación de los clientes (por nombre personal, por nombre de empresa, por su ubicación, etc.) y así agilizar toda la operación relacionada con los datos de nuestros clientes.
2. Almacenamiento de datos de Comunicación e Interacción con el Cliente: El sistema se encarga de almacenar (siempre relacionando estos datos con un cliente en particular) todos los datos de Llamadas Telefónicas, Reuniones Presenciales, Tareas Realizadas, Notas, Mensajes de Correo Electrónico. De esta forma, siempre podemos saber que fue lo último que se conversó con un cliente en particular, ya sea en una reunión, de forma telefónica o por mail.
3. Seguimiento de Casos de Soporte a Clientes: Si los productos o servicios que comercializamos involucran un seguimiento post-venta, esta funcionalidad permite realizar un seguimiento de los llamados por pedido de soporte, su estado y su resolución, para ir conformando una Base de Conocimiento interna de la empresa, que permita identificar posibles fallas reiteradas y responder de forma consistente a las mismas.
4. Identificación de Interesados y Prospectos: Dentro de la jerga del marketing, se llama prospectos a todos aquellos contactos que reúnen ciertas características que pueden llegar a ser compatibles con los intereses comerciales de nuestra empresa. De esta forma, un CRM Operacional permite almacenar datos de posibles clientes (que hayamos conseguido mediante formularios completos en una exposición, una compra de una base de datos, etc.) y luego identificar su posible interés en nuestros productos/servicios para avanzar con las gestiones comerciales, ya en carácter de Interesados.
5. Información de Oportunidades Comerciales: Una vez que un Prospecto demuestra interés en nuestros productos, el CRM Operacional se encarga de proveernos de las herramientas para poder almacenar los datos relacionados con las etapas que este interés implica (Pedido de Presupuesto, Confirmación de Pedido, Renegociación, Anulación de Pedidos, etc.), su aspecto monetario (de cuánto dinero estamos hablando) y quién es el encargado de llevar adelante la venta. De esta manera podremos obtener reportes basados en estos datos, individualizando aquellos vendedores más exitosos, aquellos clientes más compradores, en que momentos debemos reforzar nuestra tarea de ventas, etc.

CRM Analítico: Con el CRM Analítico, se analizan los datos obtenidos con el CRM operacional o mediante otras fuentes, para segmentar a los clientes o identificar relaciones con otros clientes potenciales. El análisis de clientes típicamente puede llevar a campañas dirigidas de Marketing para incrementar las ventas. Ejemplos de campañas dirigidas hacia los clientes son:

- Adquisición: Cross-sell, up-sell.
- Retención: Retener clientes que abandonan la empresa debido al desgaste o cambio de expectativas.
- Información: Permitiendo entregar información en forma frecuente y regular a los clientes.
- Modificación: Alterando los detalles de las transacciones en las relaciones con los clientes.

El análisis de los datos puede estar relacionado con:

- Análisis y administración de campañas.
- Optimización del canal de contactos.
- Optimización de los contactos con los clientes.
- Adquisición / Reactivación / Retención de clientes.
- Segmentación de clientes.
- Incremento / Medición de la satisfacción de los clientes.
- Optimización del alcance de las ventas.
- Análisis y detección de fraudes.
- Desarrollo de productos.
- Evaluación de Programas.
- Administración y mitigación de riesgos.

El análisis y recolección de datos se considera una tarea continua e iterativa. Idealmente, las decisiones de negocios se van refinando con el tiempo, basándose en la respuesta obtenida en base a las decisiones y análisis previos.

CRM Colaborativo: El CRM colaborativo facilita las interacciones con los clientes a través de todos los canales (personal, correo, fax, teléfono, Web, e-mail) y da soporte a la coordinación de los equipos de empleados y canales. Es una solución que unifica los recursos humanos, procesos y datos para que la empresa pueda servir mejor a sus clientes. Los datos y actividades pueden ser estructurados, desestructurados, conversacionales y / o de naturaleza transaccional.

El CRM colaborativo provee los siguientes beneficios:

- Habilita las interacciones eficientes y productivas con los clientes a través de todos los canales de comunicación.
- Habilita la colaboración vía Web para reducir los costos de atención al cliente.
- Integra los call-centers, habilitando la interacción con los clientes a través de múltiples personas.
- Integra la visión del cliente mientras se continúa la interacción a nivel transaccional.

Uno de los aspectos de esta estrategia de CRM, es utilizar un sistema informático como repositorio de todos los datos (mensajes de correo electrónico, llamadas telefónicas, registro de reuniones, etc.) que se generan como resultado de la interacción con nuestros clientes. Esto, obviamente, tiene sus ventajas desde varios puntos de vista.

Al contar con un sistema centralizado en un CRM, una de las ventajas principales es que los clientes pueden utilizar múltiples canales de comunicación de forma independiente. Pensemos por un momento, al cliente realmente no debería importarle con qué operador o por qué medio (fax, Web, teléfono, email) realizó su orden de compra con nuestra empresa. De esta forma, cualquier operador podrá contar con los datos relacionados con el cliente y poder asistirlo contando con toda la información.

Un detalle a tener en cuenta en este último aspecto, es la independencia que la organización logra de sus integrantes al implementar un CRM. Es conocido el conflicto que nos puede ocasionar la ausencia temporal (por razones personales, enfermedad) o renuncia definitiva de un vendedor.

Cuando el cliente vuelve a contactarse con nosotros, nadie sabe qué fue lo último que se conversó, a qué acuerdo se llegó, o peor aún, la organización tampoco tiene idea de cuales era los contactos que este vendedor manejaba. Si logramos implantar la disciplina de utilizar el sistema de CRM como herramienta de trabajo cotidiano, podremos quedar a resguardo de estos inconvenientes. Es muy importante la percepción que el cliente tenga de nuestra organización. Si cada vez que nos llama, podemos brindarle una solución, porque contamos con todos sus datos a mano, sabemos qué fue lo último que conversaron o acordaron con él, aumentaremos así la confianza que los mismos tienen en nosotros. Y este es un factor clave al momento de concretar una venta, cimentar una sólida relación con un cliente y continuar extrayendo valor de esta relación en el futuro.

Otra ventaja es poder hacer una evolución del caso de un mismo cliente de manera transversal en nuestra organización. Dado que la relación con un cliente no se termina al momento de realizar una compra, es muy importante poder hacer un seguimiento de nuestra relación a medida que

pasa el tiempo. Si el departamento de atención post-venta, soporte o cuidado del cliente no conoce qué productos compró el cliente, con qué fecha y qué servicios adicionales ha adquirido, se dificulta su correcta atención.

Una herramienta importante que ponen a nuestra disposición los sistemas de CRM respecto de la colaboración, es la asignación de registros (reuniones, llamadas, contactos, cuentas, etc.) a integrantes de la empresa. De esta manera, un gerente de ventas no solo puede realizar un seguimiento de los prospectos que tiene en cartera un vendedor, sino también asignar nuevas tareas, cuentas o contactos a integrantes de un determinado equipo de ventas.

Con el tiempo y a medida que la organización decide alinearse dentro del proyecto de implementación del CRM, el repositorio de datos se hace más completo y conciso. Todos los integrantes del equipo de trabajo utilizan la herramienta adecuadamente y podremos ingresar a una nueva etapa dentro de la implementación de esta estrategia dentro de la organización: el análisis de datos para su explotación comercial.

## **CAPÍTULO III METODOLOGÍA Y TECNOLOGÍA APLICADA.**

### **3.1 Introducción al RUP.**

RUP *Rational Unified Process*. Es una metodología de desarrollo de software que intenta integrar todos los aspectos a tener en cuenta durante el ciclo de vida del software, con el objetivo de hacer abarcables tanto pequeños como grandes proyectos software.

Dentro del proceso de ingeniería de *software* proporciona un enfoque disciplinado de asignación de tareas y responsabilidades en la organización del desarrollo. Su meta es asegurar la producción del *software* de alta calidad proporcionando herramientas para todos los pasos del desarrollo así como documentación en línea.

### **3.2 Características de RUP.**

1. Manejado por casos de uso: Los casos de uso constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.
2. Centrado en la arquitectura: Se involucra elementos significativos del sistema como son plataformas del software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Se pretende que sea lo suficientemente específica como para que todos los implicados en el desarrollo tengan una idea clara de qué es lo que están construyendo, pero lo suficientemente simple como para sustituir parte importante del sistema sin realizar grandes cambios. Representándose mediante las vistas lógicas, de implementación, proceso y despliegue, más la de casos de uso que es la cohesión de todas. Centrándose en aspectos concretos del sistema.
3. Iterativo e Incremental: Un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como proyectó aislado constituido por una o más iteraciones de las actividades principales básicas de

cualquier proceso de desarrollo. En concreto RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

4. Desarrollo basado en componentes: La división del sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o que se desarrollen y maduran sus componentes.

### **3.3 Fases RUP.**

1. Concepción, Inicio o Estudio de oportunidad

Define el ámbito y objetivos del proyecto así como la funcionalidad y capacidades del producto

2. Elaboración

La funcionalidad como el dominio del problema se estudia a profundidad y se define una arquitectura básica planificando el proyecto considerando recursos disponibles

3. Construcción

El producto se desarrolla a través de iteraciones involucrando tareas de análisis, diseño e implementación. Gran parte del trabajo es programación y pruebas documentando el sistema y el manejo del mismo proporcionando un producto construido.

4. Transición

En esta etapa se libera el producto y se entrega al usuario para un uso real. Se incluyen tareas de *marketing*, empaquetado, instalación, configuración, capacitación, soporte, mantenimiento, etc. Los manuales de usuario se completan y refinan con la información anterior

### **3.4 Disciplinas RUP.**

Las disciplinas conllevan los flujos de trabajo, los cuales son una secuencia de pasos para la culminación de cada disciplina. A continuación se describe rápidamente cada una de estas disciplinas:

a) Modelado del negocio

Comprende la estructura y la dinámica de la organización, así como los problemas actuales e identifica posibles mejoras, de los procesos del negocio. Utiliza los Diagramas de Actividad y de Clases.

b) Requerimientos

Establece y especifica los requisitos del sistema, define los límites del sistema, y una interfaz de usuario, realizando una estimación del costo y tiempo de desarrollo. Para modelar el Sistema y los componentes que lo comprenden, como lo son actores y relaciones, utiliza los diagramas de Estados de cada y las especificaciones suplementarias.

c) Análisis y diseño

Define la arquitectura del sistema y traslada requisitos en especificaciones de implementación, mediante el diseño refina el análisis para poder implementar los diagramas de clases, los diagramas de colaboración, el de clases, el de secuencia, el de estados, y el modelo de la arquitectura.

d) Implementación

Implementa las clases de diseño como componentes transformándolo en código fuente, asigna y prueba los componentes individualmente, integra los componentes en un sistema ejecutable. Utiliza los diagramas de componentes.

e) Pruebas

Verifica la integración de los componentes, que todos los requisitos han sido implementados y asegurar que los defectos detectados han sido resueltos antes de la distribución.

f) Despliegue

Asegura que el producto está preparado para el cliente, procede a su entrega y recepción por el cliente. En esta disciplina se realizan las actividades de probar el *software* en su entorno final, empaquetarlo, distribuirlo e instalarlo, así como la capacitación.

g) Gestión y configuración de cambios

Es un apoyo al control de cambios evitan confusiones costosas como la compostura de algo que ya se había arreglado y aseguran que los resultados no entren en conflicto con alguna de los siguientes actividades.

- Actualización simultánea: Es la actualización de algo elaborado con anterioridad, sin saber que alguien más lo está actualizando.
- Notificación limitada: Al realizar alguna modificación, no se deja información sobre lo que se hizo, por lo tanto no se sabe quien, como, y cuando se hizo.
- Versiones múltiples: No saber con exactitud, cual es la última versión, y al final no se tiene un orden sobre que modificaciones se han realizado a las diversas versiones.

h) Gestión del proyecto

Planea, dirige al personal, ejecutar acciones y supervisar proyectos. Administrando el riesgo.

i) Entorno

Se enfoca sobre las actividades necesarias para configurar el proceso que engloba el desarrollo de un proyecto y describe las actividades requeridas, provee a la organización que desarrollará un ambiente en el cual basarse, el cual provee procesos y herramientas para poder desarrollar el *software*.

### 3.5 Introducción al UML.

UML surge como respuesta al problema de contar con un lenguaje estándar para el modelado de *software*.

UML representa la unificación de las notaciones de los métodos Booch, *Objectory* (Ivar Jacobson) y OMT (James Rumbaugh) siendo su sucesor directo y compatible.

En Septiembre de 2001 se ha publicado la especificación de la versión 1.4. Es importante recalcar que sólo se trata de una notación, es decir, de una serie de reglas y recomendaciones para representar modelos. UML no es un proceso de desarrollo, es decir, no describe los pasos sistemáticos a seguir para desarrollar *software*. UML sólo permite documentar y especificar los elementos creados mediante un lenguaje común describiendo modelos.

### **3.6 Descripción del lenguaje UML.**

UML es un lenguaje de propósito general para el modelado orientado a objetos, combinando notaciones del Modelado Orientado a Objetos, Modelado de Datos, Modelado de Componentes, Modelado de Flujos de Trabajo(*Workflows*).

UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

El UML modela sistema mediante el uso de objetos que forman parte de él así como, las relaciones estáticas o dinámicas que existen entre ellos.

UML es un Lenguaje de Modelado Unificado basado en una notación gráfica la cual permite: especificar, construir, visualizar y documentar los objetos de un sistema programado.

En todos los ámbitos de la ingeniería se construyen modelos, en realidad, simplificaciones de la realidad, para comprender mejor el sistema que vamos a desarrollar.

Un enfoque sistemático permite construir estos modelos de una forma consistente demostrando su utilidad en sistemas de cierto tamaño. Cuando involucramos a varios desarrolladores trabajando y compartiendo información, el uso de modelos y el proporcionar información sobre las decisiones tomadas, es vital no sólo durante el desarrollo del proyecto, sino una vez finalizado éste, cuando se requiere algún cambio en el sistema.

### 3.7 Descripción de los diagramas UML.

Un modelo es la abstracción del mundo real, considerando cierto propósito. Así, el modelo describe completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle.

Un diagrama es una representación gráfica de una colección de elementos de modelado, generalmente representado con vértices conectados por arcos.

Un proceso de desarrollo de *software* debe ofrecer un conjunto de modelos que permitan expresar el producto desde cada una de las perspectivas de interés. Es aquí donde se hace evidente la importancia de UML en el contexto de un proceso de desarrollo de *software*.

- a) Diagrama de Casos de Uso: modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.
- b) Diagrama de Clases: muestra las clases (descripciones de objetos que comparten características comunes) que componen el sistema y cómo se relacionan entre sí.
- c) Diagrama de Objetos: muestra una serie de objetos (instancias de las clases) y sus relaciones.
- d) Diagramas de Comportamiento: dentro de estos diagramas se encuentran:
  - e) Diagrama de Estados: modela el comportamiento del sistema de acuerdo con eventos.
  - f) Diagrama de Actividades: simplifica el Diagrama de Estados modelando el comportamiento mediante flujos de actividades.
- g) Diagramas de Interacción: Se dividen en 2 tipos, según la interacción que enfatizan:
  - Diagrama de Secuencia: enfatiza la interacción entre los objetos y los mensajes que intercambian entre sí junto con el orden temporal de los mismos.
  - Diagrama de Colaboración: muestran la interacción entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.
- h) Diagramas de implementación
  - Diagrama de Componentes: muestra la organización y las dependencias entre un conjunto de componentes.
  - Diagrama de Despliegue: muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo.

### **3.8 Introducción al J2SE.**

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los 90's. A diferencia de los lenguajes de programación convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado en bytecode que es interpretado por una máquina virtual Java.

Java es compatible con todos los sistemas porque basa su funcionamiento en los bytecode, que no es más que una pre-compilación del código fuente de Java. Cada sistema operativo dispone de una Máquina Virtual de Java que puede interpretar los bytecode y transformarlos a sentencias ejecutables en el sistema en cuestión.

### **3.9 Tecnología J2SE.**

J2SE, es una plataforma creada por SUN en la que se ofrece perspectivas de desarrollo para empresas que quieran basar su arquitectura en productos basados en *software* libre.

J2SE (Java 2 Standard Edition) es, por decirlo de alguna manera, la base de la tecnología Java. Permite el desarrollo de applets (aplicaciones que se ejecutan en un navegador Web) y aplicaciones independientes (standalone), necesarias para construir aplicaciones robustas.

En cuanto a las aplicaciones distribuidas surge J2EE, versión ampliada de la J2SP que incluye las API necesarias para construir aplicaciones para arquitecturas distribuidas, permitiendo la construcción de sistemas transaccionales basados en tecnologías Web.

Las aplicaciones se componen de software independiente ensamblado con otros componentes a través de canales definidos.

La especificación Java define los siguientes componentes:

1. Aplicaciones cliente. Son componentes que se ejecutan en el cliente. Una aplicación cliente normalmente se utiliza para mostrar una interfaz gráfica al usuario. Interfaz gráfica del usuario

para interactuar con la aplicación, normalmente basada en un explorador o en una aplicación cliente. Mediante el uso del navegador, se facilita la programación de la interfaz algunos controles y la gestión de eventos vienen como parte del navegador, hay que validar que sea aceptable en distintos navegadores y versiones de los mismos. El diseño de la interfaz está limitado a las características que se puedan programar con el lenguaje HTML (Hypertext Markup Language) o XML (Extensible Markup Language) utilizado, y su ejecución depende del servidor.

2. Servlets y Java Server Pages (JSP). Componentes Web que se ejecutan en un servidor. Interfaz entre un componente y el sistema de nivel inferior que da soporte al componente. Se encarga de la persistencia de los datos, la gestión de recursos, la seguridad, los hilos y otros servicios a nivel de sistema para los componentes asociados con él, estos son responsables de implementar la lógica del negocio.
3. Enterprise Java Beans (EJB). Componentes que se ejecutan en un servidor Java, concretamente en el contenedor EJB del mismo, y que contiene las reglas del negocio.

Hay tres clases de Enterprise beans:

- Beans de sesión. Representa una conversación transitoria con un cliente; cuando termina la ejecución del cliente, el bean de sesión finaliza sin guardar sus datos.
  - Bean de entidad. Representa datos almacenados en una base de datos; si el cliente termina o cierra la conexión el servidor, los servicios subyacentes aseguran que los datos del bean se guarden.
  - Bean orientado a mensajes. Combina características de un bean de sesión y un JMS (Java Message Service), permitiendo que un componente de negocio pueda recibir mensajes JMS de forma asíncrona.
4. Servidor. Tomcat es un servidor Web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache. Tomcat es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. Tomcat es un

contenedor de Servlets con un entorno JSP. Un contenedor de Servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario.

Podemos dividir los contenedores de Servlets en:

- Contenedores de Servlets Stand-alone (Independientes). Estos son una parte integral del servidor Web. Este es el caso cuando usando un servidor Web basado en Java, por ejemplo, el contenedor de servlets es parte de Java Web Server. Este es el modo por defecto usado por Tomcat.

Sin embargo, la mayoría de los servidores, no están basados en Java, los que nos lleva los dos siguientes tipos de contenedores:

- Contenedores de Servlets dentro-de-Proceso. El contenedor Servlet es una combinación de un plugin para el servidor Web y una implementación de contenedor Java. El plugin del servidor Web abre una JVM (Máquina Virtual Java) dentro del espacio de direcciones del servidor Web y permite que el contenedor Java se ejecute en él. Si una cierta petición debería ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando JNI). Un contenedor de este tipo es adecuado para servidores multi-thread de un sólo proceso y proporciona un buen rendimiento pero está limitado en escalabilidad
- Contenedores de Servlets fuera-de-proceso. El contenedor Servlet es una combinación de un plugin para el servidor Web y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor Web. El plugin del servidor Web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP). Si una cierta petición debería ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando IPCs). El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad, etc.).

### 3.10 Ventajas de J2SE

- Soporte de múltiples sistemas operativos: Al ser una plataforma basada en el lenguaje *Java*, es posible desarrollar arquitecturas basadas en J2SE utilizando cualquier sistema operativo donde se pueda ejecutar una máquina virtual *Java*.

- Organismo de control: La plataforma J2SE está controlada por el JCP, un organismo formado por más de 500 empresas. Entre las empresas que lo forman están todas las más importantes del mundo informático (SUN, IBM, *Oracle*, SAP, HP, AOL, etc.) lo que garantiza la evolución de la misma.
- Competitividad: Muchas empresas crean soluciones basadas en J2SE y que ofrecen características como rendimiento, precio, etc., muy diferentes. De este modo el cliente tiene una gran cantidad de opciones a elegir.
- Madurez: Creada en el año 1997 como respuesta a la tecnología MTS de *Microsoft*, J2SE tiene ya cinco años de vida y una gran cantidad de proyectos importantes a sus espaldas.
- Soluciones libres: En la plataforma J2SE es posible crear arquitecturas completas basadas única y exclusivamente en productos de *software* libre. No sólo eso, sino que los arquitectos normalmente disponen de varias soluciones libres para cada una de las partes de su arquitectura.

### 3.11 Desventajas de J2SE

Aún así, la plataforma de J2SE también tiene desventajas, algunas importantes:

- Depende de un único lenguaje: La plataforma J2SE depende exclusivamente del lenguaje *Java*. Sólo se puede utilizar este lenguaje para desarrollar aplicaciones lo que puede suponer un gran problema si nuestro equipo no dispone de los conocimientos suficientes o tiene otras preferencias.
- Complejidad: Aunque no es una plataforma tan compleja como CORBA, no existe un VB .NET en *Java*. La creación de aplicaciones bajo J2SE requiere normalmente desarrolladores más experimentados que los necesarios para desarrollar bajo .NET.
- Heterogeneidad: Existe una gran heterogeneidad en las soluciones de desarrollo. No existe en J2SE un símil a Visual Studio .NET. La gran cantidad de herramientas disponibles causa confusión dentro de los desarrolladores y puede crear dependencias dentro de las empresas.

Existe mucha confusión, sobre todo entre la gente alejada del mundo de *Java*, sobre lo que es en realidad J2SE. La confusión más habitual es pensar que J2SE es un producto concreto que distribuye SUN *Microsystems* (ORACLE) y que se puede descargar desde su página *Web*. Nada más lejos de la realidad. No existe un J2SE concreto, no se puede ir a la página *Web* de SUN *Microsystems* (ORACLE) y descargar "el J2SE".

### 3.12 Integración con otros sistemas

En cuanto a la integración con otros sistemas, gran parte del modelo de J2SE (en especial los EJB) está basado en CORBA lo que quiere decir que es posible comunicarse sin ningún tipo de problema con otras aplicaciones creadas en otros lenguajes diferentes de *Java* y viceversa. Las posibilidades de integración todavía son mayores ya que cualquier EJB que hayamos creado se puede exponer como un servicio *Web*. Por último otra tecnología que merece la pena nombrar es la de los conectores, que por explicarlo de una manera clara, son un puente entre J2SE y sistemas *legacy* (por ejemplo un conjunto de ficheros de datos en COBOL), Los conectores tienen un API estándar que nos permite acceder a estos sistemas *legacy* de una manera transparente y sin apenas esfuerzo.

Otra de las grandes ventajas de J2SE viene del hecho de que está basado en *Java*, ya que la variedad de clientes que pueden conectarse a una arquitectura J2SE es inmensa, desde aplicaciones de escritorio, hasta móviles, pasando por televisiones, PDAs, etc., cualquiera de estos productos puede conectarse ya sea a través de cualquier protocolo de comunicación lo que nos da una gran gama de posibilidades en cuanto a la conectividad de nuestros sistemas.

### **3.13 Arquitectura de múltiples capas.**

La plataforma de Java define un modelo de programación encaminado a la creación de aplicaciones basadas en n-capas. Típicamente una aplicación puede tener cinco capas diferentes:

- Capa de cliente: Representa el interfaz de usuario que maneja el cliente.
- Capa de presentación: Representa el conjunto de componentes que generan la información que se representará en el interfaz de usuario del cliente. Típicamente se creará a través de componentes basados en *Servlets* y *JSP*.
- Capa de lógica de negocio: Contiene nuestros componentes de negocio reutilizables. Normalmente se forma a partir de componentes EJB.
- Capa de integración: Aquí se encuentran componentes que nos permiten hacer más transparente el acceso a la capa de sistemas de información. Lugar idóneo para implementar una lógica de objetos de acceso a datos, DAO (*Data Access Objects*).
- Capa de sistemas de información: Esta capa engloba a nuestros sistemas de información: bases de datos relacionales, bases de datos orientadas a objetos, sistemas *legacy*, etc.

## **CAPÍTULO IV ANÁLISIS Y DISEÑO DEL SISTEMA.**

### **4.1 Identificación de requerimientos.**

Hemos encontrado como una verdadera necesidad para toda empresa, no importando el tipo de giro, el contar con una herramienta que le ayude no solo a manejar de una mejor manera la cartera de clientes; sino que también le ayude a crear y mantener una buena relación con los mismos.

Debido a que el manejo de información de los clientes es más complejo a medida que crece la empresa, se requiere poder dar seguimiento a cualquier incidente que pueda generarse en el transcurso de dicha relación, no dependiendo de alguna persona, sino que la información se encuentre centralizada, que todos la alimenten y que todos se puedan servir de ella; manejando perfiles y niveles de seguridad que garanticen el acceso a dicha información.

Parte indispensable de esta herramienta será la administración y control de clientes, por ello se brindara a los diferentes perfiles un modulo de administración de clientes que permita conocer todo el historial de un cliente así como el personal de la empresa que tuvo asignado al momento de ser levantada la orden de servicio.

Según las necesidades del negocio, contaremos además con un modulo de manejo de campañas publicitarias y un modulo de reportes que permita el análisis del negocio desde una perspectiva integral, lo cual será alimentado por la operación diaria, se contara con una base de datos de clientes actuales y potenciales de donde se obtendrán los datos de contacto, como son: teléfono, email y dirección; piezas clave para las campañas.

### **4.2 Análisis de requerimientos.**

Se presenta un análisis que incluye los requerimientos captados así como otras especificaciones que servirán para la implementación del sistema. Aquí se estarán aplicando un Desarrollo en el ámbito de Análisis del sistema.

- Requerimientos Funcionales.

A continuación se presenta una lista de requerimientos de los cuales se partió para la implementación del sistema.

1. El sistema tendrá que ser accesible desde cualquier ubicación, garantizando el ingreso al sistema 24/7.
2. Permitir el acceso a los módulos correspondientes del sistema de acuerdo al perfil del usuario.
3. Administración de las empresas que son clientes y/o prospectos.
4. Administración de los contactos que se tienen para establecer relaciones comerciales con los clientes y/o prospectos.
5. Administración de los usuarios que pueden tener acceso al sistema.
6. Registrar y dar seguimiento a las peticiones de los clientes sin importar el vendedor asignado. Informando a los usuarios las actividades que tienen pendientes mostrando las de mayor prioridad. Asignar las actividades a los departamentos y/o empleados que el ejecutivo de cuenta considere pertinentes para la atención de las actividades solicitadas por los clientes.
7. Envío de publicidad y comunicados de interés general para los clientes.
8. Generación de reportes para la toma de decisiones.

- Requerimientos No Funcionales.

El Sistema deberá ser desplegado en un ambiente J2SE, corriendo en un dominio público y depositando sus componentes en un Contenedor *Web*, Contenedor *EJB* y una base de datos.

De la parte de Arquitectura solo se mostraran los diagramas, así como características necesarias para comprender la solución.

1.1 El acceso al sistema será vía *Web*.

1.2 El servidor de aplicaciones a utilizar será open source.

1.3 Se tendrá el servidor dentro de instalaciones que cuentan con redundancia en electricidad e Internet.

2.1 Se proporcionara un acceso generalizado para asociar los datos del usuario con una empresa.

2.2 El sistema validara que el usuario este previamente registrado por el administrador de la aplicación.

2.3 La contraseña tendrá un mínimo de 8 caracteres.

2.4 Después de 3 intentos fallidos de login la cuenta quedara deshabilitada.

- 2.5 Se mantendrá un registro de los usuarios que se encuentran dentro del sistema para evitar accesos concurrentes.
- 2.6 El sistema restringirá el acceso a cada modulo dependiendo del rol que tenga el usuario.  
Se manejaran sesiones para cada usuario logeado en el sistema.
- 3.1 El sistema permitirá el ingreso a este modulo a todos los perfiles.
- 3.2 Mediante el sistema se podrá dar de alta, modificar y deshabilitar empresas
- 3.3 El sistema desplegara un listado máximo de 10 registros en pantalla.
- 3.4 En el sistema se generaran formularios para la captura de los datos de las empresas y sus contactos, algunos campos serán llenados dinámicamente mediante catálogos.
- 3.5 Se mostrara una “consulta maestra” donde se listara la información de interés general de las empresas.
- 4.1 Cada empresa es responsable de la administración de sus clientes.
- 4.2 El sistema permitirá el ingreso a este modulo a todos los perfiles.
- 4.3 Mediante el sistema se podrá dar de alta, modificar y deshabilitar contactos que se encuentren asignados a una empresa.
- 4.4 El sistema desplegara un listado máximo de 10 registros en pantalla.
- 4.5 En el sistema se generaran formularios para la captura de los datos de los contactos, algunos campos serán llenados dinámicamente.
- 4.6 Se mostrara una “consulta maestra” donde se listaran los contactos y se mostrara información de interés general.
- 5.1 Cada empresa es responsable de la administración de sus usuarios y control de accesos de acuerdo a los perfiles que sean establecidos.
- 5.2 El sistema solo permitirá el ingreso a este modulo a los usuarios que tengan el perfil de Administrador.
- 5.3 Mediante el sistema se podrá dar de alta, modificar y deshabilitar usuarios.
- 5.4 El sistema desplegara un listado máximo de 10 registros en pantalla.
- 5.5 En el sistema se generaran formularios para la captura de los datos de los usuarios, algunos campos serán llenados dinámicamente.
- 5.6 Se mostrara una “consulta maestra” donde se listaran los usuarios y se mostrara información de interés general.
- 6.1 El sistema contara con un modulo de órdenes para el seguimiento de la atención al cliente.
- 6.2 Los parámetros ingresados en la orden, serán asociados a un consultor y a un contacto de la empresa.
- 6.3 Todos los perfiles podrán consultar el detalle de cada orden.

- 6.4 El consultor será notificado de cada nueva orden asignada.
- 6.5 El sistema permitirá saber el estatus de cada orden hasta dar una solución al cliente
- 6.6 El sistema permitirá manejar en un solo modulo los diferentes tipos de órdenes que puede realizar un cliente (Nueva venta, Tarea, Incidencia, Cita o Llamada telefónica).
- 5.1 Se generara un script de consulta a nivel BD de los contactos, para explotar los datos personales y hábitos de compra.
- 5.2 El sistema asociara a cada campaña creada una lista de contactos/clientes de acuerdo a la segmentación de mercado (Tipo de cliente) que desee el usuario.
- 5.3 El sistema podrá enviar publicidad mediante correo electrónico a la lista de contactos que se encuentran asociados a la campaña.
- 5.4 Se tendrá que utilizar un servidor con IP fija que permita montar servicios de correo electrónico y DNS.
- 8.1 Para la generación de reportes se maneja el software JasperReports
- 8.2 Los reportes se podrán generar por rango de fechas.
- 8.3 Cada reporte desplegara gráficamente un resumen de la información.
- 8.4 El formato empleado por el sistema para mostrar los reportes será PDF
- 8.5 El sistema generara los reportes dinámicamente en base a los datos de la operación diaria

#### **4.3 Stakeholder y descripciones de usuarios.**

Los Stakeholder, son las personas u organizaciones que están directamente envueltas en la elaboración o tomas de decisiones claves a cerca de la funcionalidad y propiedades del Sistema. Para este sistema los *Stakeholders* son: Gerente de la Empresa y Desarrolladores del Sistema.

Los usuarios son las entidades individuales que utilizan el Sistema, estos son:

*Cliente* (Usuario del Sistema, Usuario Autenticado, Cliente).- Persona que contrata servicios o adquirirá bienes a un proveedor.

*Ejecutivo de Cuenta*.- Personal que será asignado para que atienda a un cliente en particular, en todo lo que requiera.

*Supervisor*.- Personal Autorizado a gestionar las actividades de sus subalternos dentro de una organización.

*Gerente*.- Ejecutivo de una organización que tiene autoridad para tomar decisiones.

*Empleado*.- Personal encargado de realizar actividades operativas en una organización.

*Marketing*.- Área de Mercado encargada de manejo de publicidad en una organización.

*Analista*.- Personal encargado de obtener estrategias adecuadas a partir del estudio de mercado.

*Sistema*.- Conjunto de programas interrelacionados que nos permitirán obtener el resultado deseado.

*Administrador*.- Personal designado para la gestión de los parámetros.

#### 4.4 Ambiente a utilizar

Se asume que los usuarios ingresaran y utilizarán el Sistema vía *Internet* mediante un navegador *Web*. También se asume que el Administrador trabajará dentro de una red privada (la misma donde estará instalado el sistema de servidores).

#### 4.5 Casos de uso.

Se identificaron los siguientes casos de uso:

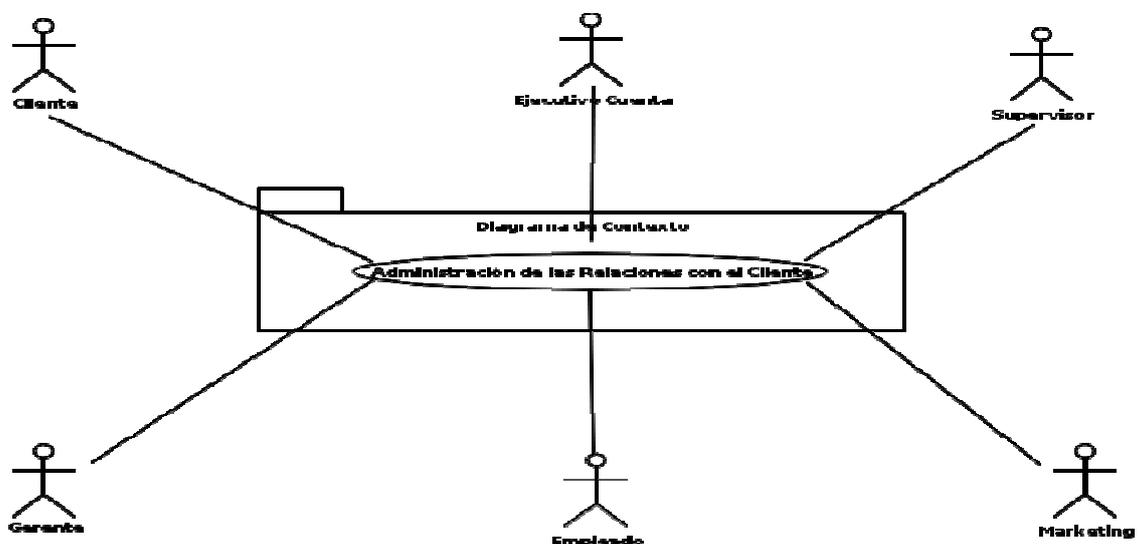


Imagen 4.1 Diagrama de contexto.

Actualmente la mayoría de las PYMES no cuentan con alguna herramienta informática CRM, por ser soluciones demasiado robustas o costosas; por lo que recurren al conocimiento a corto plazo que tengan del cliente y las empresas más sofisticadas a hojas de cálculo.

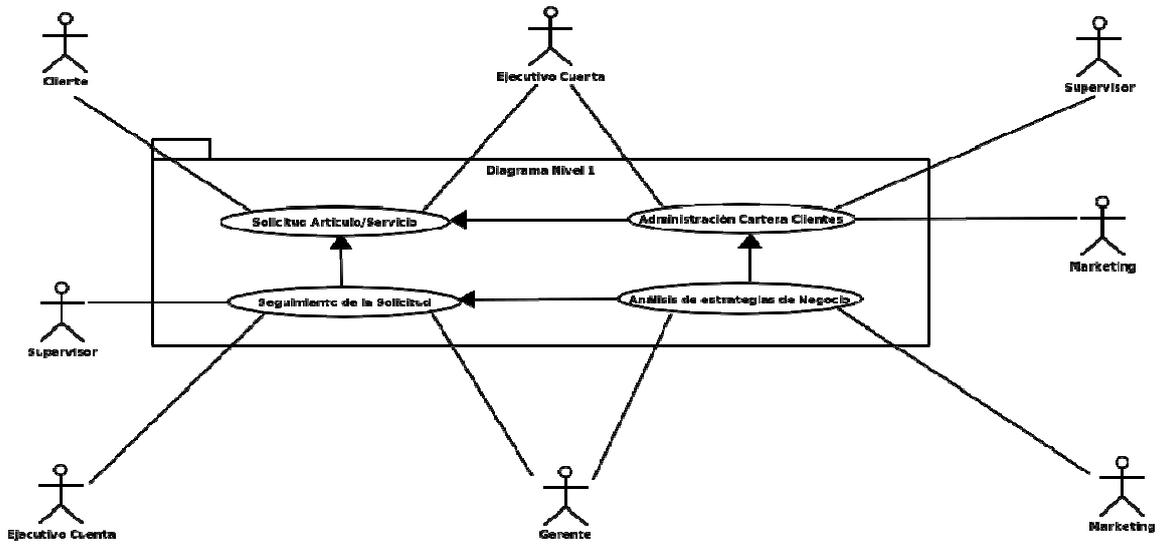


Imagen 4.2 Diagrama de nivel 1

La administración de las relaciones con el cliente, se lleva a cabo mediante una solicitud de compra, la cual es registrada por el ejecutivo con el fin de concretar la venta o en su defecto llevar un seguimiento de la misma, realizando las actividades necesarias para lograr la satisfacción del cliente.

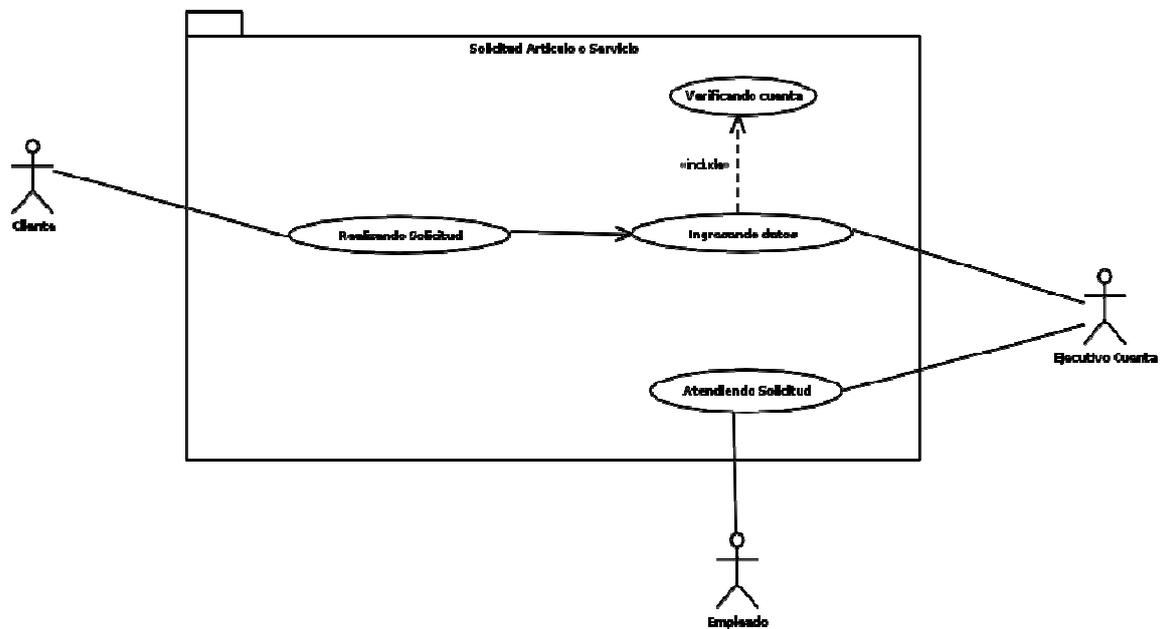


Imagen 4.3 Diagrama de Pedido

Cuando un cliente realiza un pedido se verifica que ya se tengan sus datos como empresa y se tenga al menos un contacto, posteriormente se procede a levantar la solicitud de nueva venta, la cual es dada de alta por el ejecutivo, asignando prioridad y tiempos de respuesta los cuales pueden ser modificados a consideración de un supervisor; para después comenzar con las actividades necesarias para poder culminar con el proceso de venta.

#### **Descripción caso de uso nueva orden**

Caso de uso: **ORD1** Nueva Orden.

Actores: Cliente, Ejecutivo de cuenta, Supervisor, Empleado, Sistema.

Tipo: Primario

Propósito: Crear una nueva orden de venta

Precondiciones: Usuario Autenticado (**AUT1**)

Tener dada de alta al menos una empresa (**EMP1**).

Tener dado de alta al menos un contacto para la empresa (**CLI1**).

Flujo Principal:

- 1 Este caso de uso inicia cuando el cliente solicita un producto.
- 2 El Ejecutivo de cuenta establece los tiempos de resolución, pudiendo ser el mismo día cuando inicia y termina la operación de compra/venta, establece las prioridades, asigna un contacto de referencia y describe la venta(**E1**).
- 3 El sistema asigna la venta al consultor que está dando de alta la venta y guarda la orden.
- 4 Se notifica al consultor que tiene asignada una nueva venta para que realice las actividades necesarias (**ORD2**) para poder terminar con la compra/venta en los tiempos establecidos.
- 5 Se procede a listar las órdenes de venta dadas de alta (**A1**).

Subflujos:

**A1** Listar Ordenes. El sistema asociara las órdenes generadas por el cliente para mostrarlas en pantalla.

Excepciones:

**E1:** El sistema verifica que los datos obligatorios hayan sido ingresados, en caso contrario notifica al usuario para que los ingrese.

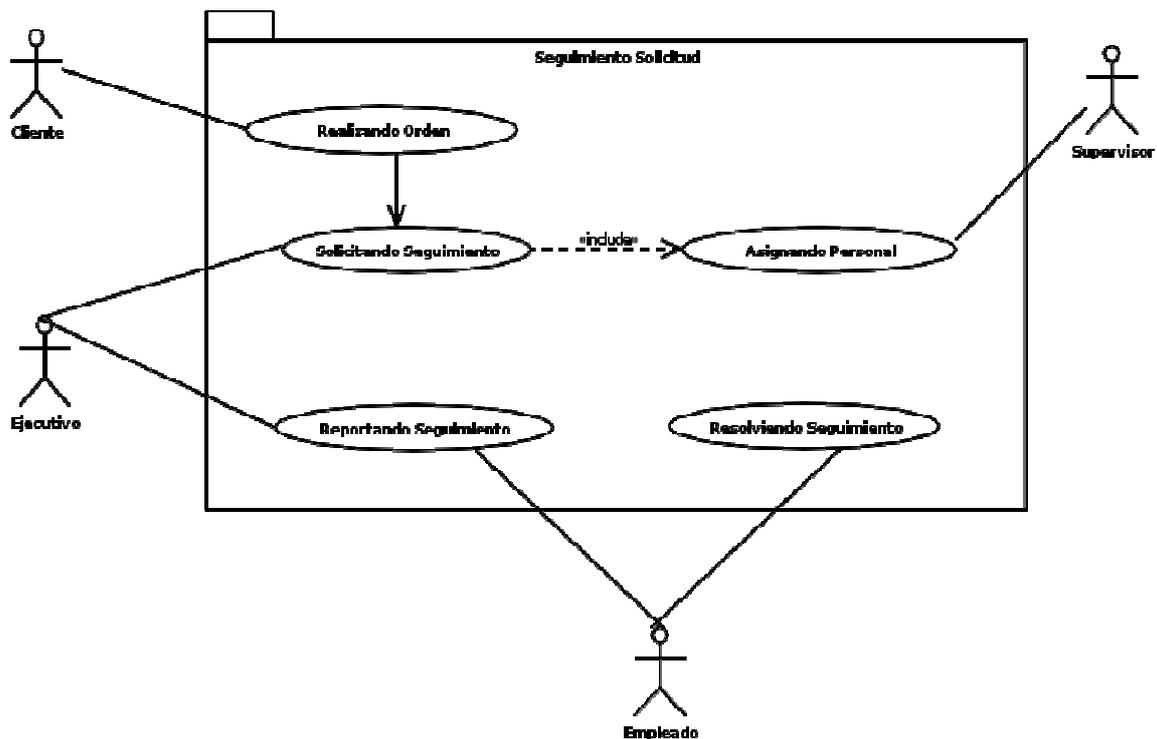


Imagen 4.4 Diagrama de seguimientos

Las solicitudes generadas por el cliente son ingresadas y atendidas por el ejecutivo de cuenta, el cual determinara la acción y tiempo de respuesta a seguir para la oportuna solución, posteriormente se asignara la tarea al empleado correspondiente y una vez realizada la actividad el ejecutivo de cuenta es informado para que vuelva a tener contacto con el cliente e informarle la solución.

#### Descripción de caso de uso seguimiento compra

Caso de uso	<b>ORD2</b> Seguimiento de compra.
Actores	Cliente, Ejecutivo de cuenta, Supervisor, Empleado, Sistema.
Tipo	Primario
Propósito	Crear una nueva actividad para poder dar seguimiento a la compra de un cliente.
Precondiciones	Tener registrada una compra para el cliente ( <b>ORD1</b> ).

Flujo Principal:

- 1 Este caso de uso inicia cuando el cliente realiza una compra y es necesario asignar actividades secundarias para poder dar seguimiento a la solicitud del cliente.

- 2 El Ejecutivo de cuenta identifica una de las siguientes actividades a realizar (Tarea, Incidencia, Cita o Llamada) estableciendo tiempos, prioridades y asignando al empleado que dará solución a la solicitud del cliente.
- 3 El sistema mostrara un listado con las ordenes realizadas **(A1)** por el cliente para poder relacionar la petición actual.
- 4 El sistema guardara la información de la nueva orden, notificara al empleado asignado por medio de una alerta **(E1)**.

Subflujos:

**A1** Listar Ordenes. El sistema asociara las órdenes generadas por el cliente para mostrarlas en pantalla.

Excepciones:

**E1:** El sistema verifica que los datos obligatorios hayan sido ingresados, en caso contrario notifica al usuario para que los ingrese.

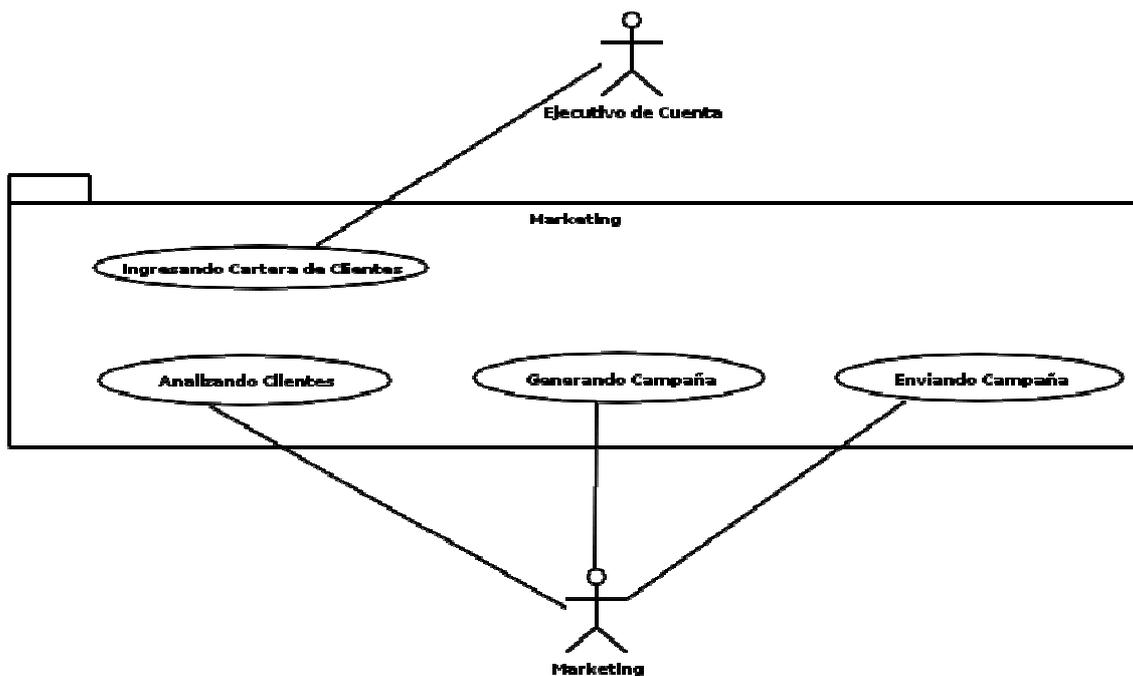


Imagen 4.5 Diagrama de campañas publicitarias.

Para el envío de campañas publicitarias, marketing genera una lista de contactos clasificándolos por su tipo (clientes o prospectos) a los cuales se les envían boletines informativos y/o promociones.

**Descripción casos de uso alta campaña.**

Caso de uso	<b>MAR1</b> Alta Campaña.
Actores	Analista, Marketing, Sistema
Tipo	Primario
Propósito	Alta de una nueva Campaña
Precondiciones	Cartera de Clientes <b>(CLI1)</b> .
Iniciado por:	Usuario
Terminado por:	Usuario

Flujo Principal:

- 1 Este caso inicia cuando un Usuario desea iniciar una campaña.
- 2 El Usuario captura los datos de la nueva campaña y selecciona a qué tipo de cliente será enviada **(E1)**.
- 3 Al Usuario se le mostrara una lista del Tipo de Cliente seleccionado **(CLI2)**, e indicará a quienes será enviado el correo.
- 4 Se guarda la Campaña

Subflujos: Ninguno

Excepciones:

**E1:** No es posible generar la nueva campaña. No hay clientes dados de alta

**Descripción caso de uso listado de campañas.**

Caso de uso	<b>MAR2</b> Listado de campañas.
Actores	Analista, Marketing, Sistema
Tipo	Secundario
Propósito	Listar la información de las campañas dadas de alta en el sistema.
Precondiciones	Tener una sesión activa.
Iniciado por:	Usuario
Terminado por:	Usuario

Flujo Principal:

- 1 Este caso de uso inicia cuando el Usuario solicita al sistema mostrar el listado de las campañas.
- 2 El sistema relacionara con la empresa las campañas que se encuentra dentro del sistema.
- 3 El sistema muestra un listado de las campañas por página permitiendo ver los campos que especifican la campaña.

Subflujos: Ninguno

Excepciones: Ninguno

### Descripción caso de uso envío campaña.

Caso de uso **MAR3** Envío de Campañas

Actores Analista, Marketing, Sistema

Tipo Secundario

Propósito Envío de email Marketing

Precondiciones Haber dado de Alta un Campaña **(MAR1)**.

Iniciado por: Analista

Terminado por: Analista

Flujo Principal:

- 1 Este caso de uso inicia cuando el analista desea enviar vía email publicidad a los clientes.
- 2 El sistema mostrara las campañas por empresa que se encuentra dentro del sistema **(MAR2)**.
- 3 El analista dará clic sobre el hipervínculo para enviar la campaña **(E1)**.

Subflujos: Ninguno

Excepciones:

**E1:** Fallo al enviar email. El servidor de correo SMTP no pudo ser alcanzado.

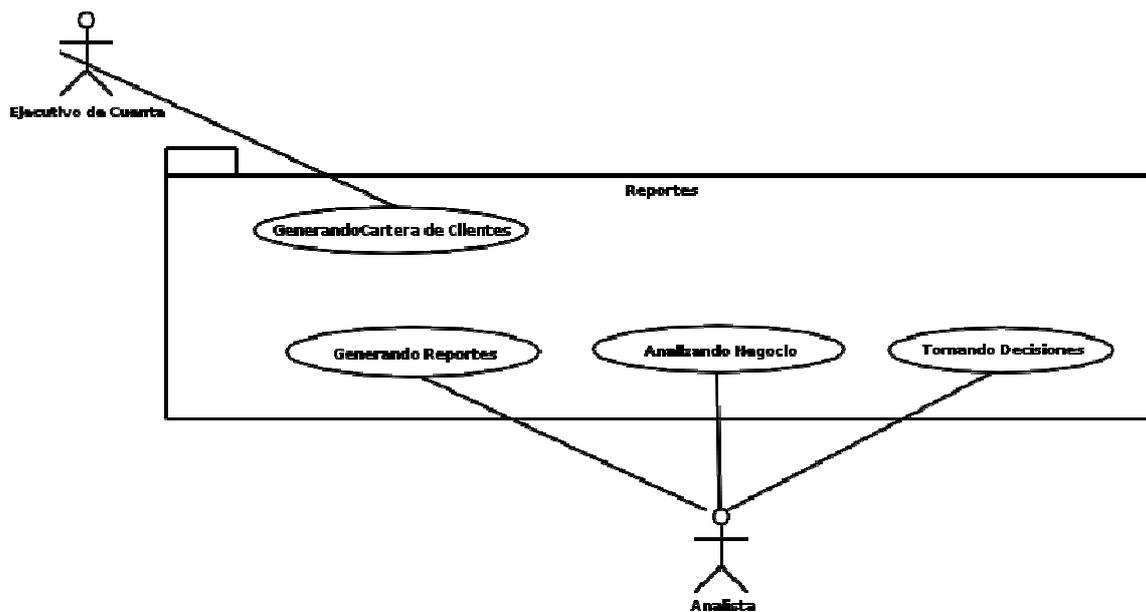


Imagen 4.6 Diagrama de Reportes y Toma de Decisiones

Para la toma de decisiones el gerente se auxiliara en la generación de reportes proporcionados por el sistema, los cuales serán alimentados por la operación diaria.

#### **Descripción caso de uso reportes.**

Caso de uso	<b>REP1</b> Reportes
Actores	Ejecutivo de Cuenta, Analista, Marketing, Sistema
Tipo	Primario
Propósito	Generación de Reportes
Precondiciones	Cartera de Clientes <b>(CLI2)</b> .
Iniciado por:	Analista
Terminado por:	Analista

Flujo Principal:

- 1.- Este caso inicia cuando el usuario necesita generar reportes.
- 2.- Si el usuario así lo necesita podrá obtener la información detallada por cliente **(A1)**, se podrá consultar el detalle por ejecutivo **(A2)**, existiendo la posibilidad de obtener un total de ventas y actividades de cada ejecutivo **(A3)**.
- 3.- Una vez que se cuente con la información Impresa, el analista podrá obtener los resultados deseados para una correcta toma de decisión.

Subflujos:

**A1** Reporte Ventas por Cliente se presentara una grafica en la cual se puede visualizar el número de compras de cada cliente, realizadas durante el día.

**A2** Reporte Ventas por Ejecutivo presentara gráficamente el numero de ventas de cada Ejecutivo, realizadas durante el día.

**A3** Reporte que mostrara de manera grafica y en colores diferentes el numero de ventas y las atenciones realizadas diariamente por Ejecutivo.

Excepciones: Ninguna

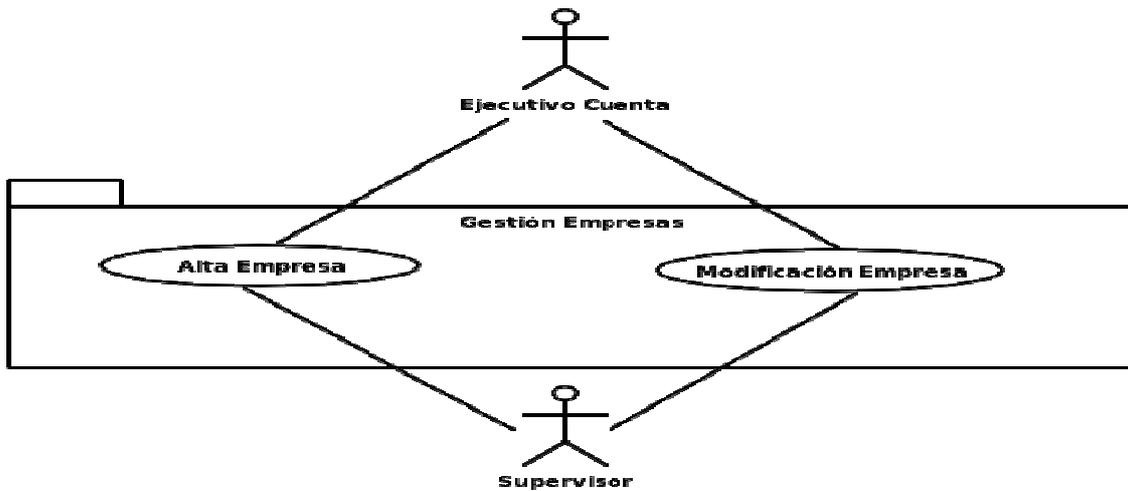


Imagen 4.7 Diagrama Gestión empresas.

El ejecutivo genera un catalogo de empresas a las que se les brindara el servicio. Ingresando datos generales de la misma con la opción de poder modificarlos.

#### Descripción caso de uso alta de empresa.

Caso de uso **EMP1** Alta de empresas

Actores Ejecutivo de cuenta, Supervisor, Administrador, Sistema.

Tipo Primario

Propósito

Dar de alta una empresa dentro del catalogo de las empresas a las cuales se les proporciona un bien y/o servicio.

Precondiciones Tener una sesión activa.

Flujo Principal:

- 1 Este caso de uso inicia cuando el usuario decide dar de alta una empresa.
- 2 El sistema permite capturar el nombre de la empresa, giro, domicilio de facturación, domicilio de envío y si la empresa a dar de alta es un corporativo o un cliente individual **(E1)**.
- 3 El sistema continuo (E2) en el caso de uso **(EMP3)**.

Subflujos: Ninguno

Excepciones:

**E1:** Capturar datos obligatorios.

**E2:** El sistema no pudo guardar la información de la empresa por lo que no continuara con el caso de uso.

### **Descripción caso de uso modificación de empresa.**

Caso de uso **EMP2** Modificación de empresas

Actores Ejecutivo de cuenta, Supervisor, Administrador, Sistema

Tipo Primario

Propósito

Modificar los datos de una empresa para mantener la información lo más actualizada posible.

Precondiciones Tener dadas de alta las empresas **(EMP1)**.

Flujo Principal

- 1 Este caso de uso inicia cuando el usuario solicita al sistema mostrar el listado de empresas y selecciona alguna para ser modificada.
- 2 El sistema permite capturar el nombre de la empresa, giro, domicilio de facturación, domicilio de envío y si la empresa es un corporativo o un cliente individual **(E1)**.
- 3 El sistema continuo **(E2)** en el caso de uso **(CUE3)**.

Subflujos: Ninguno

Excepciones:

**E1:** Capturar datos obligatorios.

**E2:** El sistema no pudo guardar la información de la empresa por lo que no continuara con el caso de uso.

### **Descripción caso de uso listado de empresas.**

Caso de uso **EMP3** Listado de empresas

Actores Ejecutivo de cuenta, Supervisor, Administrador, Sistema

Tipo Primario

Propósito Listar la información de las empresas dadas de alta en el sistema.

Precondiciones Tener dadas de alta las empresas **(EMP1)**.

Flujo Principal:

- 1 Este caso de uso inicia cuando el usuario solicita al sistema mostrar el listado de empresas.
- 2 El sistema verifica las empresas que pertenecen o están relacionadas con el usuario que se encuentra dentro del sistema.
- 3 El sistema muestra un listado de 10 empresas por página permitiendo ver la información de interés general con la posibilidad de ver el detalle **(A1)** o modificar la información de la cuenta **(EMP2)**.

Subflujos: **A1** Ver detalle. El sistema recaba la información completa de la empresa seleccionada y muestra dicha información.

Excepciones

**E1:** Capturar datos obligatorios.

**E2:** El sistema no pudo guardar la información de la empresa por lo que no continuara con el caso de uso.

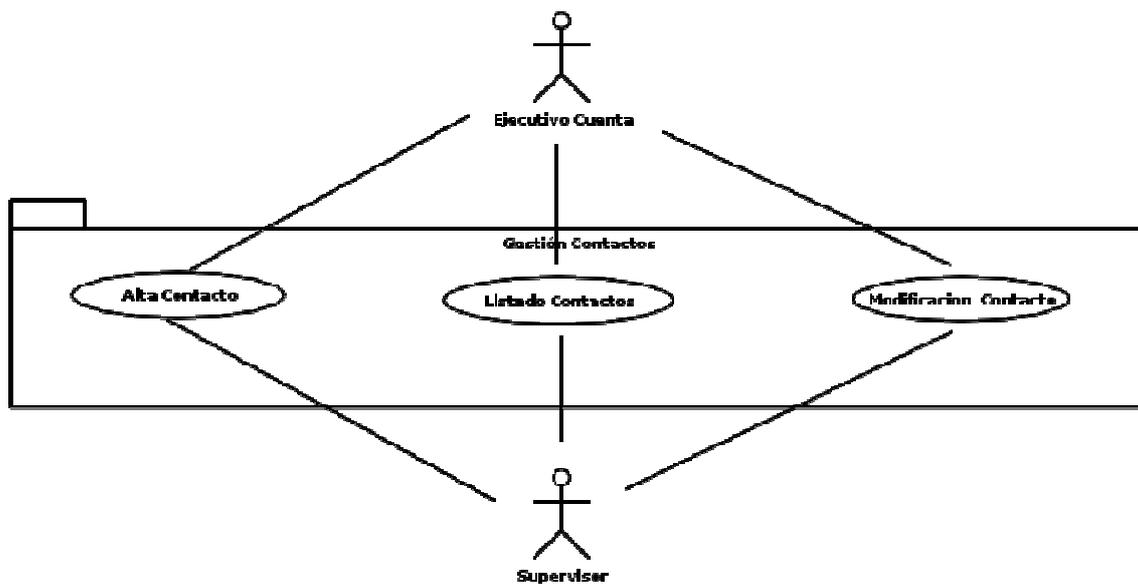


Imagen 4.8 Gestión Contactos

Complementando la gestión de empresas se darán de alta los contactos que la conforman, permitiendo establecer las relaciones comerciales.

#### Descripción caso de uso nuevo contacto.

Caso de uso **CLI1** Nuevo Contacto.

Actores Ejecutivo de Cuenta.

Tipo Primario

Propósito

Dar de alta nuevos contactos para una Empresa, con quienes mantendremos una relación.

Precondiciones

Haber registrado con anterioridad la Empresa a la cual pertenece el nuevo contacto **(EMP1)**.

Iniciado por: Ejecutivo de Cuenta

Terminado por: Ejecutivo de Cuenta

Flujo Principal

- 1 Este caso de uso inicia cuando un Ejecutivo, tiene la necesidad de dar de alta un nuevo Contacto para una Empresa **(E1)**.
- 2 El Ejecutivo de Cuenta seleccionara de una lista, la Empresa a la cual pertenece el nuevo Contacto **(CLI2)**.
- 3 El Ejecutivo de Cuenta ingresa los datos del nuevo Contacto **(E2)**.

Subflujos: Ninguno

Excepciones:

**E1:** No se cuenta con ninguna Empresa registrada, el sistema pedirá que se de alta al menos una empresa para poder dar de alta un Contacto.

**E2:** El sistema verifica que los datos obligatorios hayan sido ingresados, en caso contrario notifica al usuario para que los ingrese.

#### **Descripción caso de uso listar contactos.**

Caso de uso **CLI2** Listar Contactos.

Actores Ejecutivo de Cuenta, Administrador, Supervisor, Sistema

Tipo Secundario

Propósito Listar los contactos dados de alta.

Precondiciones Haber agregado contactos **(CLI1)**.

Iniciado por: Ejecutivo de Cuenta

Terminado por: Ejecutivo de Cuenta

Flujo Principal

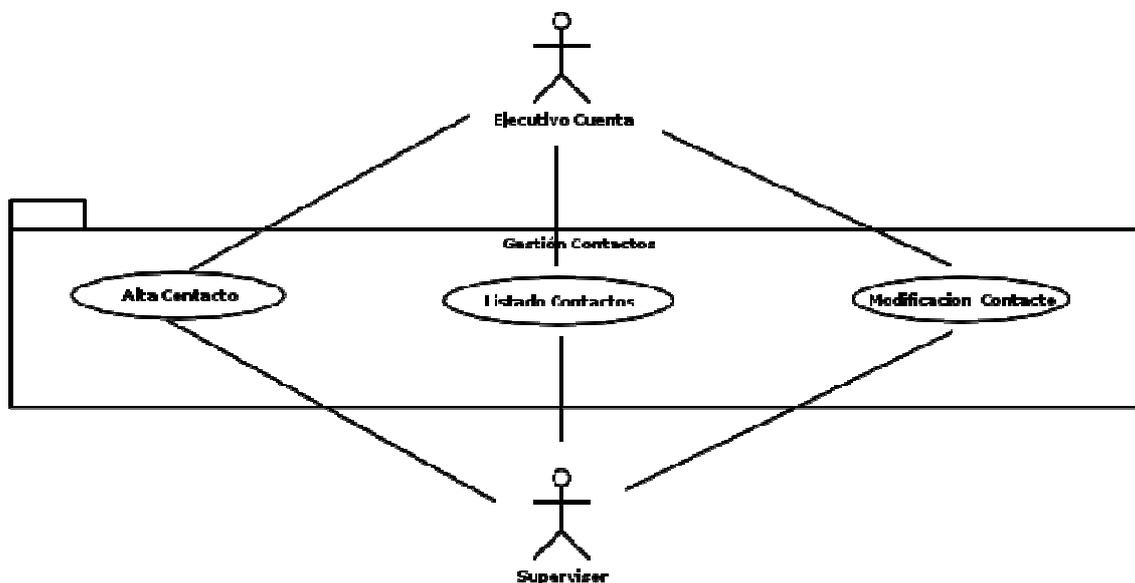
- 1 Este caso de uso inicia cuando el usuario solicita al sistema mostrar la lista de contactos.
- 2 El sistema verifica las empresas relacionadas con el usuario que se encuentra dentro del sistema, y liga los contactos a las mismas **(E1)**.
- 3 El sistema muestra un listado de 10 contactos por página permitiendo ver la información de interés general con la posibilidad de ver el detalle y modificar la información **(A1)**.

Subflujos

**A1** Modificar Usuario. Permite modificar la información del contacto, como son su nombre, teléfono, y domicilio.

Excepciones

**E1:** No se cuenta con ninguna Empresa registrada, el sistema pedirá que se ingrese al menos una empresa.



**Diagrama 4.9** Gestión de Empleados.

**Descripción caso de uso nuevo empleado.**

Caso de uso **EMP1** Nuevo Empleado

Actores Administrador, Supervisor, Sistema

Tipo Primario

Propósito

Dar de alta en el sistema nuevos usuarios con el rol Consultor o Supervisor para poder acceder al sistema y dar seguimiento a las solicitudes de los clientes.

Precondiciones Acceso al sistema con el rol Administrador.

Iniciado por: Administrador.

Terminado por: Sistema.

Flujo Principal

- 1 El Administrador selecciona la opción Nuevo Empleado del Menú Consultor.
- 2 El sistema solicita los datos del nuevo empleado **(E-1)**.
- 3 El sistema guarda los datos del nuevo usuario y continúa en el caso de uso **(EMP2)**.

Subflujos Ninguno

Excepciones

**E-1:** No se capturan todos los datos. El sistema mostrará un mensaje indicando que datos hacen falta capturar.

#### **Descripción caso de uso listado empleados.**

Caso de uso                    **EMP2** Listado de empleados

Actores                        Administrador, Supervisor, Sistema

Tipo                            Secundario

Propósito

Mostrar un listado de todos los empleados que forman parte de la empresa y dan solución a las solicitudes de los clientes.

Precondiciones            Usuario Autenticado con el perfil Apropiado.  
Tener una sesión vigente.

Iniciado por:                Administrador o Supervisor.

Terminado por:            Administrador o Supervisor

Flujo Principal:

- 1 El usuario solicita al sistema mostrar los empleados que se encuentran dados de alta sin importar el status en el que se encuentren.
- 2 Se validan todos los usuarios que corresponden a la empresa del usuario.
- 3 El sistema mostrara 10 empleados por página y los datos de interés general **(A1)** de cada uno de los empleados con la opción de editar **(A2)** o ver el detalle.

Subflujos:

**A1** El usuario cambia el status del empleado. De acuerdo al status actual del empleado se realiza la modificación, en caso de encontrarse como activo se inactiva; si se encuentra inactivo se activa.

**A2** El usuario selecciona modificar el consultor. Se dirige al caso de uso EMP3

Excepciones:                Ninguna

#### **Descripción caso de uso modificar empleado.**

Caso de uso                    **EMP3** Modificar Empleado

Actores                        Administrador, Supervisor, Sistema

Tipo                            Primario

Propósito:

Modificar en el sistema los usuarios dados de alta para poder acceder al sistema y dar seguimiento a las solicitudes de los clientes.

Precondiciones Acceso al sistema con el rol Administrador o Supervisor.  
Haber seleccionado el usuario a modificar desde el listado (**EMP2**).

Iniciado por: Administrador.

Terminado por: Sistema.

Flujo Principal

- 1 El Administrador selecciona la opción Nuevo Empleado del Menú Consultor.
- 2 El sistema solicita los datos del nuevo empleado (**E1**).
- 3 El sistema guarda los datos del nuevo usuario y continúa en el caso de uso **EMP2**.

Subflujos Ninguno

Excepciones

**E-1:** No se capturan todos los datos. El sistema mostrará un mensaje indicando que datos hacen falta capturar.

#### **Descripción caso de uso mostrar menú.**

Caso de Uso **AUT2** Mostrar Menú.

Actores Usuario, Sistema.

Tipo Primario.

Propósito

Mostrar un menú de opciones que permita navegar por el sistema de acuerdo al perfil del usuario autenticado.

Precondiciones Acceso de usuario validado (**AUT1**).

Iniciado por: Sistema.

Terminado por: Sistema.

Flujo Principal

- 1 El sistema generará dependiendo el perfil un menú de acciones disponibles.
- 2 El usuario elige alguna de las opciones que le son presentadas (**E1**):

Orden

- Nueva Orden: **ORD1**
- Listar Ordenes: **ORD2**

Cliente

- Nuevo Contacto: **CLI1**
- Listar Contactos: **CLI2**

#### Empresa

- Nueva Cuenta: **EMP1**
- Modificar Cuenta **EMP2**
- Listar Cuentas: **EMP3**

#### Consultor

- Nuevo Empleado: **CON1**
- Listar Empleados: **CON2**

#### Marketing

- Alta Campaña: **MAR1**
- Listado Campaña: **MAR2**
- Envío Campaña: **MAR3**

#### Reportes

- Reportes: **REP1**

Subflujos: Ninguno

Excepciones:

**E1:** Inactividad de sesión, el usuario es re direccionado al caso de uso **AUT1**.

#### **Descripción caso de uso ingresar al sistema.**

Caso de uso	<b>AUT1</b> Ingresar al sistema.
Actores	Usuario del sistema, Sistema.
Tipo	Primario
Propósito	Validar el acceso al sistema.
Precondiciones	El usuario debió haber sido dado de alta.
Iniciado por:	Usuario del Sistema
Terminado por:	Usuario del Sistema

#### Flujo Principal

1. El "Usuario" trata de acceder al sistema **(E1) (E3)**.
2. El sistema muestra al usuario la pagina inicial de Autenticación y le son solicitados su usuario, contraseña y empresa para poder acceder al sistema.

3. El sistema valida contra la base de datos los parámetros que ingreso el usuario para poder dar acceso **(E2)** y mostrar el menú de actividades **(AUT2)**.

4. El sistema dirige al usuario a la pantalla inicial.

Subflujos: Ninguno

Excepciones:

**E-1:** El usuario trata de realizar una acción sin logearse. El usuario es re direccionado a la página inicial de login.

**E-2:** Los parámetros de Login no son correctos. El sistema vuelve a mostrar al usuario la página de login.

**E-3:** Se inactiva la sesión por inactividad. El usuario es re direccionado a la página inicial de login.

#### **4.6 Diseño de la interfaz.**

Se generará una interfase 100% amigable con el usuario, buscando la simplicidad de las operaciones y la intuición del usuario. Se especifica en detalle cómo se verán las interfaces de usuario al ejecutar cada uno de los casos de uso.

#### Funcionamiento global del Sistema

Comenzaremos describiendo las diversas acciones que el usuario puede realizar dentro del sistema como son:

- Empresas: el usuario podrá dar de alta una nueva empresa, listarla y modificarla. Se podrá hacer la diferenciación entre personas físicas y morales.
- Contactos: el usuario podrá asignar contactos a una empresa, modificarlos y listarlos.
- Usuarios: el usuario podrá dar de alta un nuevo usuario, cambiar su status, listarlos y modificarlos. Los perfiles que se podrán dar de alta y modificar son: Ejecutivo de Cuenta y Supervisor.
- Órdenes: el usuario podrá dar de alta una nueva orden, indicar su tipo, asignar dependencias, dar seguimiento y solución a una orden. Además de que el sistema automáticamente en nueva asignación le mandara una alerta al responsable del seguimiento y solución de la misma.
- Reportes: el usuario podrá generar los siguientes reportes en formato PDF:

- Marketing: el usuario podrá dar de alta una nueva campaña, listarlas y enviarlas vía correo electrónico.

#### **4.7 Diagramas de Secuencia.**

Un diagrama de secuencia muestra los pasos de la Realización del Caso de Uso, con sus respectivos estereotipos (Frontera o Presentación, Control y Entidad).

Este Diagrama de Secuencia utiliza los métodos ya establecidos y que son mostrados en el Diagrama de Clases donde se entrelazan todas las clases *boundary*, *control* y *entity* que interactúan en todo el sistema, en el cual se ingresan los atributos y métodos de cada clase, por lo que en los diagramas de secuencia y colaboración únicamente se hace referencia a estos elementos, y se suponen que ya fueron entrelazados; en otras palabras por ejemplo en el diagrama de Secuencia se parte de una Clase hacia otra por medio de un método, que ejecuta cierta acción en una secuencia en el tiempo, y así sucesivamente hasta completar el escenario.

A continuación se muestra a detalle el diagrama de secuencia del Modulo de Autenticación, además del diseño de la vista y la implementación del MVC.

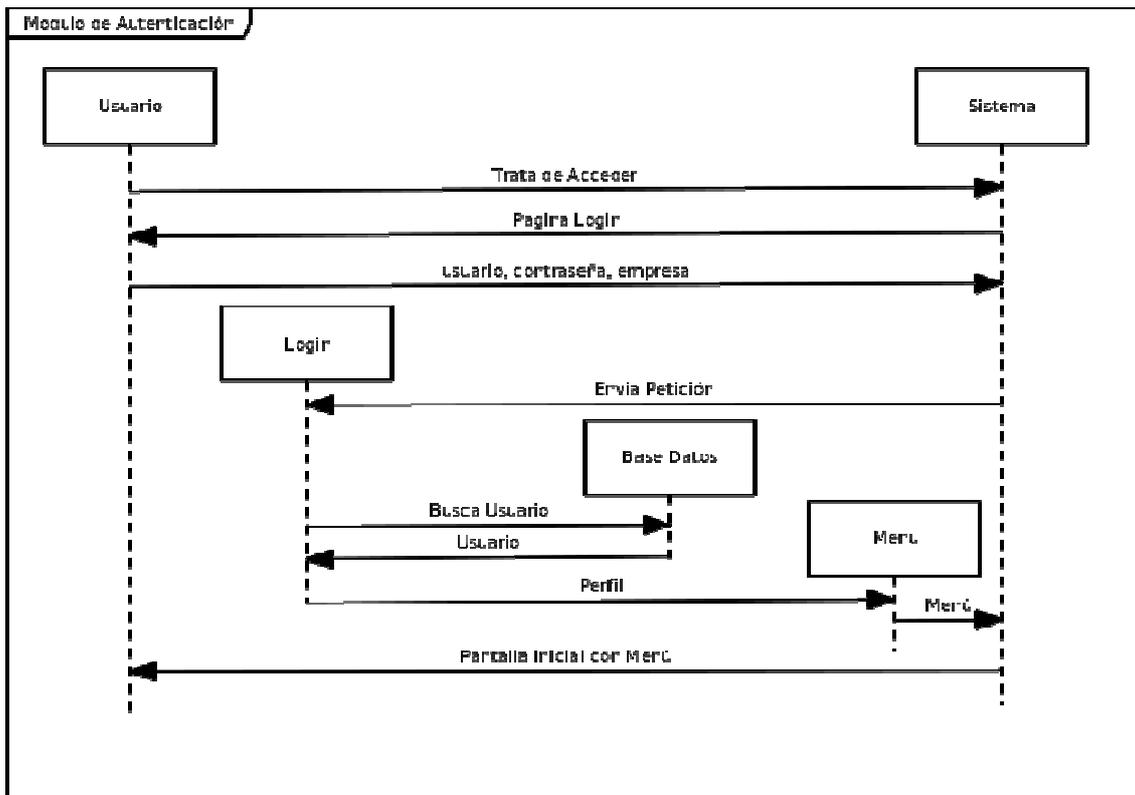


Imagen 4.10 Diagrama Secuencia Modulo de Autenticación

Imagen 4.11 Diseño Interfaz Página Login

En el Modulo Empresas, se manejan para la parte de la vista, varios formularios en HTML (vista); y el controller Gestión Empresas, además de que la persistencia a la base implementada Hibernate será mediante el bean Empresa.

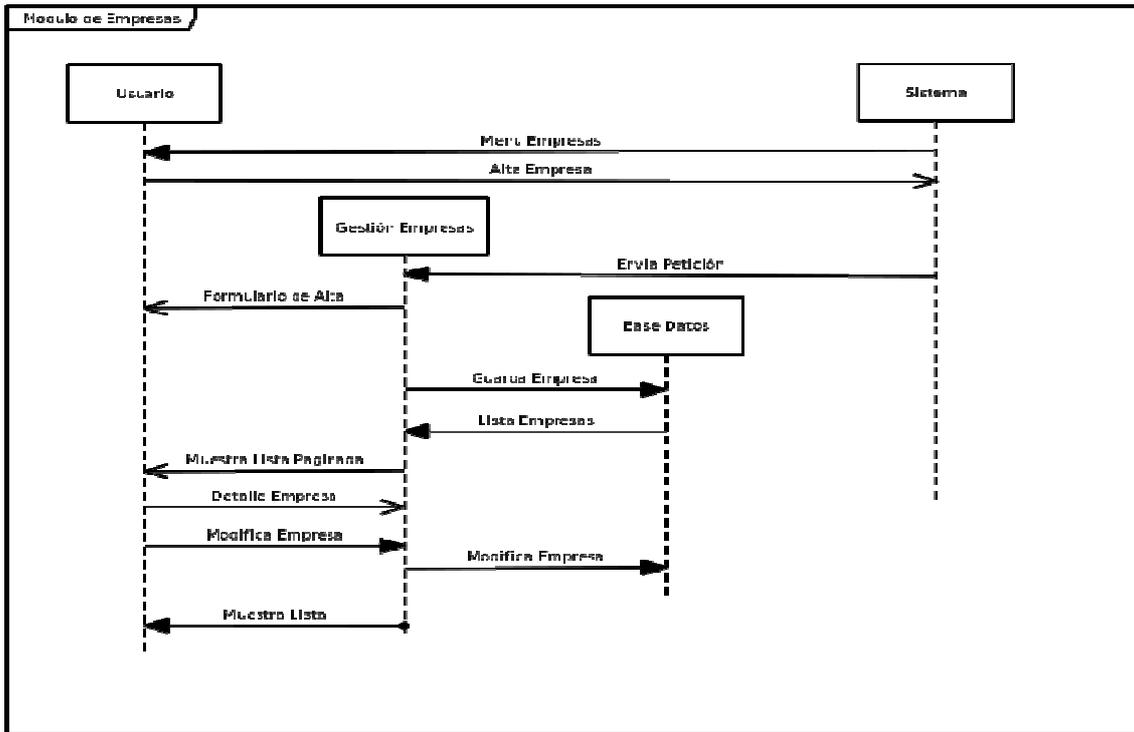


Imagen 4.12 Diagrama Secuencia Modulo de Empresas

Se muestra la pantalla de alta de empresa, se maneja un catalogo dinámico para la selección del giro, la base de datos podrá guardar dos direcciones; tanto de facturación como de envío, si es la misma se evitara el reescribir haciendo uso de la funcionalidad y el botón copiar.

**Nombre:**   
**Giro:**   
**Individuales?:**  SI  NO  
**Web:**   
**Domicilio Facturacion:**   
**Domicilio Envio:**

Imagen 4.13 Diseño Interfaz Alta Empresa

Para poder dar de alta un Contacto, anteriormente se debió de haber cargado y seleccionado una Empresa, en el siguiente flujo se presupone ya ha sido dada de alta y seleccionada con anterioridad una Empresa.

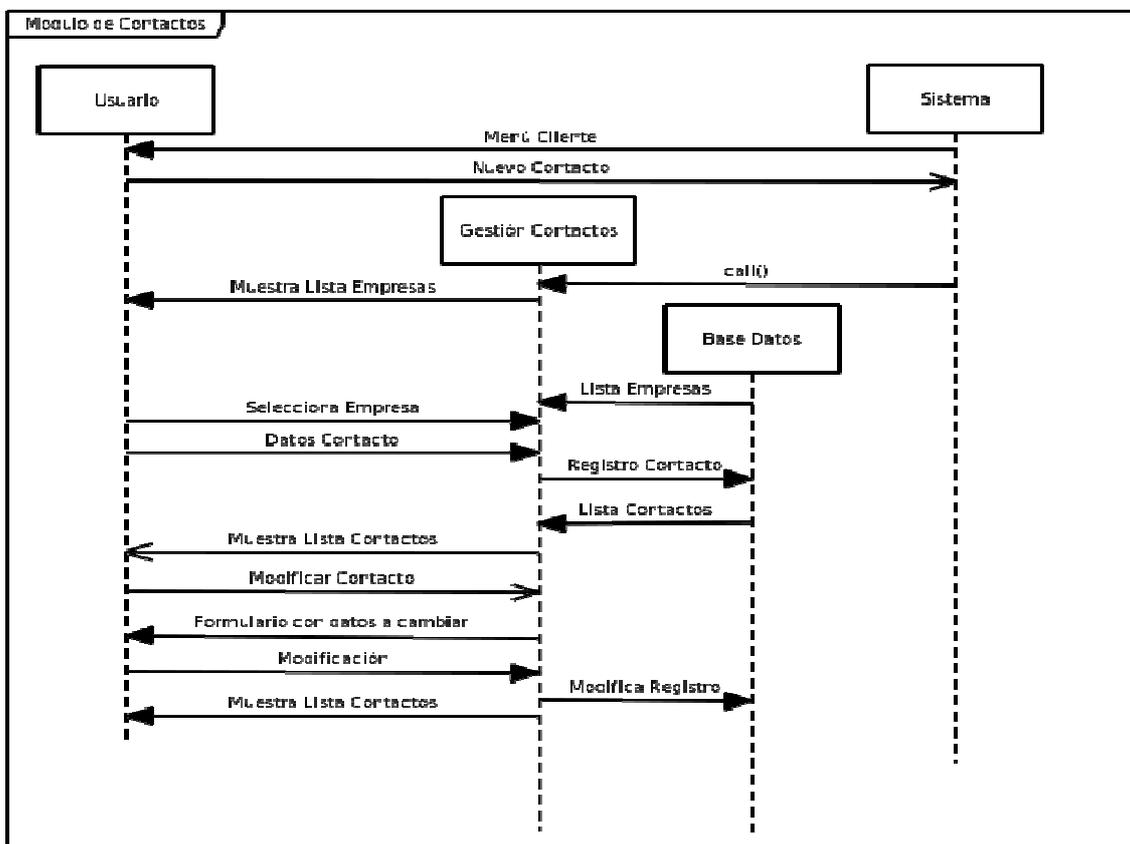


Imagen 4.13 Diagrama Secuencia Modulo de Contactos.

Cada que se de alta un nuevo contacto se podrá hacer la diferenciación de tipo de contacto que es. Esto con la finalidad de poder manejar los dos tipos de clientes fiscalmente. Personas físicas (profesionistas) y personas morales (empresas, asociaciones, etc.)

Nombre:

Domicilio:

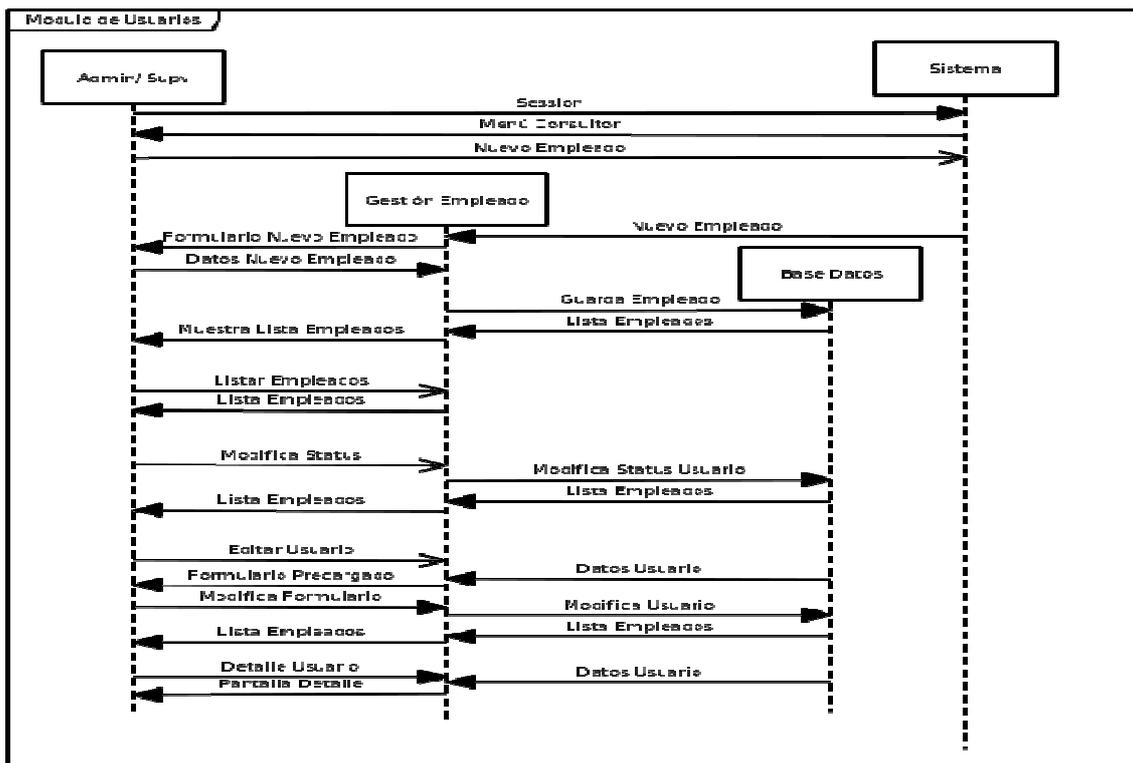
Email:

Tipo Cliente:

Telefono:

**Imagen 4.15 Diseño Interfaz Alta Contacto**

El siguiente flujo solo estará habilitado para el perfil Administrador. Esto debido a que como característica del sistema se pretende limitar las funciones de cada usuario mediante el manejo de un perfil para cada usuario.



**Imagen 4.16 Diagrama Secuencia Modulo de Usuarios.**

Al ingresar empleados al sistema se pretende manejar una interfaz amigable para la captura de información.

Nombre:

Rol:

Login:

Pwd:

Pwd:

[Agregar Telefono](#)

**Imagen 4.17 Diseño Interfaz Alta Usuario**

Es necesario que cada nuevo empleado, tenga al menos un teléfono de contacto. Se pretende que este sea el ID con el cual se pueda diferenciar cada empleado.

Telefono:

Extension:

Tipo:

**Imagen 4.18 Diseño Interfaz Alta Teléfono**

El siguiente es el flujo del escenario básico de la Gestión de Ordenes. Permite registrar al cliente como un prospecto y almacenar la información en la cartera de clientes.

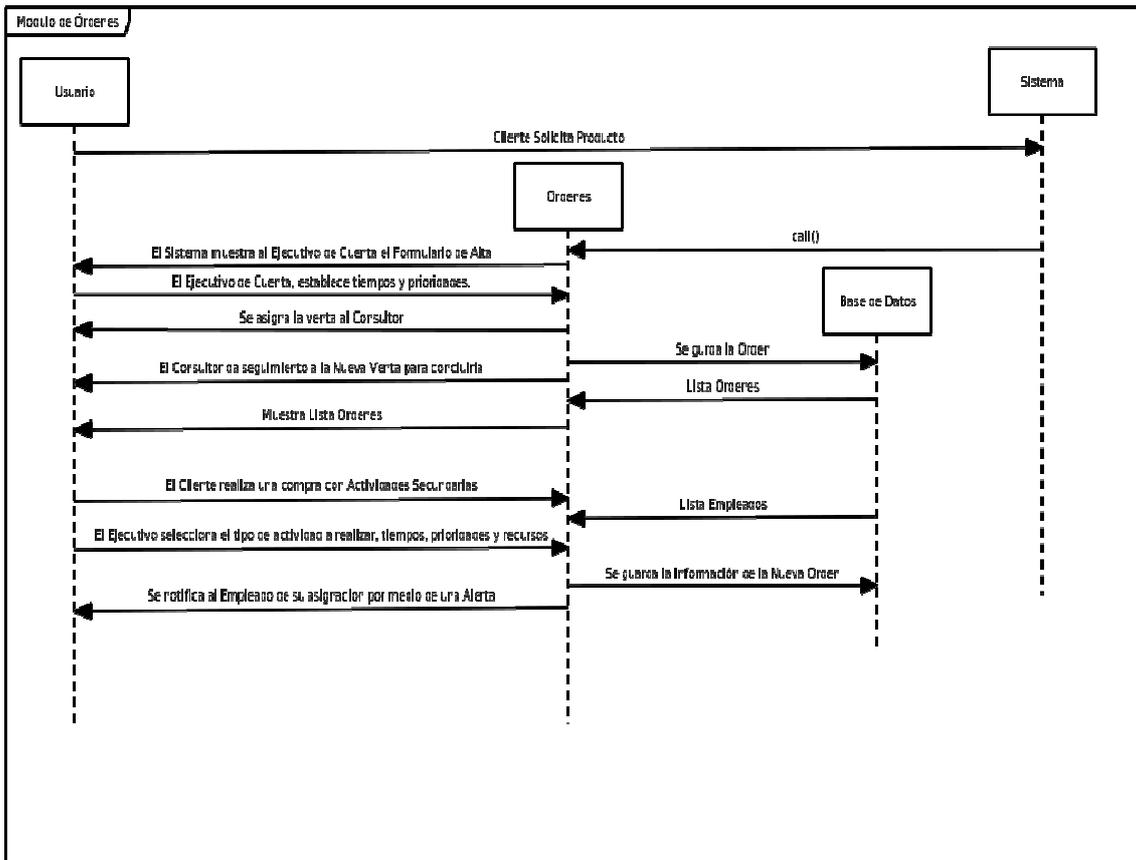


Imagen 4.19 Diagrama Secuencia Modulo de Órdenes

Se intenta lograr una solución general de un CRM, por ello la pantalla de alta de Orden, maneja conceptos generales principalmente definidos por el usuario.

Dar de alta una orden para

Asunto:  Estado:

Fecha Inicio:  Contacto:

Prioridad:  Tipo Actividad:

Asignado a:

Descripcion:

Imagen 4.20 Diseño Interfaz Alta Orden

Los reportes se generarán en PDF, para ello es necesario listar el menú de opciones y seleccionar el reporte que deseemos obtener.

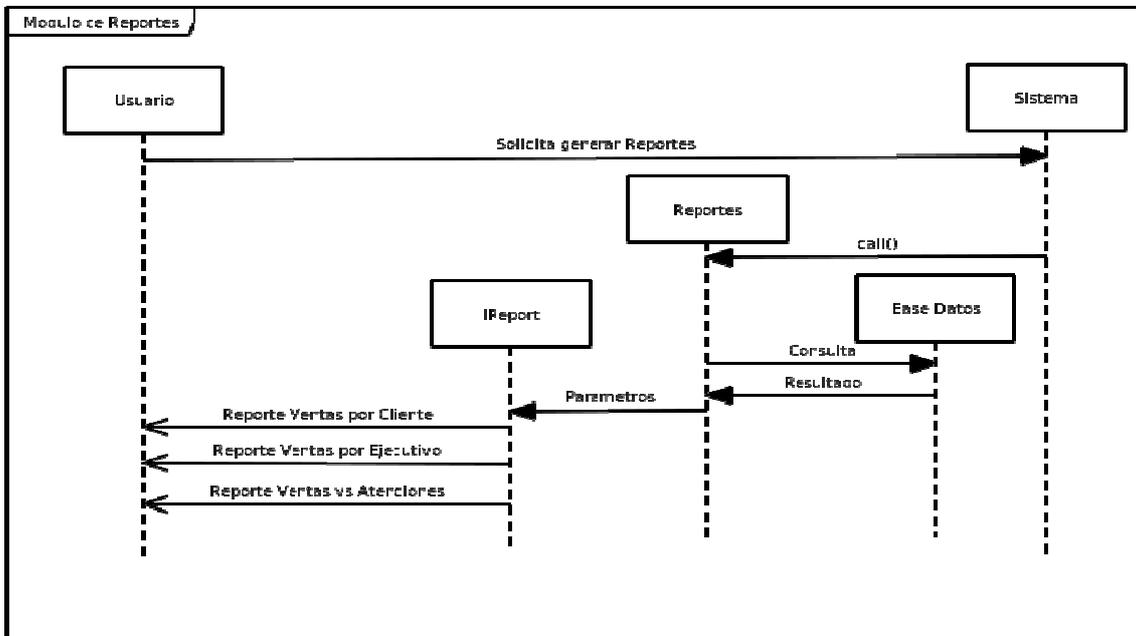


Imagen 4.21 Diagrama Secuencia Modulo Reportes

Para el envío de campañas, se usara la API de Java Mail, los formularios de alta de campañas, simplemente son para capturar el contenido del mensaje y la lista a quienes será enviado dicho correo.

Mediante el análisis de Marketing se pretende alcanzar la atención de un nuevo mercado, haciendo les llegar información relevante que coincida con los intereses registrados en la cartera de clientes.

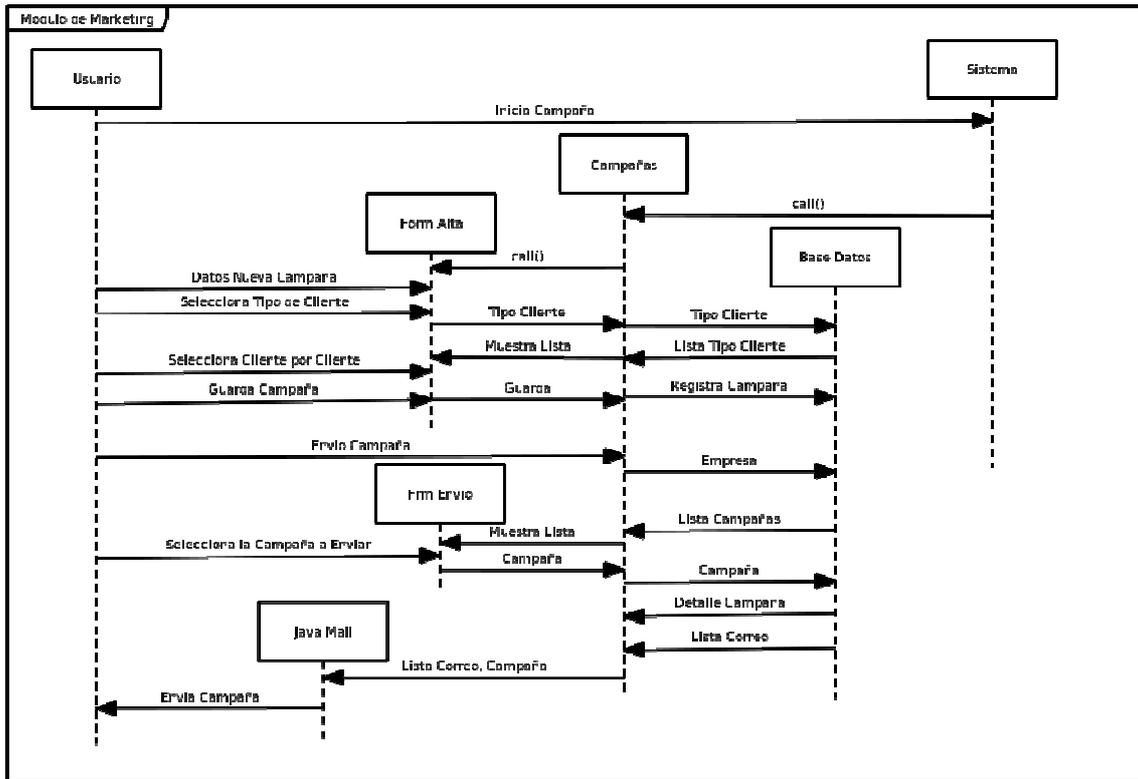


Imagen 4.22 Diagrama Secuencia Modulo Marketing

Cuando el mercado logó en cuestión segmente la cartera de clientes, se podrá generar un listado con los prospectos y asignarle un nombre a la campaña que se esta creando.

Seleccione el tipo de clientes:

Nombre de la campaña:

Descripcion:

Cientes

Imagen 4.23 Diseño Interfaz Alta Campaña

## CAPÍTULO V DESARROLLO Y PRUEBAS DEL SISTEMA.

### 5.1 Diseño de la realización del caso de uso.

Conociendo ya el análisis del sistema a implementar, pasaremos a diseñarlo en el entorno J2SE, por lo que se trabajará el diseño aplicándolo con componentes *Java*, pudiendo comprender el enlace que existe entre los componentes de diseño y los mecanismos del J2SE utilizados.

Implementación Proceso de Presentación de Petición:

- Todas las peticiones de usuario son manejadas por un solo Controlador de Presentación de Petición.
- El controlador principal examina cada petición para verificar consistencia y si el usuario necesita ser autenticado y/o autorizado.
- Si se autentica y autoriza al usuario, el Controlador de Presentación de Petición transmite a la petición un despachador apropiado.
- Un despachador maneja las interacciones del usuario para un Caso de Uso específico (hay un despachador para cada realización de Caso de Uso). Los despachadores "entienden" el flujo del Caso de Uso y coordinan interacciones entre el usuario y el Caso de Uso.
- Los despachadores no producen ninguna salida al usuario. En lugar de eso, llaman a vistas para crear código HTML o XML u otro código que regrese al dispositivo del usuario.
- Los despachadores no ejecutan la "lógica del negocio". En su lugar, utilizan a delegados del negocio como *proxies* a los servicios de servidor de aplicaciones.
- El controlador principal y los despachadores son implementados como *Servlets*. Por lo tanto, pueden ser colocados en diversos procesos y/o en diversas máquinas, que pueden ser deseables por razones de seguridad.
- Las vistas son implementadas como JSPs por facilidad de desarrollo.
- Los Delegados del Negocio son implementados como *Java Beans* que funcionan en el conducto de los delegados que llaman.

Las siguientes implementaciones de la arquitectura están contempladas en el mecanismo Acceso a Sesión EJB:

- Todos los componentes del servicio de negocio son implementados como *Session EJBs*.
- Un cliente (en la mayoría de los casos es el despachador) que requiere el acceso a un componente del servicio de negocio, crea una instancia de un delegado de la sesión EJB.

Como consecuencia de utilizar los dos mecanismos, la aplicación tiene una separación distinta en el estereotipo *boundary*, entre la presentación lógica y la lógica del negocio. Esa *boundary* es la delegada del negocio ese es el Proxy a los servicios de negocio de la aplicación. Esta separación es muy deseable por algunas razones:

- Proporciona un contrato explícito entre los diseñadores y los desarrolladores del nivel de presentación y el nivel del negocio del sistema.
- Permite que el nivel del negocio cambie independientemente del nivel de la presentación.
- Permite el desarrollo concurrente de los dos niveles.

El patrón utilizado aplica el Estándar J2EE el cual nos provee de todos los componentes *Java* necesarios para poder implementar el sistema y así poder interrelacionar dentro del mismo modelo los Casos de Uso establecidos con los componentes *Java* existentes y poder completar la Realización de los Casos de Uso en el Diseño del modelo del sistema.

El modelo creado va a servir para que los desarrolladores y cualquier persona que desee empezar a desarrollar el sistema, pueda entender de que se está hablando y por donde empezar, ya que se observan todas las interrelaciones que debe tener el sistema para funcionar como se especifican en los requerimientos.

Con base a los requerimientos planteados anteriormente se deberá implementar este sistema utilizando un servidor que funcionará como un contenedor *Web* (en este caso *Tomcat* por ser una implementación oficial de referencia para los *Java Servlet* y *Java Server Pager*) en el cual se tendrán los *Applets* y *Java Scripts*, *JSP's* a los cuales accederán los usuarios y serán vistos en un Navegador.

## 5.2 Diagrama de clases

Es de suma importancia mencionar que es distinto hablar del Modelo de Caso de Uso en si y de la Realización del mismo, y para diferenciar esto es preciso entender que un Modelo de Caso de Uso

en si muestra los pasos a seguir para llegar a un objetivo, con sus flujos alternos que ejemplifican escenarios o sea diversos caminos por donde se puede dirigir el Caso de Uso, esto es lo que entiende el cliente; lo mismo pasa en la Realización del Caso de Uso, únicamente que en este caso varía la forma de realizar los diagramas ya que se convierte o divide el Caso de Uso en varias clases de diferentes estereotipos, dependiendo de cuantos Entidad (*entity*), Presentación o Frontera (*boundary*) y Control (*control*) se identifiquen.

Una clase entidad suele reflejar datos o entidades del mundo real o sea elementos necesarios dentro de un sistema. Una clase frontera maneja comunicaciones entre el entorno del sistema y el sistema, suelen proporcionar la interfaz del sistema con un usuario o con otro sistema. Una clase control modela el comportamiento secuenciado específico de uno o varios casos de uso. Se trata de clases que coordinan los eventos necesarios para llevar a cabo el comportamiento que se especifica en el caso de uso, representan su dinámica.

En la figura 5.1 se muestran las clases participantes, así como las diferentes relaciones que existen entre ellas. Estas clases ya están divididas dependiendo de sus funciones, o sea que pueden ser Frontera o Presentación, Control y Entidad según las necesidades encontradas.

Este diagrama responde a los requerimientos planteados, y dado que son clases utilizan sus atributos y métodos para la intercomunicación entre ellas, esta intercomunicación está establecida por medio de las líneas que unen las clases, y su funcionalidad se podrá apreciar en los diagramas de secuencia y colaboración, ya que estos demuestran los caminos a seguir para representar los distintos escenarios.

Pasaremos de cómo lo entiende el cliente a como lo entiende el desarrollador del sistema, ya que se entra mas directamente en la implementación del sistema pero de una manera genérica.

A continuación se describe de manera gráfica la estructura del Sistema; mostrando sus clases, atributos y las relaciones entre ellos.

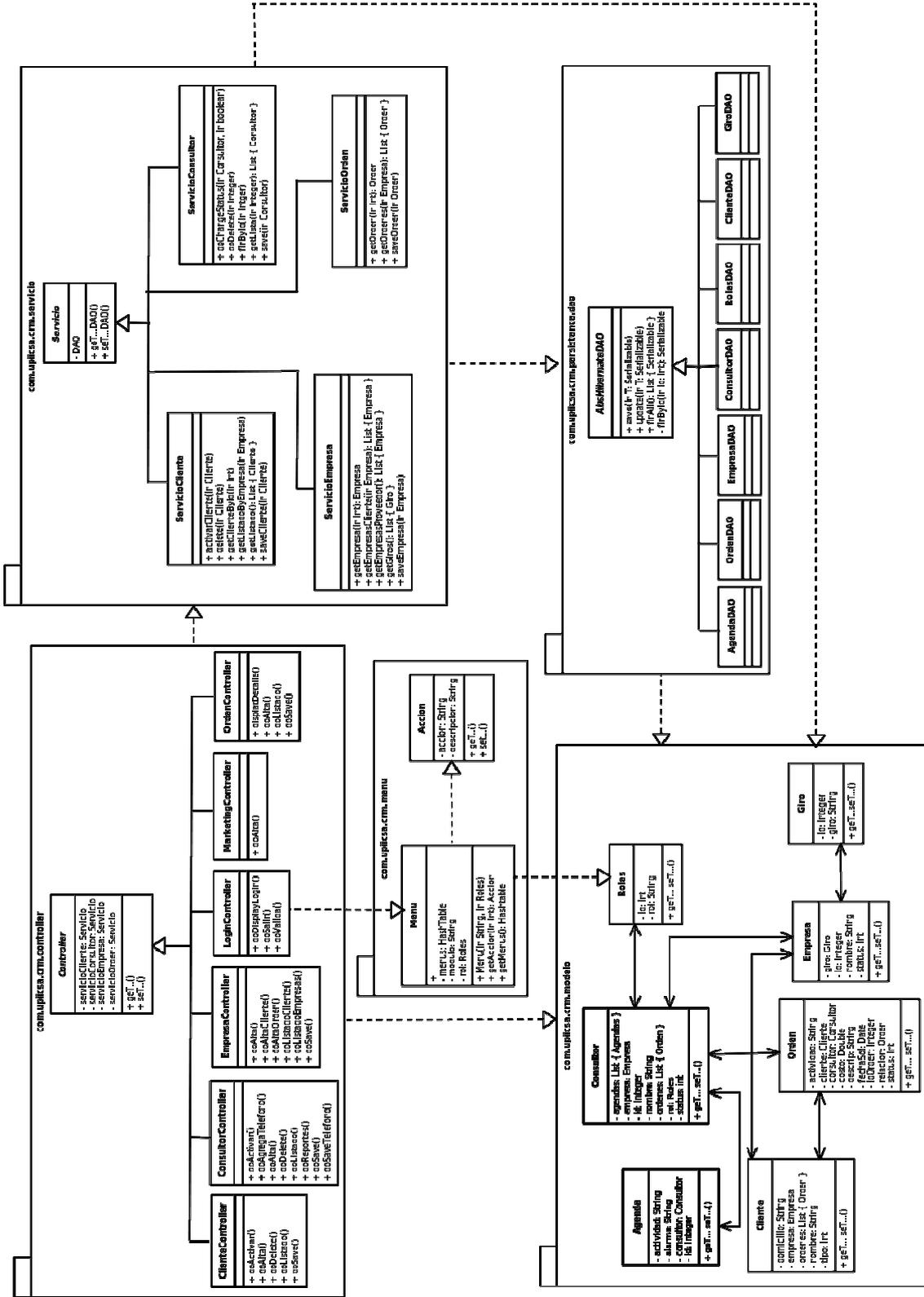


Imagen 5.1 Diagrama de Clases

### 5.3 Diccionario de clases.

A continuación se explica la funcionalidad de cada una de las clases vistas en el diagrama anterior:

<b>Nombre</b>	<b>Controller</b>
<b>Tipo</b>	Abstracta
<b>Descripción</b>	Clase Abstracta que extiende de MultiActionController, se usa para inyectar todo los servicios y al heredar de esta, no se tengan que volver a inyectar.
<b>Atributos</b>	<b>servicioCliente : ServicioCliente</b> Inyección de todos los métodos para el modulo Cliente. <b>servicioConsultor : ServicioConsultor</b> Inyección de todos los métodos para el modulo Consultor. <b>servicioEmpresa : ServicioEmpresa</b> Inyección de todos los métodos para el modulo Empresa. <b>servicioOrden : ServicioOrden</b> Inyección de todos los métodos para el modulo Ordenes.
<b>Operaciones</b>	Getters y Seters de cada atributo

### 5.2 Descripción clase Controller.

<b>Nombre</b>	<b>ClienteController</b>
<b>Tipo</b>	Controller
<b>Descripción</b>	Clase encargada de toda la lógica y re direccionamiento a la vista; para el modulo Clientes.
<b>Operaciones</b>	<b>doActivar(HttpServletRequest, HttpServletResponse)</b> Acción que cambia el status de un Cliente. <b>doAlta(HttpServletRequest, HttpServletResponse)</b> Acción que muestra la vista de Alta de Nuevo Cliente <b>doListado(HttpServletRequest, HttpServletResponse)</b> Acción que despliega la lista de Clientes en la vista <b>doSave(HttpServletRequest, HttpServletResponse)</b> Acción para dar de alta un nuevo cliente

### 5.3 Descripción clase ClienteController.

<b>Nombre</b>	<b>ConsultorController</b>
<b>Tipo</b>	Controller
<b>Descripción</b>	Clase encargada de toda la lógica y redireccionamiento a la vista; para el modulo Consultor.
<b>Operaciones</b>	<p><b>doActivar(HttpServletRequest, HttpServletResponse, HttpSession)</b> Acción para activar a un Empleado con Rol Ejecutivo de Cuenta y/o Supervisor</p> <p><b>doAgregaTelefono(HttpServletRequest, HttpServletResponse, HttpSession)</b> Acción para agrega un teléfono al consultor, en la capa de Vista.</p> <p><b>doAlta(HttpServletRequest, HttpServletResponse, HttpSession)</b> Acción para mostrar la vista de Alta de Consultor.</p> <p><b>doListado(HttpServletRequest, HttpServletResponse, HttpSession)</b> Acción para mostrar una lista de Empleados en la Vista.</p> <p><b>doReportes(HttpServletRequest, HttpServletResponse, HttpSession)</b> Acción para mostrar el menú de reportes.</p> <p><b>doSave(HttpServletRequest, HttpServletResponse, HttpSession)</b> Acción para guardar un nuevo Empleado.</p>

#### 5.4 Descripción clase ConsultorController.

<b>Nombre</b>	<b>EmpresaController</b>
<b>Tipo</b>	Controller
<b>Descripción</b>	Clase encargada de toda la lógica y re direccionamiento a la vista; para el modulo Empresa.
<b>Operaciones</b>	<p><b>doAlta(HttpServletRequest, HttpServletResponse)</b> Acción que direcciona a la Vista de Alta Empresa.</p> <p><b>doListadoEmpresas(HttpServletRequest, HttpServletResponse)</b> Acción que muestra el Listado de Clientes de una Empresa.</p> <p><b>doSave(HttpServletRequest, HttpServletResponse)</b> Acción para registrar una nueva Empresa</p>

#### 5.5 Descripción clase EmpresaController.

<b>Nombre</b>	<b>AbsHibernateDAO</b>
<b>Tipo</b>	Abstract
<b>Hereda</b>	<b>HibernateDaoSupport</b>
<b>Descripción</b>	Al implementar el Framework de Hibernate, se cuenta con un mapeo directo a las tablas en la base de datos, dicho mapeo lo entendemos en java en forma de Pojos, (clases serializables). Al manejar objetos serializables, conociendo su clase podemos saber a qué tabla hacemos referencia y persistirla; creando así un solo DAO abstracto con todas las operaciones a la base, del cual heredaran todos los demás DAO's.
<b>Operaciones</b>	<p><b>delete(T)</b> Método para borrar el registro en la base. Recibe como parámetro un objeto serializable.</p> <p><b>findAll()</b> Método que regresa una lista de Objetos.</p> <p><b>findById(ID)</b> Método que regresa un objeto de la base, según su ID.</p> <p><b>getClassName()</b> Método que nos indica la clase a la cual persistiremos.</p> <p><b>save(T)</b> Método para insertar nuevos registros.</p> <p><b>saveBatch(Collection&lt;T&gt;)</b> Método para insertar varios registros.</p> <p><b>setSessionFactoryHibernate(SessionFactory)</b> Método para obtener la conexión a la base.</p> <p><b>update(T)</b> Método para modificar un registro.</p>

### 5.6 Descripción clase AbsHibernateDAO.

### 5.4 Diagrama entidad-relación.

En seguida se muestra de manera grafica, las entidades o tablas creadas en la base de datos y sus atributos; además de las relaciones que mantienen dichas tablas; estas son necesarias para desarrollar toda la lógica de negocio.



## 5.5 Diccionario de datos.

Se empleara un sistema de Base de Datos como MySQL, para guardar la información necesaria como usuarios, empresas, empleados, ordenes etc. MySQL facilita todo ya que provee una interfase JDBC que hace fácil la integración con el servidor de Aplicaciones para utilizarlo en un manejo del contenedor de la persistencia de las Clases *Entity* o entidades, ya que todo dato perteneciente al CRM debe estar almacenado de manera persistente y no perderse.

Entidad	Atributos	Tipo	Largo	Restricciones	Descripción
<b>Agenda</b>	idconsultor	INT UNSIGNED	10	NOT NULL	Referencia al Consultor
	fecha	DATETIME		NOT NULL	Fecha de alta de la asignación
	actividad	VARCHAR	100	NOT NULL	Descripción de la actividad a realizar
	alarma	INT	5	NOT NULL	Importancia
<b>Campaña</b>	idcampania	INT UNSIGNED	10	NOT NULL	ID consecutivo de la campaña
	nombre	VARCHAR	100	NOT NULL	Nombre de la campaña
	idtiposclientes	INT	1	NOT NULL	Tipo de Cliente a los cuales será enviada
	idempresa	INT	10	NOT NULL	Empresa a la que pertenece
	descripción	VARCHAR	200	NULL	Descripción de la campaña

<b>Ciente</b>	idcliente	INT	10	NOT NULL	Identificador del Cliente
	nombre	VARCHAR	150	NOT NULL	Nombre del Cliente
	domicilio	VARCHAR	200	NOT NULL	Domicilio del Cliente
	idempresacliente	INT	10	NOT NULL	Empresa a la cual pertenece el cliente
	vigente	TINYINT	1	NOT NULL	Vigencia
	idtiposclientes	INT	1	NOT NULL	Tipo de Cliente
	idempresa	INT	10	NOT NULL	Empresa que dio de alta a este cliente( es la empresa logeada)
	email	VARCHAR	40	NOT NULL	Correo electrónico del Cliente

### 5.8 Descripción datos almacenados

Es importante mencionar que para esta parte de persistencia, es necesario ver a estos “datos” como “objetos”. Hemos hecho la implementación del Framework Hibernate para ello. Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación,

mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen.

Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por **SQL**. Hibernate genera las sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

A continuación mostraremos unos simples ejemplos de la implementación de Hibernate en nuestro sistema en código fuente.

## Cliente.hbm.xml

```
<?xml version="1.0" ?>
<!DOCTYPE hibernate-mapping (View Source for full doctype...)>
- <hibernate-mapping default-cascade="none" default-access="property" default-lazy="true" auto-
import="true">
- <class name="com.upiicsa.crm.modelo.Cliente" table="cliente" catalog="crm" mutable="true"
polymorphism="implicit" dynamic-update="false" dynamic-insert="false" select-before-
update="false" optimistic-lock="version">
- <id name="idcliente" type="java.lang.Integer">
  <column name="idcliente" />
  <generator class="identity" />
</id>
- <many-to-one name="tiposclientes" class="com.upiicsa.crm.modelo.Tiposclientes"
fetch="select" unique="false" update="true" insert="true" optimistic-lock="true" not-
found="exception" embed-xml="true">
  <column name="idtiposclientes" not-null="true" />
</many-to-one>
- <many-to-one name="empresacliente" class="com.upiicsa.crm.modelo.Empresacliente"
fetch="select" unique="false" update="true" insert="true" optimistic-lock="true" not-
found="exception" embed-xml="true">
  <column name="idempresacliente" not-null="true" />
</many-to-one>
- <many-to-one name="empresa" class="com.upiicsa.crm.modelo.Empresa" fetch="select"
unique="false" update="true" insert="true" optimistic-lock="true" not-found="exception" embed-
xml="true">
  <column name="idempresa" not-null="true" />
</many-to-one>
- <property name="nombre" type="string" unique="false" optimistic-lock="true" lazy="false"
generated="never">
  <column name="nombre" length="45" />
</property>
- <property name="domicilio" type="string" unique="false" optimistic-lock="true" lazy="false"
generated="never">
  <column name="domicilio" />
</property>
- <property name="vigente" type="boolean" unique="false" optimistic-lock="true" lazy="false"
generated="never">
  <column name="vigente" not-null="true" />
</property>
- <property name="email" type="string" unique="false" optimistic-lock="true" lazy="false"
generated="never">
  <column name="email" />
</property>
- <set name="clihtels" table="clihtel" inverse="true" lazy="true" fetch="select" sort="unsorted"
mutable="true" optimistic-lock="true" embed-xml="true">
- <key on-delete="noaction">
  <column name="idcliente" not-null="true" />
</key>
  <many-to-many class="com.upiicsa.crm.modelo.Clihtel" not-found="exception" embed-
xml="true" />
</set>
</class> </hibernate-mapping>
```

## Orden.hbm.xml

```
<?xml version="1.0" ?>
- <hibernate-mapping default-cascade="none" default-access="property" default-lazy="true" auto-
import="true">
- <class name="com.upiicsa.crm.modelo.Orden" table="orden" catalog="crm" mutable="true"
polymorphism="implicit" dynamic-update="false" dynamic-insert="false" select-before-
update="false" optimistic-lock="version">
- <id name="idorden" type="java.lang.Integer">
  <column name="idorden" />
  <generator class="identity" />
</id>
- <many-to-one name="consultor" class="com.upiicsa.crm.modelo.Consultor" fetch="select"
unique="false" update="true" insert="true" optimistic-lock="true" not-found="exception" embed-
xml="true">
  <column name="idconsultor" not-null="true" />
</many-to-one>
- <many-to-one name="tiposorden" class="com.upiicsa.crm.modelo.Tiposorden" fetch="select"
unique="false" update="true" insert="true" optimistic-lock="true" not-found="exception" embed-
xml="true">
  <column name="idtipoorden" not-null="true" />
</many-to-one>
- <many-to-one name="status" class="com.upiicsa.crm.modelo.Status" fetch="select"
unique="false" update="true" insert="true" optimistic-lock="true" not-found="exception" embed-
xml="true">
  <column name="idstatus" not-null="true" />
</many-to-one>
- <many-to-one name="cliente" class="com.upiicsa.crm.modelo.Cliente" fetch="select"
unique="false" update="true" insert="true" optimistic-lock="true" not-found="exception" embed-
xml="true">
  <column name="idcliente" not-null="true" />
</many-to-one>
- <property name="actividad" type="string" unique="false" optimistic-lock="true" lazy="false"
generated="never">
  <column name="actividad" length="45" not-null="true" />
</property>
- <property name="descripcion" type="string" unique="false" optimistic-lock="true" lazy="false"
generated="never">
- <property name="fechasol" type="timestamp" unique="false" optimistic-lock="true" lazy="false"
generated="never">
  <column name="fechasol" length="19" />
</property>
- <property name="ultfeate" type="timestamp" unique="false" optimistic-lock="true" lazy="false"
generated="never">
  <column name="ultfeate" length="19" />
</property>
  <one-to-one name="nota" class="com.upiicsa.crm.modelo.Nota" constrained="false" embed-
xml="true" />
- <property name="relacion" type="int" unique="false" optimistic-lock="true" lazy="false"
generated="never">
  <column name="actrelid" />
</property>
</class> </hibernate-mapping>
```

## 5.6 Pruebas.

En el RUP, las pruebas son enfocadas a través del uso de un proceso iterativo y de herramientas. Un enfoque iterativo para probar permite a la organización tratar las pruebas casi de la misma forma que el desarrollo de software es enfocado. Cada Construcción de software es un objetivo para las pruebas. Según se vayan produciendo nuevas Construcciones, el cuerpo de pruebas será añadido y refinado. Eventualmente, todas las pruebas en el cuerpo de pruebas serán acumuladas de tal manera que pueden ser usadas para las posteriores pruebas de regresión en el ciclo de vida del desarrollo de software. Este enfoque permite a una organización identificar posibles riesgos al inicio de un proyecto, reducir el costo de corregir fallas enfocando los recursos cuando y donde tendrán el mayor impacto, acercarse a los gaps de calidad tempranamente en el proceso de desarrollo y maximizar la efectividad por medio de adaptar el enfoque, el proceso o el presupuesto según va progresando el proyecto.

Dentro de una iteración, hay varias construcciones probadas: la primera cuando el sistema es integrado y la última para probar todo el sistema.

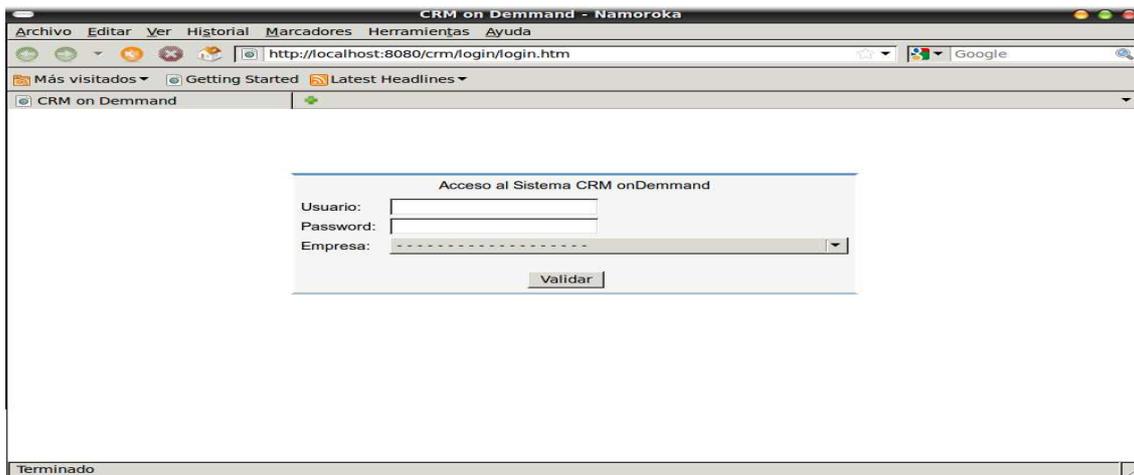
El Workflow de Pruebas consiste de los siguientes Workflows de detalle:

- Planificar las Pruebas  
El principal artefacto producido es el Plan de Pruebas.
- Diseñar las Pruebas  
Los principales artefactos producidos son el Modelo de Pruebas (Test Model), los Casos de Prueba (Test Case), los Procedimientos de Prueba (Test Procedures) y el documento de Análisis de Carga de Trabajo (Workload Analysis Document).
- Implementar las Pruebas  
Los principales artefactos producidos son el Script de la Prueba y el Componente de la Prueba.
- Ejecutar las Pruebas en la etapa de Integración de Pruebas  
El principal artefacto producido es el documento Resultado de Pruebas.
- Ejecutar las Pruebas en la etapa de Pruebas del Sistema  
El principal artefacto producido es el documento Resultado de Pruebas.
- Evaluar las Pruebas  
Los principales artefactos producidos son el Sumario de Evaluación de Pruebas (Test Evaluation Summary) y los Requerimientos de Cambio (Change Request).

## 5.7 Ejercicio Práctico.

Como parte de la realización de las pruebas al sistema se genera un caso practico en el cual mostraremos el funcionamiento del sistema, así como los errores que se pudiesen generar dentro de la operación diaria.

Se ingresa usuario, password y empresa a la que se pertenece.



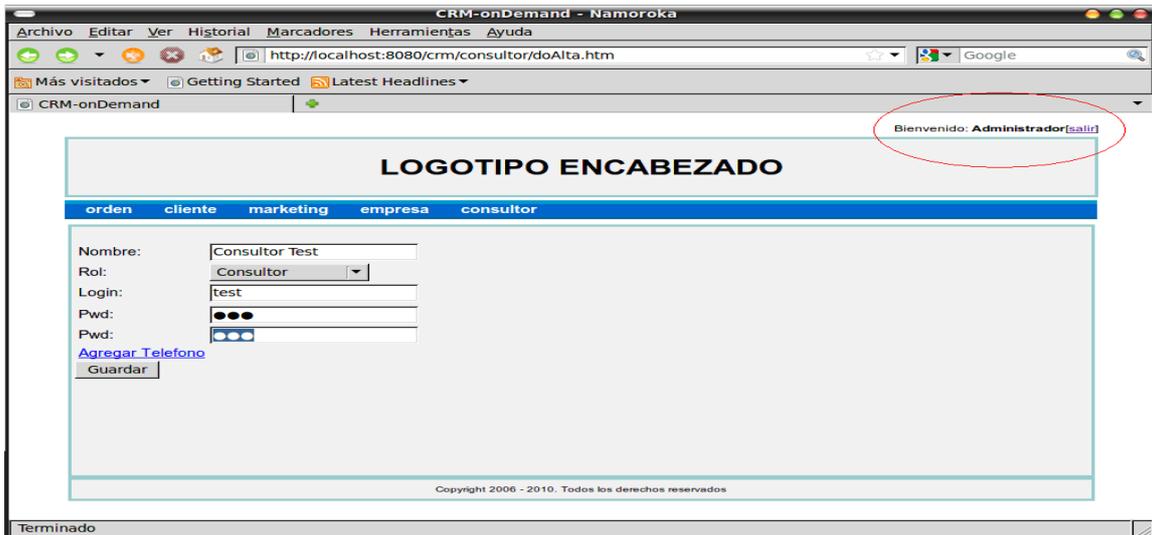
## 5.9 Pantalla Login.

Una vez ingresada la información correspondiente al login, el sistema validara la información. De no existir el usuario o el password sea incorrecto enviara el error. En este caso el login es correcto y nos muestra el menú principal del sistema



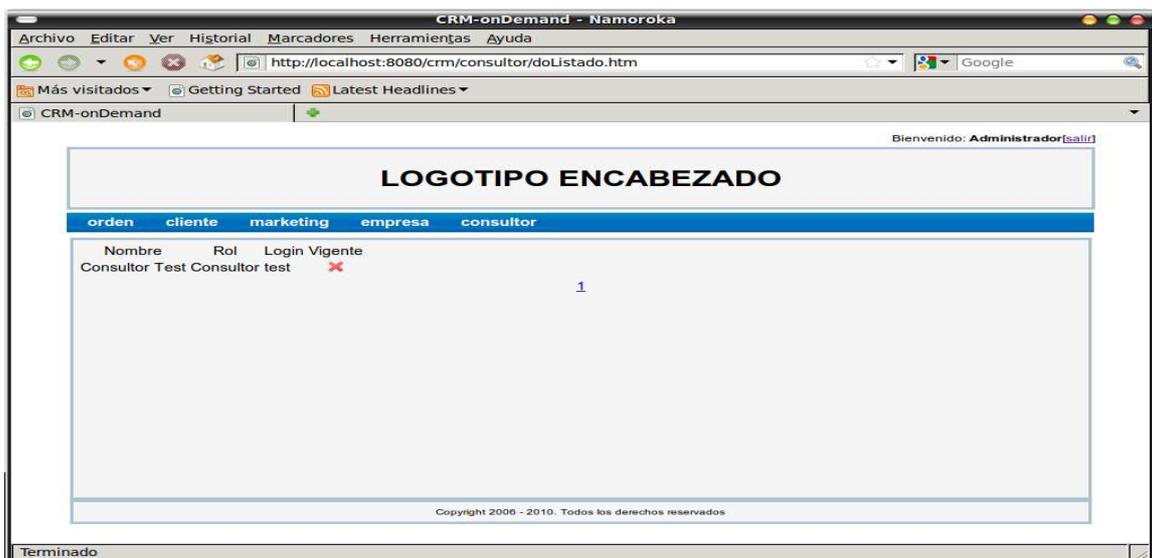
## 5.10 Pantalla Menú Principal.

Ingresaremos la información correspondiente para crear un consultor de la empresa a la cual pertenecemos, el sistema deberá permitirnos ingresar la información siempre que se tenga el perfil adecuado. Como se puede ver en la figura 5.6 el usuario es Administrador, por lo cual tenemos los suficientes privilegios para dar de alta empleados.



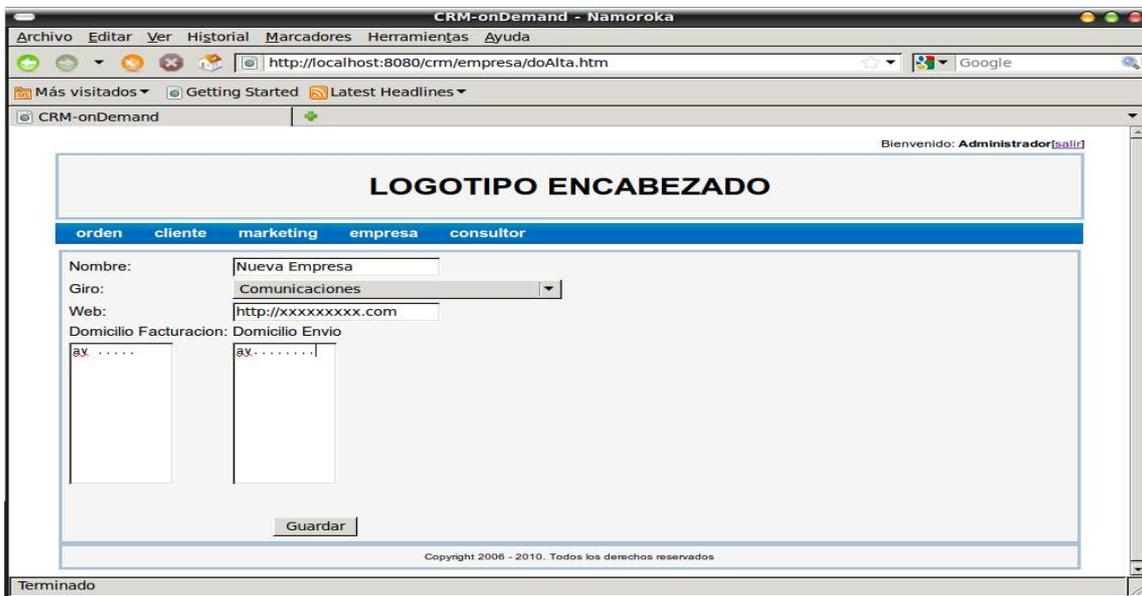
5.11 Pantalla Alta Empleado.

Podremos activar o desactivar usuario según sea el caso, esto con la finalidad de poder llevar el control de los empleados que pertenecen a la empresa.



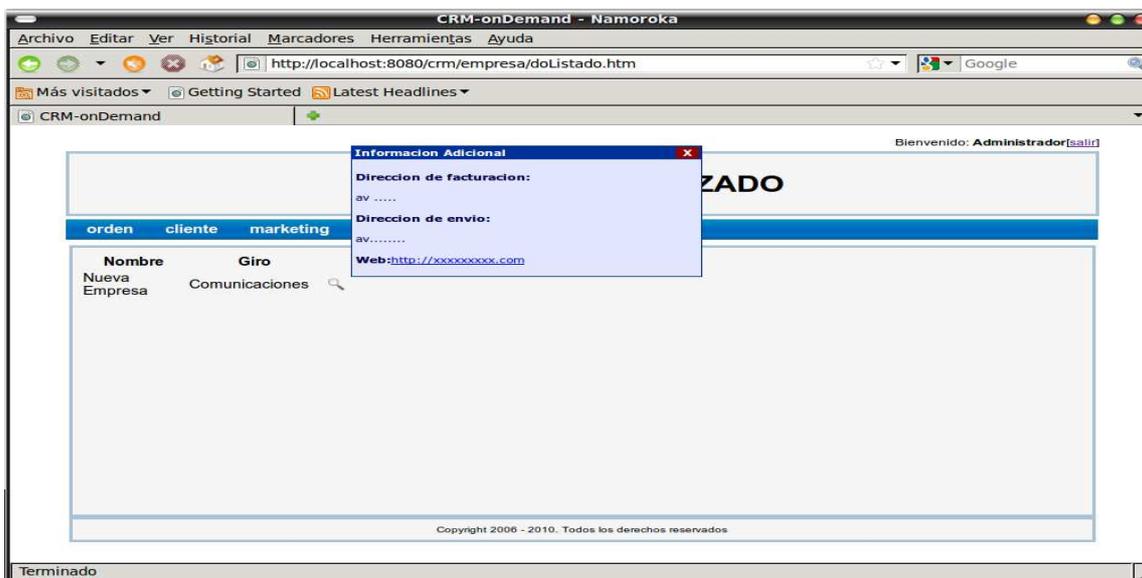
5.12 Pantalla Listado Empleados.

Para agregar empresas, se ingresara la información correspondiente. El sistema validara que los datos sean correctos, que la dirección no exceda el número de caracteres permitidos, y que la pagina Web de la empresa inicie con el estándar de URL.



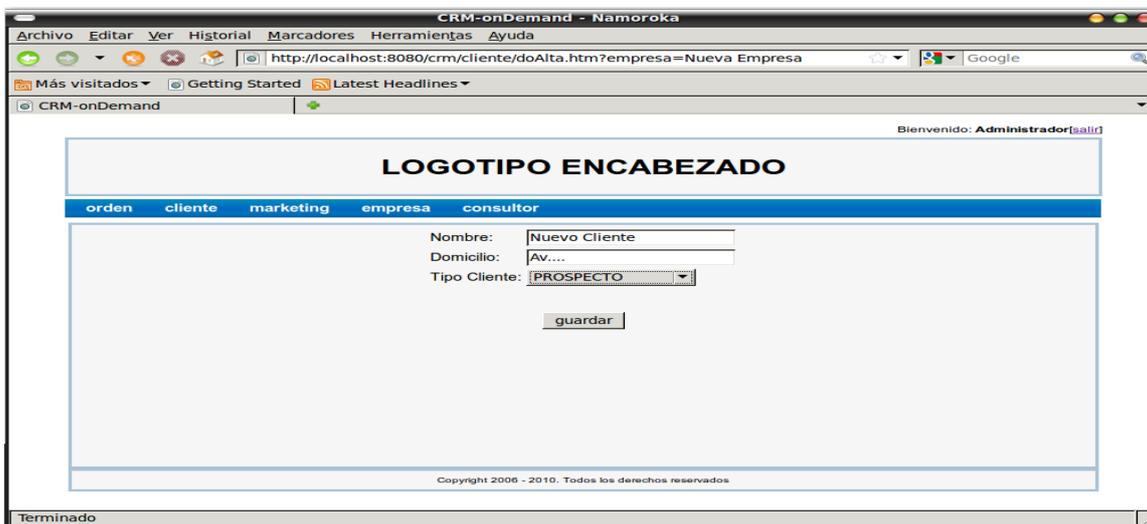
5.13 Pantalla Alta Empresa.

En el listado de empresas ingresadas al sistema se podrá visualizar en todo momento la información perteneciente a la empresa, facilitando la navegación por el sistema.



5.14 Pantalla Listado Empresas.

Los clientes de cada empresa podrán ser ingresados al sistema mediante la captura de la información correspondiente. En este caso no es necesario tener el perfil de Administrador para realizar la operación.



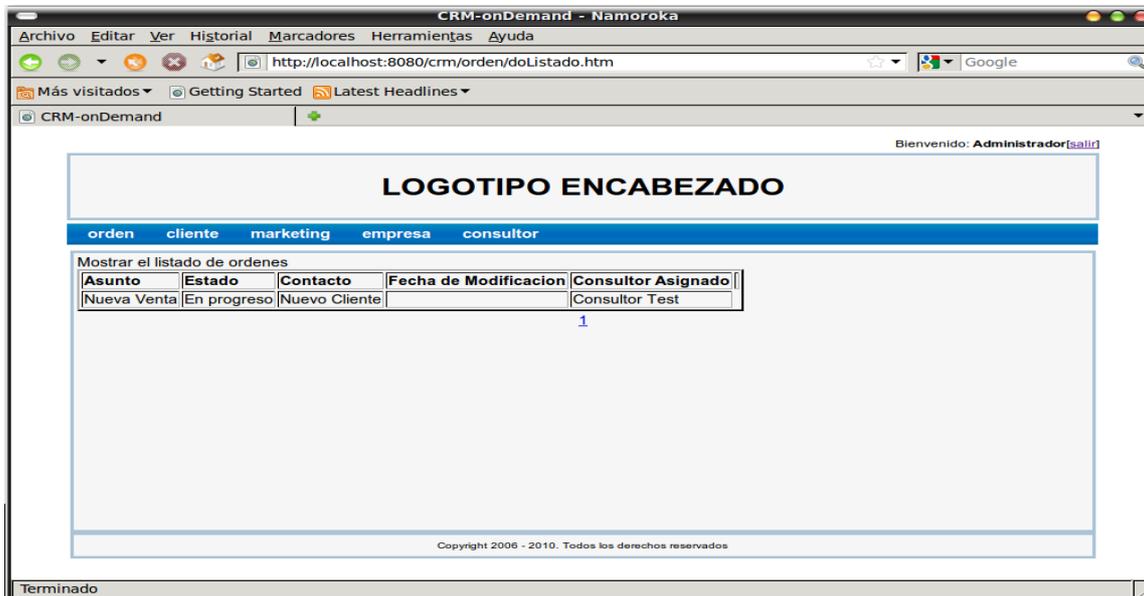
### 5.15 Pantalla Alta Clientes.

La atención al cliente será atendida mediante la generación de una orden, el sistema validará que exista previamente el cliente. De igual forma valida que tengamos empleado a los cuales se les asignará la orden.



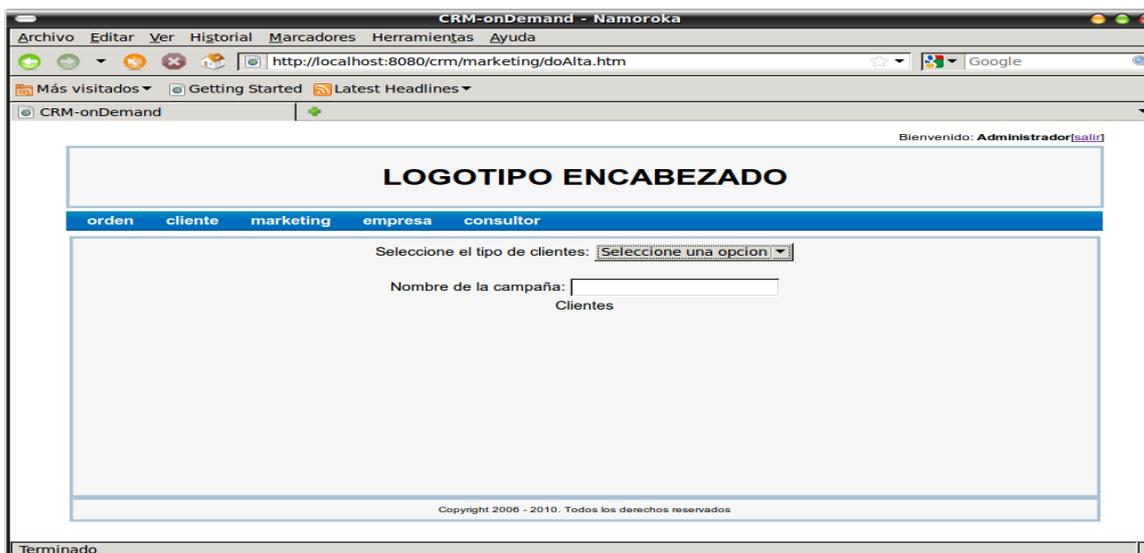
### 5.16 Pantalla Alta Orden.

Se podrá consultar el estatus de la orden, el sistema informara al consultor asignado de la orden permitiendo agregar las actividades correspondientes para la atención y/o solución de la orden.



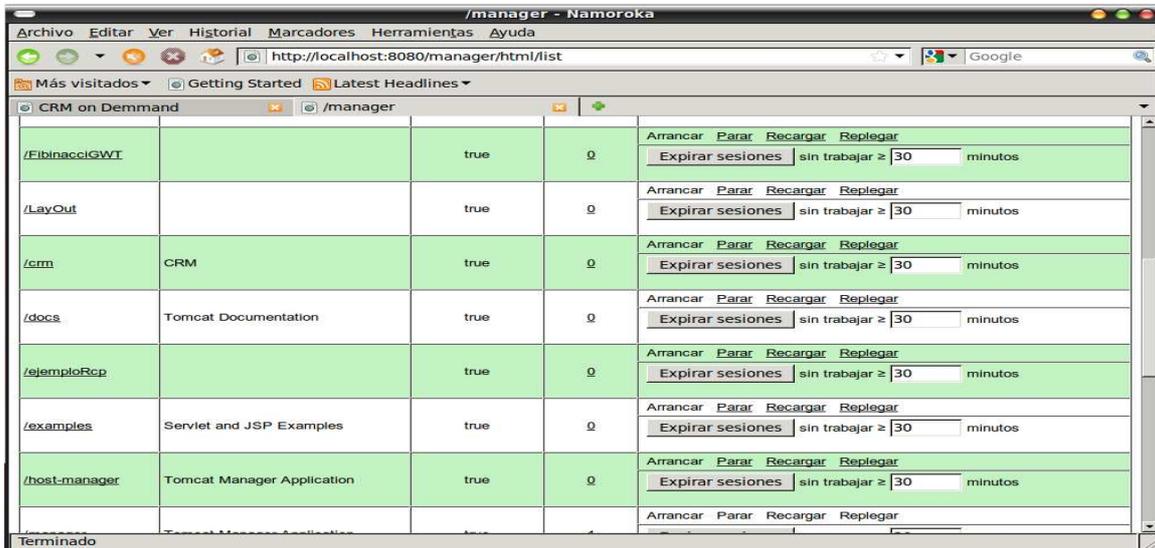
### 5.17 Pantalla Listado Ordenes.

El envío de información de internet permitirá ampliar el mercado actual de la empresa, seleccionando clientes prospectos a los cuales se les puede ingresar en una lista que formará parte de las campañas publicitarias.



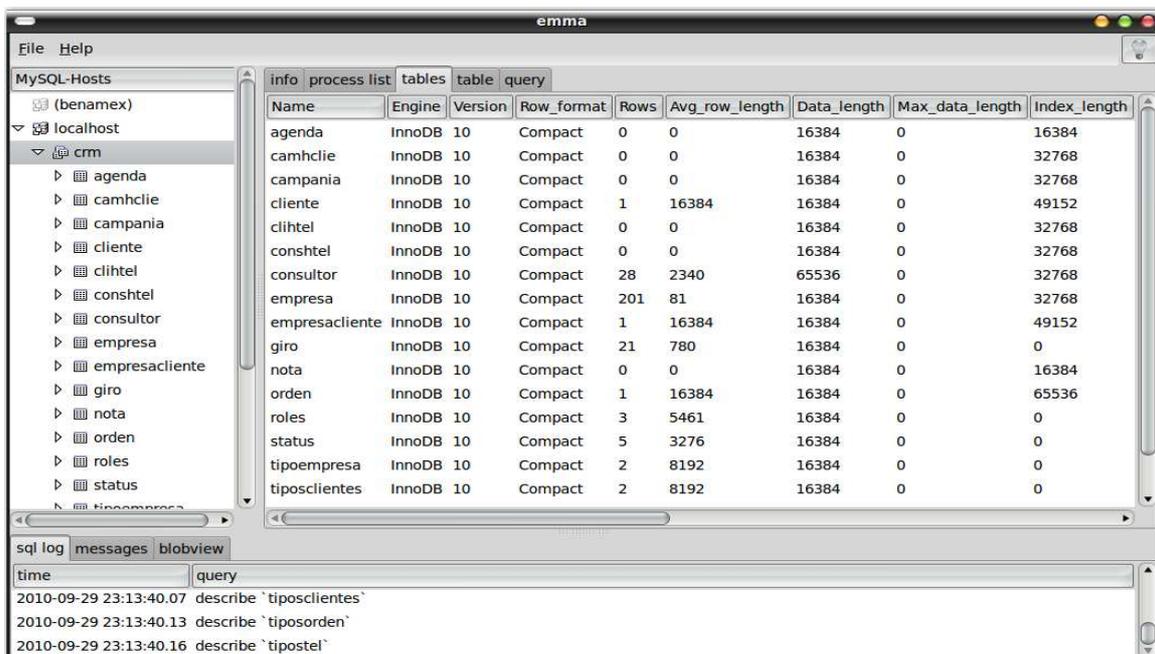
### 5.18 Pantalla Alta Campaña.

El manejo de sesiones es importante dentro de un sistema web, esto con la finalidad de expirar las sesiones despues de cierto tiempo de inactividad por parte del usuario, generando un sistema robusto en cuanto a seguridad se refiere.



5.19 Pantalla Tiempo Sesiones.

La base de datos se diseño con el objetivo de soportar gran cantidad de transacciones en linea, se realizo una prueba de estrés con el fin de garantizar la integridad y disponibilidad de los datos.

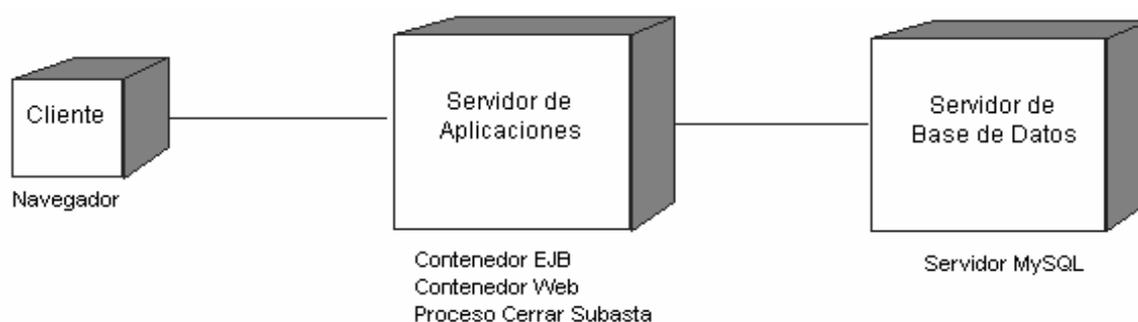


5.20 Pantalla Tablas BD.

## 5.8 Vista Despliegue.

La vista de despliegue del sistema demuestra los nodos físicos en los cuales el sistema se ejecuta y la asignación de los procesos.

La aplicación se puede desplegar en diversas configuraciones de hardware. Por ejemplo, puede ser desplegado en una sola estación de trabajo. La figura 5.14 muestra la configuración más típica del despliegue que debe utilizar el equipo del desarrollo.



**5.21 Pantalla Vista Despliegue.**

## 5.9 Matriz de Pruebas.

La fase de pruebas es una de las más costosas del ciclo de vida software. En sentido estricto, deben realizarse pruebas de todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de productos que forman parte de la aplicación (ejemplo., la base de datos). Obviamente, se aplican diferentes técnicas de prueba a cada tipo de producto software.

La seguridad informática es una subárea de la informática que se enfoca exclusivamente en la protección de la información, lo que comprende software, bases de datos, meta datos, archivos y todo lo que la organización valore (activo) y signifique un riesgo si ésta llega a manos de otras personas. Este tipo de información se conoce como información privilegiada o confidencial.

Cabe mencionar que parte fundamental de la seguridad informática es tener una matriz adecuada del perfilado existente, esto con la finalidad de poder asignar los permisos correspondientes para cada usuario.

La prueba de requisitos pretende comprobar los tres principales atributos de calidad de los requisitos, con el fin de detectar tantos errores como sea posible y cuanto antes: corrección (carencia de ambigüedad), completación (especificación completa y clara del problema) y consistencia (que no haya requisitos contradictorios).

Se proponen la utilización de una Matriz de Prueba de Requisitos (*RTM: Requirements Testing Matrix*), en la que se lista cada requisito junto a sus casos de uso y casos de prueba:

La fase de diseño tiene como objetivo generar un conjunto de especificaciones completas del sistema que se va a implementar, transformando los requisitos en un Plan de implementación.

La prueba del diseño debe comprobar su consistencia, completación, corrección, factibilidad (es decir, que el diseño sea realizable) y trazabilidad (es decir, que podamos “navegar” desde un requisito hasta el fragmento del diseño en que éste se encuentra).

Las revisiones e inspecciones de código fuente son una técnica para la detección manual de errores en el código. Para ello suelen utilizarse listas de comprobación (*checklists*), que enumeran defectos y en los que el revisor anota su presencia o ausencia. Puede consultarse una lista de comprobación de errores típicos del lenguaje Java, entre los que se incluyen el control de overflows y underflows, restricción adecuada del acceso a los miembros de las clases, control de apertura y cierre de ficheros, etc.

A continuación listaremos los resultados obtenidos en las pruebas realizadas para cada una de las áreas dentro del sistema, estas se realizaron dependiendo del perfilado en el sistema.

Área	Función/Tarea	Prueba	Resultados Esperados	Resultados de la Prueba	Comentarios
Seguridad	Autenticación	Usuario no existe	Invalid username / password	Invalid username / password	OK
		Contraseña incorrecta	Invalid username / password	Invalid username / password	OK
		Usuario /contraseña correcta	Conectarse a la aplicación, aparecer Menú Principal	Se conectó a la aplicación, apareció Menú Principal	OK
		Presionar botón Cancel	Salir del programa	Terminó el programa	OK

Área	Función/Tarea	Prueba	Resultados Esperados	Resultados de la Prueba	Comentarios
Negocio	Rol Administrador	Generación de Menú Administrador.	Generación correcta del menú	Menú	OK
		Altas Empresa	Generar una nueva Empresa	Nueva Empresa	OK
		Listado Empresa	Listar solo las Empresas pertenecientes	Lista correcta	OK
		Alta Contacto	Asociar el nuevo contacto a la Empresa seleccionada	Relación	OK
Negocio	Rol Ejecutivo de Ventas	Alta Empresa	Nueva Empresa	Flujo total de alta de una Orden y seguimiento	OK
		Alta Contacto	Nuevo Contacto Asociado a la Empresa	Flujo total de alta de una Orden y seguimiento	OK
		Nueva Orden	Nueva Orden	Flujo total de alta de una Orden y seguimiento	OK
		Seguimiento Orden	Flujo total de alta de una Orden y seguimiento	Flujo total de alta de una Orden y seguimiento	OK
		Listados	Lista Empresa, Contacto y Ordenes	Flujo total de alta de una Orden y seguimiento	OK

**5.22 Matriz resultados.**

## **CONCLUSIONES.**

El haber realizado el análisis, diseño y desarrollo de un CRM onDemand para la gestión de las relaciones con el Cliente, dio como resultado una herramienta de gran utilidad con grandes potencialidades, no solo debido a la tecnología implementada; sino también a la solución generalizada y principalmente a la manera en que se planea hacer negocio con la herramienta misma, ir creciendo todos juntos, para al final lograr obtener un Software lo bastante robusto que implemente soluciones generales a cualquier problemática que las empresas pudiesen presentar en el ejercicio de la gestión de las relaciones con sus Clientes, obteniendo así un buen posicionamiento en el ámbito de los negocios y de las soluciones tecnológicas.

Debido a que realizamos todo el proceso de desarrollo de software utilizando el paradigma de orientación a objetos, fue posible llevar a cabo el proceso desde una perspectiva más apegada a la realidad, lo cual nos permitió realizar todas las actividades de una manera más ágil y rápida, permitiendo enfocarnos en lograr el sistema que se planeó.

La experiencia profesional que poseemos en el área del desarrollo de software resultó muy útil en cada etapa del proceso, pues cada uno de los integrantes del equipo aportó los conocimientos, aptitudes y habilidades adquiridos durante su vida escolar y laboral. Asimismo, el desarrollo de éste sistema también nos permitió adquirir más conocimientos, experiencia y complementar nuestra formación como informáticos, pudiendo demostrar que somos capaces de brindar soluciones tecnológicas a los problemas que nos presenten. Además de desarrollar una visión emprendedora, capaz de observar oportunidades de negocio, donde podremos competir.

## BIBLIOGRAFÍA.

Schmuller Joseph. Aprendiendo UML en 24 horas, Editorial Prentice Hall, México, 2000.

Ecco, Umberto, Como se hace una tesis, Editorial Gedisa, México, 2006.

Hernández R., Fernández C. & Baptista P. Metodología de la Investigación, 3ª edición, Editorial Mc. Graw Hill, México, 2003.

Ceballos Sierra, Francisco Javier. Java 2 Interfaces Graficas y Aplicaciones para Internet. Editorial Alfaomega, México.

Gich J, Teoría General de Sistemas. Editorial Trillas, México, 2003.

Bertalanffy, L. Teoría General de Sistemas, Editorial Fondo de Cultura Económica, México.

### Documentación en Línea

MySQL documentación, <http://www.mysql.com>, Recuperado Febrero/2010.

NetBeans, <http://www.netbeans.org>, Recuperado Febrero/2010

Eclipse IDE, <http://www.eclipse.org>, Recuperado Febrero/2010

JSP Technology, <http://java.sun.com/products/jsp/index.jsp>, Recuperado Febrero/2010

Java Servlet Technology, <http://java.sun.com/products/servlet/index.jsp>, Recuperado Febrero/2010

HTML 4.01 Specification, <http://www.w3.org/TR/REC-html40/>, Recuperado Febrero/2010

UML Resource Page, <http://www.uml.org/>, Recuperado Febrero/2010

Spring Hispano, <http://www.springhispano.org>, Recuperado Febrero/2010

Sistemas CRM, <http://www.fotosok.com/sistemascrm/crm-on-demand-crm-web.htm>, Recuperado Febrero/2010

Metodología CRM, <http://www.mundocrm.com/noticias/view-topic:5-Software-CRM.html>,  
Recuperado Febrero/2010

Marketing CRM, <http://www.marketing-xxi.com/crm-costumer-relationship-management-o-gestion-de-las-relaciones-con-los-clientes-61.htm>, Recuperado Febrero/2010

Teorías del CRM,

<http://www.gerenciaynegocios.com/teorias/crm/03%20%20ventajas%20y%20desventajas.htm>,

Recuperado Febrero/2010