

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Desarrollo de Sistema Negocio a Cliente (B2C) para
Venta de Servicios Electrónicos

Trabajo de graduación en modalidad de tesis presentado por
Juan Diego Dighero Barragán
para optar al grado académico de Licenciado en
Ingeniería en Ciencias de la Computación

Guatemala
2013

**Desarrollo de Sistema Negocio a Cliente (B2C) para
Venta de Servicios Electrónicos**

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería




Desarrollo de Sistema Negocio a Cliente (B2C) para
Venta de Servicios Electrónicos

Trabajo de graduación en modalidad de tesis presentado por
Juan Diego Dighero Barragán
para optar al grado académico de Licenciado en
Ingeniería en Ciencias de la Computación


Guatemala
2013

Vo. Bo. :


(f) 

Ing. Bidkar Pojoy
Asesor

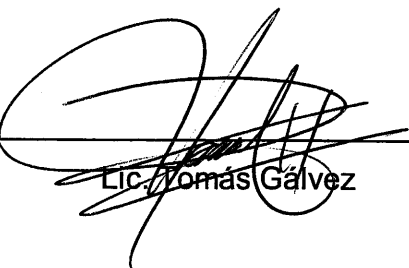
Tribunal Examinador:

(f) 

MSc. Douglas Barrios

(f) 

Ing. Bidkar Pojoy, Asesor

(f) 

Lic. Tomás Gálvez

Fecha de aprobación: Guatemala, 8 de julio 2013

Prefacio

Este proyecto fue desarrollado debido a la necesidad que tuvo la empresa Grupo TIR de tener un sistema capaz de realizar la venta de su catálogo de productos electrónicos de una forma eficaz y segura, en el que también se pueden realizar las configuraciones necesarias y consultar las ventas de los productos. A su vez este documento representa el fin de los estudios de la carrera en el cual se aplicaron los conceptos aprendidos durante el periodo de estudios de una forma práctica y diferente a la acostumbrada.

ÍNDICE

	Página
Prefacio.....	vi
Lista de gráficos.....	ix
Lista de tablas.....	x
Resumen.....	xii
I. Introducción.....	1
II. Objetivos.....	3
A. Objetivo general.....	3
B. Objetivos específicos.....	3
III. Marco teórico.....	4
A. Bases de datos.....	4
1. Ventajas de un manejador de base de datos.....	4
B. Modelo entidad relación.....	5
1. Análisis de requerimientos.....	5
2. Diseño conceptual de la base de datos.....	5
3. Diseño lógico de la base de datos.....	5
4. Refinamiento del esquema.....	6
C. Modelo relacional.....	7
D. XML.....	8
E. Red privada virtual (Virtual Private Networking, VPN).....	8
F. Pruebas de aceptación de usuario (User Acceptance Testing, UAT).....	9
G. Pruebas unitarias.....	10
H. Metodología de desarrollo de cascada.....	10
1. Análisis de requerimientos.....	10
2. Diseño.....	10
3. Desarrollo o implementación.....	11
4. Pruebas e instalación.....	11
IV. Metodología.....	12
A. Herramientas.....	12
V. Análisis.....	14
A. Análisis del negocio.....	14

B.	Análisis de requerimientos.....	15
1.	Productos.....	15
2.	Cliente.....	16
3.	Cuentas.....	16
VI.	Diseño.....	19
A.	Diseño del proyecto.....	19
1.	Bases de datos.....	19
2.	Servicios web.....	19
3.	Driver del nuevo producto.....	19
4.	Modelo del nuevo producto.....	19
5.	Firewall y VPN.....	19
B.	Bases de datos.....	20
C.	Servicios web.....	24
1.	Servicio web para el cliente.....	24
2.	Servicio web para el distribuidor.....	27
3.	Códigos de respuesta de los servicios.....	29
D.	Driver y modelo del producto nuevo.....	30
E.	Estados de una solicitud.....	31
F.	Consultas al distribuidor.....	32
G.	Comparación fonética.....	33
1.	Double Metaphone.....	33
H.	Manejo de bitácoras.....	36
I.	Interfaz gráfica del configurador de productos.....	37
VII.	Discusión.....	38
VIII.	Pruebas e instalación.....	39
A.	Pruebas.....	39
1.	Pruebas unitarias.....	39
2.	Pruebas de aceptación de usuario.....	39
3.	Pruebas de estrés.....	40
B.	Instalación.....	41
IX.	Impacto y próximos desarrollos.....	42
X.	Conclusiones y recomendaciones.....	43
XI.	Bibliografía.....	44

XII. Glosario.....	45
--------------------	----

LISTA DE GRÁFICOS

	Página
1. El conjunto de entidades empleados.....	6
2. El conjunto de relaciones trabaja_en.....	7
3. Red privada virtual.....	9
4. Diagrama de despliegue del proyecto.....	20
5. Código Ejemplo de la utilización de procedimientos almacenados.....	22
6. Diagrama entidad relación del configurador de productos.....	23
7. Diagrama entidad relación de la base de datos transaccional.....	23
8. Diagrama de secuencia de compra.....	25
9. Diagrama de secuencia de compra, receptor no afiliado.....	25
10. Secuencia de compra, pérdida de conexión cliente.....	26
11. Secuencia de compra, pérdida de conexión distribuidor.....	27
12. Diagrama de secuencia de afiliación de receptor del producto.....	28
13. Diagrama de secuencia de confirmación de entrega del producto.....	29
14. Uso de la librería de C# para la construcción de XML.....	30
15. Diagrama de estados de las solicitudes.....	32
16. Secuencia de consulta de saldo y transacciones diarias.....	33
17. Algoritmo implementado para la comparación de cadenas.....	35
18. Ejemplo de archivo de configuración de NLog.....	36
19. Ejemplo de uso de NLog.....	37

LISTA DE TABLAS

	Página
1. Códigos de respuesta del servicio web.....	29
2. Estados de una solicitud del nuevo producto.....	31
3. Resultados del algoritmo Double Metaphone.....	34
4. Resultados del algoritmo distancia Levensthein.....	34
5. Resultados de la implementación para la comparación fonética.....	35
6. Plantilla utilizada para las pruebas de aceptación de usuario.....	40

Resumen

El proyecto desarrollado para Grupo TIR busca resolver la problemática de la empresa el cual no permite al usuario final realizar configuraciones o cambios en los productos electrónicos de una forma fácil y rápida, como tampoco un fácil crecimiento de su catálogo de productos lo ha conllevado a malos funcionamientos como por ejemplo la inestabilidad en días donde se recibe un número de transacciones altos, el proyecto también incluye el desarrollo del despacho de uno de los primeros productos que funcionaran bajo este esquema.

La venta de los productos electrónicos se realiza a través de un servicio web y anteriormente funcionaba de tal forma en que las solicitudes eran recibidas y al cliente se le devolvía un estado exitoso, estas no eran procesadas en el momento con el distribuidor lo cual ocasionaba mucha desconfianza y molestia de parte de los clientes ya que muchas veces el producto no era recibido en un tiempo corto o muchas veces este no era ni siquiera recibido en el mismo día, por último en el caso que ocurriera algún problema en el despacho del producto con el distribuidor el cliente no se enteraba ya que las solicitudes únicamente eran encoladas y no se le daba un resultado en el momento de lo ocurrido. Con el desarrollo del proyecto se busca que las futuras implementaciones de despacho de productos electrónicos sean realizados en línea, es decir que al momento en que el cliente realice su solicitud ésta será procesada y se retroalimentara el resultado de la operación ya sea una respuesta exitosa o fallida.

I. Introducción

El curso de Taller de Proyecto Profesional de la Universidad del Valle de Guatemala tiene como requisito que los estudiantes en su último año de carrera deben de realizar una práctica profesional que consta de 400 horas de trabajo en una empresa, en la que deberán de solucionar un problema o desarrollar el proyecto por si mismos aplicando los conocimientos aprendidos durante la carrera de Ingeniería en Ciencias de la Computación de una forma práctica.

En este caso se tuvo la oportunidad de realizar las prácticas en la empresa Grupo TIR la cual tiene ya diez años operando. La empresa se dedica al desarrollo inmobiliario y a las telecomunicaciones siendo en esta la rama donde se desarrolló este proyecto. Los desarrollos de software que se realizan en la empresa son para uso interno o para ofrecer un servicio o producto electrónico. Las ventas de los productos electrónicos se han realizado vía un servicio web que atiende las peticiones de los clientes que se conectan al mismo.

La venta de los productos electrónicos se realiza de una forma correcta hoy en día, aunque internamente es muy complicado obtener la información de ventas y muchas veces esta no es congruente y requiere mucha labor operativa para determinar cuáles han sido los problemas y obtener la información correcta, otro problema que se ha presentado es que la configuración de los productos no es fácil y requiere de procesos no documentados para realizarlas, lo cual muchas veces a conllevado a problemas operativos. La forma en que se despachan los productos sin importar cuál sea actualmente ha demostrado que presenta muchos inconvenientes ya que cuando el cliente realiza su solicitud vía el servicio web no se le devuelve el resultado de la operación si no que solo es aceptada y encolada y previamente es procesada lo cual puede ocasionar que la solicitud tenga un tiempo de despacho alto o esta tenga algún error.

Debido al reciente crecimiento de la empresa y a los problemas del sistema mencionados anteriormente se decidió conceptualizar de nuevo toda la venta de los productos electrónicos, no solo para poder despachar los productos de una mejor forma sino que también para poder dar un mejor servicio a los clientes y por ultimo para darle la potestad al usuario final de realizar las configuraciones que el desee sin necesidad de que este proceso sea complicado. Ya que la empresa está ampliando su catálogo de productos y ha decidido incorporar un nuevo producto se decidió que este nuevo sea el modelo a seguir en la forma en que se despachan los

productos la cual se realizara en línea, este funcionara de tal forma en que las solicitudes se procesan en línea.

El trabajo se limita al análisis y diseño del sistema de configuración de productos electrónicos de la empresa y no al desarrollo de la interfaz gráfica como también realizar el desarrollo del despacho de las solicitudes del nuevo producto el cual servirá de modelo para los futuros desarrollos de despacho de productos.

El proyecto se realizó utilizando la metodología de desarrollo de cascada la cual ofrece muchas ventajas como la que cada fase tiene del proyecto tiene un inicio y un fin bien definido y también ayuda a que se realice un buen análisis y diseño para que el desarrollo del mismo sea fácil y rápido.

II. Objetivos

A. Objetivo general

Crear una plataforma sostenible y escalable para la venta de productos electrónicos.

B. Objetivos específicos

1. Resolver la problemática actual en la que la configuración de los productos no es fácil y resulta muy complicado para el usuario final.
2. Permitir a la empresa la venta de un nuevo producto mediante el desarrollo del despacho del mismo.
3. Desarrollar el proyecto según los requerimientos y lineamientos establecidos para poder obtener los resultados deseados.
4. Que el desarrollo del despacho de este producto sea un modelo a seguir para los próximos desarrollos.
5. Diseñar y desarrollar un proyecto de software utilizando los conocimientos aprendidos durante la carrera.

III. Marco teórico

A. Bases de datos

Una base de datos es una colección de datos. Un manejador de base de datos es un software que está diseñado para crear y manejar grandes cantidades de información eficientemente y permitiendo que esta persista durante largos periodos de tiempo. Estos sistemas se encuentran entre los tipos de software

Una base de datos es una colección de datos. Un manejador de base de datos es un software que está diseñado para crear y manejar grandes cantidades de información eficientemente y permitiendo que esta persista durante largos periodos de tiempo. Estos sistemas se encuentran entre los tipos de software más complejos disponibles.

1. Ventajas de un manejador de base de datos

- Independencia de la información: los manejadores de bases de datos pueden proporcionar una visión abstracta de la información para así poder aislar al código de la aplicación de tales detalles.
- Acceso a la información eficiente: un manejador de base de datos utiliza técnicas sofisticadas para almacenar y obtener información eficientemente.
- Integridad y seguridad de la información: si la información siempre es ingresada por medio del manejador de base de datos, este puede garantizar que se cumplan las restricciones de integridad en los datos. Por ejemplo, antes de ingresar información acerca del salario de un trabajador, el manejador puede verificar que el presupuesto del departamento no se exceda. Asimismo, el manejador puede hacer cumplir los controles de acceso que determinan que información puede ser visible para las diferentes clases de usuarios.
- Administración de la información: cuando varios usuarios comparten la información, la centralización de la información puede ofrecer mejoras significativas.
- Acceso concurrente y recuperación de errores: un manejador de base de datos calendariza los accesos concurrentes a la información de tal forma que para el usuario pueda parecer que solo está siendo consultada por un usuario. Además, el usuario protege a los usuarios de los efectos de los fallos del sistema.

- Disminuye el tiempo de desarrollo de aplicaciones: Claramente, el manejador de base de datos soporta muchas funciones importantes que son comunes a muchas aplicaciones que acceden a la información almacenada en el manejador. Esto, en combinación con la interfaz de alto nivel de la información, facilita el desarrollo de aplicaciones. Estas aplicaciones tienden a ser más robustas que las aplicaciones desarrolladas desde cero, ya que muchas tareas son manejadas por el manejador en lugar de ser implementadas por la aplicación.

B. Modelo entidad relación

El modelo ER es importante ante todo por el rol que juega en diseño de una base de datos. Provee conceptos útiles que nos permiten pasar de la descripción informal de que es lo que el usuario quiere de su base de datos a algo más detallado, y preciso, descripción que puede ser implementada en un manejador de base de datos.

Una relación puede ser descrita como un conjunto de registros. Una descripción de la información en términos del modelo de datos se le denomina esquema. En el modelo relacional, el esquema de una relacional define su nombre, el nombre de cada campo (o atributo o columna), y el tipo de cada campo.

El proceso del diseño de una base de datos puede estar dividido en seis pasos. El modelo ER es más relevante en estos cuatro primeros pasos:

1. Análisis de requerimientos: el primer paso en el diseño de la base de datos de una aplicación es entender cuál es la información que se va a almacenar en la base de datos, que aplicaciones deben de estar construidas sobre la misma, y que operaciones son las más frecuentes y sujetas a los requerimientos de desempeño. En otras palabras, se debe de encontrar que es lo que los usuarios quieren de la base de datos.

2. Diseño conceptual de la base de datos: la información obtenida en el análisis de requerimientos es utilizada para desarrollar una descripción de alto nivel de la información que va a ser almacenada en la base de datos, junto con las restricciones que deben de regir esta información.

3. Diseño lógico de la base de datos: se debe de escoger el manejador de base de datos para implementar el diseño de la base de datos, y convertir el diseño conceptual en el esquema de base de datos en el modelo de información del manejador de base de datos. Solo se

consideran manejadores de base de datos relacionales, y por lo tanto, la tarea en diseño lógico es convertir el esquema entidad relación a un esquema de base de datos.

4. Refinamiento del esquema: este paso consiste en analizar la colección de relaciones en nuestro esquema de la base de datos para identificar problemas potenciales, y refinarlos. En contraste a los pasos de análisis de requerimientos y al diseño conceptual, que son esencialmente subjetivos, el refinamiento del esquema puede ser guiado por una teoría muy elegante y poderosa llamada normalización.

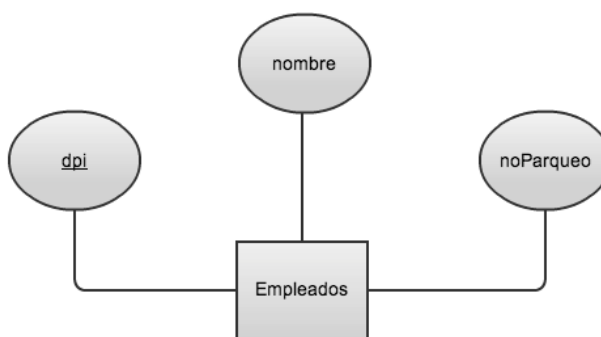
Una entidad es un objeto del mundo real que es distinguible de otros objetos. Ejemplos incluyen los siguientes: el departamento de juguetes, el gerente del departamento de juguetes, la dirección de casa del gerente de juguetes. A menudo es útil identificar una colección de entidades similares.

Una entidad es descrita utilizando atributos. Todas las entidades en un determinado conjunto de entidades tienen los mismos atributos. Por ejemplo, la entidad Empleados puede utilizar el nombre, número de parqueo y DPI como atributos. En este caso se almacenará para cada empleado su nombre, número de parqueo y DPI.

Para cada atributo asociado con un conjunto de entidades, se debe de identificar un dominio para los posibles valores. Por ejemplo, el dominio asociado con el atributo nombre de Empleados debe de ser un conjunto de 20 caracteres. Por aparte para cada entidad se elige una llave. Una llave es un conjunto mínimo de atributos cuyos valores identifican de forma única una entidad en el conjunto. Puede haber mas de una llave candidata, si este es el caso, se designa una de ellas como la llave primaria.

El conjunto de entidades Empleados con los atributos dpi, nombre y noParqueo es representado en la Figura 1. Una entidad es representada como un rectángulo, y un atributo es representado como un ovalo. Cada atributo en la llave primaria es subrayado.

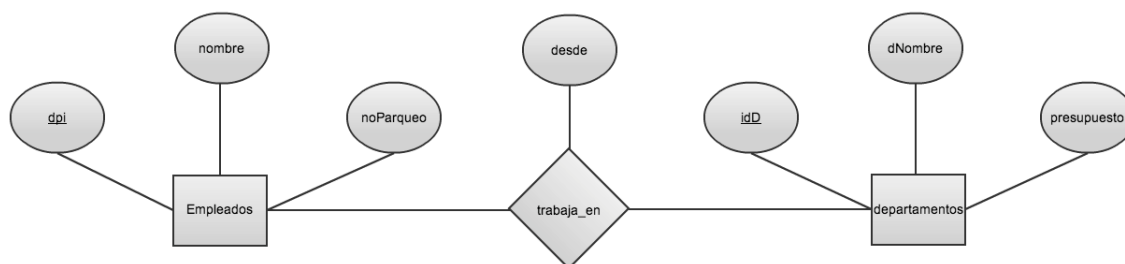
Figura No.1. El conjunto de entidades empleados



Una relación es una asociación entre una o más entidades. Por ejemplo, se puede tener una relación en donde Juan trabaja para el departamento de farmacéuticos. Al igual que con las entidades, se pueden reunir conjuntos de relaciones similares en un conjunto de relaciones.

En la Figura 2, se muestra el conjunto de relaciones trabaja_en, en donde cada relación indica en qué departamento trabaja cada empleado.

Figura No.2. El conjunto de relaciones trabaja_en.



Una relación también puede tener atributos descriptivos. Los atributos descriptivos son utilizados para registrar información acerca de la relación, en lugar registrar información acerca de las relaciones participantes; por ejemplo, si se quisiera registrar que Juan trabaja en el departamento de farmacéutica desde enero de 1988. Esta información es capturada en la Figura 2 agregando el atributo desde a trabaja_en. Una relación debe de ser identificada de forma única por las entidades participantes, sin hacer referencia a los atributos descriptivos. En la relación trabaja_en, por ejemplo, cada relación trabaja_en debe ser identificada de forma única mediante la combinación del dpi del empleado y del dld del departamento. Por lo tanto, para un determinado par de empleado-departamento, no se puede tener más de un valor asociado de desde.

C. Modelo relacional

El modelo relacional es muy simple y elegante; una base de datos es una colección de una o más relaciones, donde cada relación es una tabla con filas y columnas. Esta representación tabular permite que incluso los usuarios no experimentados entiendan los contenidos de la base de datos, y permite el uso de lenguajes de alto nivel consulten su información. Las principales ventajas del modelo relacional sobre otros modelos de información son la simplicidad de la

representación de la información y la simplicidad con la que consultas complejas pueden ser expresadas.

Uno de los pilares importantes para representar la información en el modelo relación es la relación. Una relación consiste de un esquema de la relación y de una instancia de la relación. Una instancia de la relación es una tabla, y el esquema de la relación describe los encabezados de la tabla. Primero se describe el esquema de la relación y luego la instancia de la relación. El esquema especifica el nombre de la relación, el nombre de cada campo (o columna, o atributo), y el dominio de cada campo.

D. XML

XML es el acrónimo de *extensible markup language* y fue desarrollado y finalizado por W3C (World Wide Consortium) en 1998. XML es un documento en texto plano, con estructura de árbol que es leíble para las personas y también para diferentes tipos de software. Es una versión más liviana y flexible que su predecesora el *Standard General Mark-up Language* (SGML).

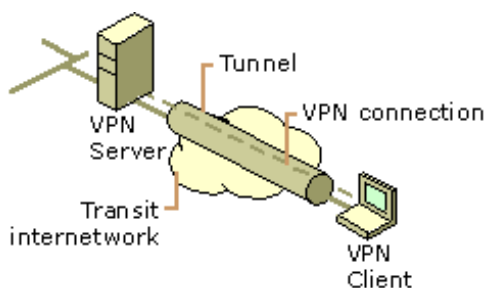
Los elementos del XML también conocidos como *tags* son los elementos principales de un documento XML. El inicio del contenido de un elemento esta marcado con `<ELM>`, la cual es llamado el *tag* inicial, y al final del contenido se marca con `</ELM>`, llamado el *tag* final. Un ejemplo de un XML es el siguiente.

```
<Libro>
  <Autor>
    <Nombre>Juan Alberto</Nombre>
  </Autor>
</Libro>
```

E. Red privada virtual (Virtual Private Networking, VPN)

Una red privada virtual es una extensión de una red privada que incluye vínculos de redes compartidas o públicas como el Internet. Una VPN permite enviar datos entre dos ordenadores a través de una red interna compartida o pública de forma que emula las propiedades de un enlace punto a punto privado.

Figura No. 3. Red privada virtual



Para emular un enlace punto a punto, los datos se encapsulan o empaquetan con un encabezado que proporciona la información de enrutamiento que le permite recorrer el tránsito compartido o público para llegar a su punto final. Para emular un vínculo privado, los datos que se envían es cifrada por confidencialidad. Los paquetes interceptados en la red compartida o pública no se pueden descifrar sin las claves de cifrado. La porción de la conexión en la que se encapsulan los datos privados se conoce como el túnel. La porción de la conexión en la que los datos privados están cifrados se conoce como la conexión a la red privada virtual.

Las conexiones VPN permiten a los usuarios que trabajan en casa de una forma segura a un servidor remoto corporativo. Desde la perspectiva del usuario, la conexión VPN es una conexión punto a punto entre su ordenador y un servidor corporativo.

F. Pruebas de aceptación de usuario (User Acceptance Testing, UAT)

Una prueba de aceptación tiene como propósito demostrar al cliente el cumplimiento de un requisito del software. Es una fase en donde el software es probado en un ambiente controlado similar al ambiente de producción del software. Es una forma segura de reducir o eliminar las solicitudes de cambio, y reducir drásticamente los costos del proyecto. Realizar las pruebas tiene como fin varios propósitos, pero entre lo más importante es que se puede medir la calidad de una determinada aplicación.

El objetivo de las pruebas de aceptación de usuario es evaluar si el sistema puede soportar la carga del día a día del negocio y los escenarios de los usuarios y asegurar que el sistema es suficiente y adecuado para su uso en los negocios. Las tareas de las pruebas de aceptación de usuario son:

1. Analizar los requerimientos del negocio.
2. Identificar los escenarios UAT.
3. Definir el plan de pruebas de UAT.
4. Crear los casos de prueba UAT.
5. Ejecutar las pruebas.
6. Registrar los resultados.
7. Confirmar que los objetivos del negocio han sido alcanzados.

G. Pruebas unitarias

Las pruebas unitarias es un proceso de prueba de subprogramas individuales, subrutinas, o procedimientos en un programa. Es decir, en lugar de probar la aplicación como un todo, las pruebas son enfocadas en los bloques de construcción más pequeños de la aplicación. Este tipo de pruebas permiten gestionar los elementos combinados de las pruebas, ya que la atención se centró inicialmente en unidades más pequeñas del programa, estas pruebas también facilitan la tarea de depuración de código, ya que cuando se presenta un error particular se sabe específicamente en donde se originó.

H. Metodología de desarrollo de cascada

La metodología de desarrollo de cascada es un proceso estructura en donde la salida de una de las fases representa una de las entradas de la siguiente fase, cada fase está estructurada de tal forma que se lleven a cabo diferentes actividades. Las fases de esta metodología son:

1. Análisis de requerimientos: en esta fase se especifican los requerimientos y las necesidades del proyecto. El propósito de esta fase es identificar las cualidades requeridas para la aplicación en términos de funcionalidad. Los requerimientos definirán que va a hacer el sistema y no el cómo.
2. Diseño: el propósito de esta fase es transformar los requerimientos establecidos en

una estructura que sea sostenible para el desarrollo del proyecto por ejemplo la arquitectura general.

3. Desarrollo o implementación: durante esta fase se utiliza el diseño realizado en la fase anterior para poder implementar o desarrollar el proyecto.

4. Pruebas e instalación: el objetivo de realizar las pruebas es determinar si el proyecto cumple con los requerimientos previamente establecidos y determinar si tiene un funcionamiento correcto.

IV. Metodología

Para el desarrollo de este proyecto se utilizó la metodología de cascada, esta se escogió debido a que las necesidades y los requerimientos del proyecto estaban muy bien definidos. Esta metodología consta de 4 fases específicas análisis de requerimientos, diseño, implementación o desarrollo, pruebas e instalación.

Para la fase de análisis se definieron los requerimientos con los cuales tendría que cumplir el proyecto y que estos pudieran servir como un plano a seguir en la construcción del proyecto. Los requerimientos se definieron conjunto a las partes interesadas en diferentes reuniones para determinar cuáles eran sus verdaderas necesidades y también se aprendieron de los errores que se cometieron en el pasado. En el diseño del proyecto se diseñaron cada uno de los componentes que conformarían el proyecto para así poder facilitar el desarrollo del mismo. Durante la codificación del desarrollo del proyecto se documentó debidamente el código para su fácil entendimiento y descripción de las acciones realizadas dentro del mismo.

Las pruebas del proyecto se dividieron en tres partes, las pruebas unitarias que garantizan que el desarrollo de que cada uno de los componentes del proyecto funcionen como se diseñaron, también se realizaron las pruebas de aceptación de usuario, las cuales se encargaron de verificar que se cumplieran los objetivos del proyecto desde el punto de vista del negocio y así poder darle el visto bueno al proyecto y por último las pruebas de estrés realizadas a los servicios web para garantizar que estaban capacitados para soportar un alto número de transacciones. Las pruebas se realizaron en un ambiente controlado o de desarrollo para no afectar las operaciones diarias, esto es porque se pueden presentar malos funcionamientos durante las pruebas que pueden afectar los ambientes de producción.

A. Herramientas

A continuación se listan algunas herramientas que se utilizaron para el diseño, implementación y pruebas del proyecto:

- Ya que se cuenta con los servidores Windows 2008 R2 y con las licencias de Microsoft SQL Server 2008 R2 y Microsoft Visual Studio 2010, se decidió utilizar las herramientas de Microsoft para el desarrollo del proyecto.

- El lenguaje de programación se escogió C#.
- Las pruebas unitarias se realizaron directamente en Visual Studio.
- Para realizar los diagramas se utilizó la herramienta web Gliffy.
- Para realizar las pruebas de estrés de los servicios web se utilizó la aplicación App Load Perfect.
- Para el manejo de las bitácoras de la aplicación se utilizó el framework NLog.

V. Análisis

A. Análisis del negocio

La empresa hoy en día ofrece diferentes productos electrónicos, lo que ha llevado a que se realicen diferentes desarrollos para el despacho de los mismos, como también para poder ofrecerlos a los clientes para poder incrementar sus ventas. Este modelo de negocio ha tenido un crecimiento considerable en los últimos años por lo que es importante que los sistemas que se encargan de realizar las ventas de este negocio funcionen adecuadamente. Actualmente las ventas de los productos se realizan a través de servicios web en los cuales se ofrece un catálogo de diferentes productos.

Al ir incrementando poco a poco la cantidad de clientes que se conectan o hacen uso de los servicios web es cada vez más complicado para la empresa llevar el control de las ventas como también de la configuración de los productos y clientes con el sistema actual, ya que algunos productos no pueden ser configurados por el usuario final si no que únicamente por algún técnico de la empresa, lo que ocasiona un cuello de botella en donde solo esta persona puede o sabe cómo realizar las configuraciones, por último muchos de estos procedimientos de cómo realizar estas configuraciones no están documentados.

La forma en que se despachan los productos actualmente es que las solicitudes se encolan y son procesadas posteriormente, lo que ocasiona que muchas veces se acepten transacciones que posiblemente no puedan ser procesadas o despachadas con el distribuidor. Esto ha ocasionado muchos descontentos con los clientes y al área comercial de la empresa les resulta complicado resolver todos estos casos.

Son claras las necesidades que tiene la empresa para poder gestionar la configuración de sus productos de una forma fácil y rápida sin tener que depender de alguna persona en específico para poder realizarlo, ya que esto ocasiona cuellos de botella. También la forma en que son procesadas las solicitudes para su despacho ha causado inconformidad de parte de los clientes por lo que se espera que los próximos productos que se incorporen su despacho sea de una forma diferente y resuelva esta problemática y luego cambiar la forma en que se despachan los productos que se ofrecen en la actualidad.

El análisis se realizará con los encargados de la venta de productos electrónicos para así poder entender y determinar las necesidades y requerimientos

B. Análisis de requerimientos

Una de las fases de la metodología de desarrollo de cascada es el análisis de requerimientos de las partes interesadas, los cuales se obtuvieron mediante diferentes reuniones en las cuales se discutieron las necesidades actuales, como también errores que se cometieron en el pasado y que no se quisieran incluir en este desarrollo. Primero se obtuvieron los requerimientos para el configurador de producto y luego se obtuvieron los mismos para el despacho del nuevo producto.

Los requerimientos que se obtuvieron para el configurador de productos después de realizar diferentes reuniones fueron los siguientes:

1. Productos

- Se deben poder activar o desactivar en su totalidad, esto es debido a malos funcionamientos de parte del distribuidor se pueden desactivar.
- Cada producto debe tener las siguientes propiedades:
 - Un nombre.
 - Un país.
 - Una moneda.
 - Un distribuidor.
 - Un tipo. Los cuales pueden ser:
 - Denominación.
 - Rango.
 - Un esquema de ventas. Los cuales pueden ser:
 - Comisión.
 - Descuento.
 - Sin cuenta.
 - Un horario de servicio, por ejemplo para poder configurar que las solicitudes de este producto solo sean procesadas en determinadas horas del día.
 - Puede tener uno o más calendarios de mantenimiento, ya que debido a que el distribuidor puede realizar mantenimientos programados estos también se puedan programar en el configurador y no procesar las solicitudes de este producto en ese lapso de tiempo.
 - Puede tener una o más configuraciones de despacho, una de ellas deberá ser la predeterminada.

2. Cliente

- Un cliente es la empresa a la que se le presta el servicio o vende el producto.
- Un cliente debe o puede tener las siguientes propiedades:
 - Un nombre.
 - Crédito o no crédito de su cuenta asignada.
- A cada cliente se le puede asignar uno a más productos.
 - Al asignarle un producto se le puede asignar una configuración de despacho específica para el cliente o la configuración predeterminada del producto.
- A cada producto que tenga asignado se le puede asignar un descuento o comisión, siempre y cuando el producto aplique.
- A cada producto que tenga asignado se le pueden asignar diferentes denominaciones o rangos del producto.
- Al tener un producto asignado de esquema de ventas de descuento o comisión las ventas del mismo se deberán reflejar en su cuenta.
- A cada cliente se le puede asignar una o más cuentas, únicamente se puede asignar una cuenta por moneda y por cada cuenta que sea asignada se tiene un límite de crédito en el caso que el cliente tenga crédito.

3. Cuentas

- La cuenta debe tener las siguientes propiedades:
 - Un nombre.
 - Una moneda.
 - Un saldo.
 - Un historial de movimientos, los cuales deben de tener
 - Tipo de movimiento los cuales pueden ser
 - Crédito
 - Debito
 - Que cliente realizó el movimiento.
 - Producto de la transacción.
 - Fecha de la transacción.
 - Valor de la transacción.
 - Saldo anterior a la transacción.
 - Saldo actual.
 - Comentario.

- Una cuenta puede ser asignada a varios clientes.

Al finalizar la toma de los requerimientos del configurador de productos se obtuvieron los requerimientos para el despacho del nuevo producto, los cuales se obtuvieron de igual forma mediante reuniones con las partes interesadas. los requerimientos que se obtuvieron son los siguientes:

Uno de los principales requerimientos para el despacho de este producto fue que las solicitudes deberían de ser procesadas en línea o en el momento y ya no utilizar el modelo que se utilizaba anteriormente en donde se encolaban las transacciones y eran procesadas posteriormente ya que esto ocasionaba muchos descontentos con los clientes debido a los tiempos de entrega altos o en algunos casos no era entregado y estos deberían de ser resueltos para poder hacer la entrega del producto ya que ya había realizado un cobro por el mismo. Con esto también se pretende que a futuro todo el despacho de productos se realice de esta misma manera.

- El despacho del producto debe seguir el protocolo de comunicación del distribuidor.
 - La comunicación se deberá realizar vía VPN.
 - Las solicitudes serán XML.
 - Las solicitudes hacia el distribuidor se realizarán a través de un servicio web publicado, los métodos de este servicio web son los siguientes:
 - compra
 - consultaAfiliacion
 - consultaSolicitud
 - cancelacion
 - consultaSaldo
 - consultaTrxDiarias
 - Se deberá de crear un servicio web para la comunicación distribuidor – empresa, los métodos de este servicio deberán ser:
 - confirmacionEntrega
 - receptorAfiliado
- Este producto deberá ser configurado en el configurador de producto.
 - El producto es de esquema de ventas es sin cuenta, por lo cual a los clientes que se le asigne este producto no deberán de tener una cuenta para poder vender este producto.

- El tipo de producto es de rangos, por lo que se definirán los rangos permitidos para este producto.
- El producto se venderá a los clientes a través de un servicio web.
 - La comunicación también se realizará a través de una VPN.
 - Las solicitudes serán XML.
 - Los métodos o funciones que deberá tener este servicio web son:
 - compra
 - cancelaSolicitud
- Se deberá almacenar la información de las solicitudes en la base de datos para propósitos de reportería.
- Se deberá crear una bitácora para registrar las solicitudes, como también excepciones que pueden ocurrir durante el despacho del producto.
- Las solicitudes de este producto deberán manejar diferentes estados para poder determinar el resultado de la operación.
- Se deberá consultar diariamente el saldo de la cuenta del distribuidor como también las transacciones realizadas el día anterior.
- Al recibir la confirmación que el receptor se ha afiliado se deberá realizar una comparación de los apellidos del receptor en el que uno de los apellidos debe de coincidir para determinar que es a esa persona que se le envió la compra.

VI. Diseño

A. Diseño del proyecto

El diseño de la arquitectura proyecto fue importante antes de iniciar el desarrollo del mismo para determinar cuáles serían los principales componentes a desarrollar y a utilizar y como se relacionan entre ellos. Los componentes que conformaran el proyecto son los siguientes:

1. Bases de datos

- Base de datos transaccional del nuevo producto.
- Base de datos para almacenar la información del configurador del producto.

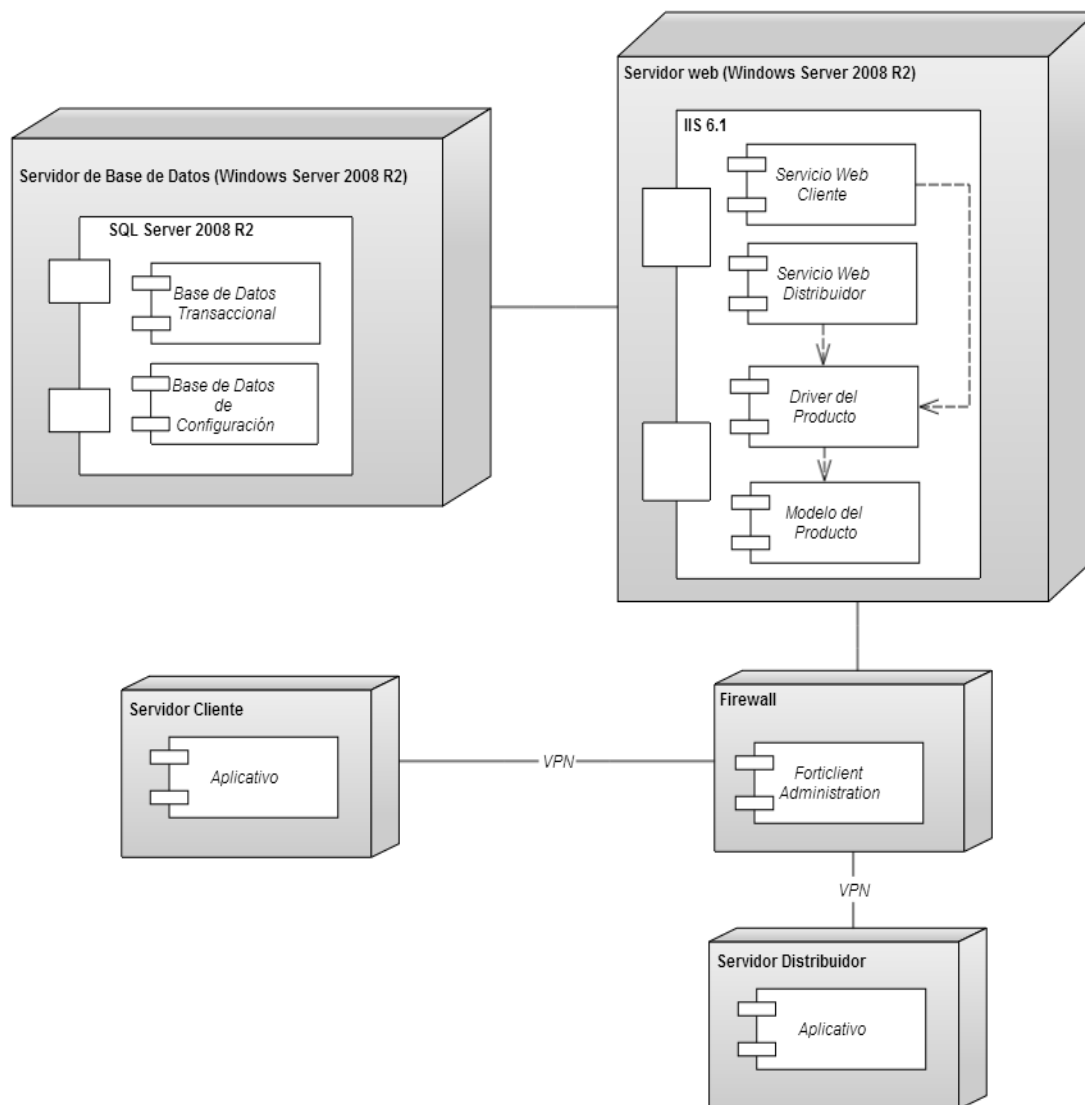
2. Servicios web. Se diseñaron dos servicios web, uno de los cuales estará encargado de atender las peticiones del cliente y otro para manejar las solicitudes que se reciben de parte del distribuidor.

3. Driver del nuevo producto. El driver del nuevo producto se encarga de modelar toda la lógica del negocio, es decir la conectividad de la base de datos, escritura de bitácoras, validaciones de solicitudes, envío de solicitudes, entre otras.

4. Modelo del nuevo producto. El modelo del nuevo producto se encarga de modelar las solicitudes XML que debe de enviar el driver hacia el servicio web del distribuidor.

5. Firewall y VPN. Estos componentes de la arquitectura del proyecto no fueron configurados ya que estos se encuentran fuera del ámbito del proyecto, pero se incluyeron debido a que conforman una parte importante de la comunicación.

Figura No. 4. Diagrama de despliegue del proyecto



B. Bases de datos

Las bases de datos a utilizar se realizaron mediante de diagramas entidad relación basándose en los requerimientos del proyecto. El manejador de base de datos que se utilizó fue SQL Server 2008 R2, ya que este es un manejador robusto y cuenta con compatibilidad con los principales lenguajes de programación como C#, PHP y Java, lo que permite que sin importar el lenguaje del aplicativo las bases de datos diseñadas podrán ser accedidas debidamente, por último como se había mencionado previamente se cuenta con su licencia para poder utilizarlo.

Para el diseño de la base de datos se consideró una forma de nombramiento para las entidades y sus atributos el cual consiste de las siguientes reglas:

- Todas las entidades se nombran el plural.
- Todos los atributos se nombran singular.
- Las llaves primarias se identifican de la siguiente manera `idNombreEntidad` o `codigoNombreEntidad`.
- Para el nombramiento de ya sea una entidad o atributo, se utiliza la primer letra de la primera palabra en minúscula y la segunda en mayúscula y así sucesivamente. Ejemplo: `productosCliente`.
- Los atributos que son llaves foráneas deben de llamarse de igual forma que al atributo que hacen referencia.

Al mantener un nombramiento uniforme de las entidades y atributos de las base de datos se logra que el desarrollo de las consultas a la base de datos sean más fáciles de realizar, y a la vez que las mismas sean más entendibles para el desarrollador.

Se decidió que todas las interacciones tanto de lectura como de escritura se realizaran a la base de datos debería de ser por medio de procedimientos almacenados, ya que nos ofrecen las siguientes ventajas:

- Se evita tener código SQL dentro del código aplicativo el cual puede ocasionar problemas de mantenimiento y problemas si se cambia el diseño de alguna entidad.
- Se busca reutilizar las rutinas SQL.
- Proveen una interfaz entre el aplicativo y la base de datos.
- Permiten se puedan realizar rutinas SQL complejas directamente desde el manejador.
- Reduce el riesgo de sufrir ataques de inyección SQL.

Figura No. 5. Código ejemplo de la utilización de procedimientos almacenados

```

public bool insertaInformacion(int paramA, string paramB) {
    // Se inicializa la conexion a la base de datos
    SqlConnection sqlConnection = new SqlConnection(productSQLConnection);
    bool result = false;
    try {
        int output = 0;
        // Se define el StoredProcedure a ejecutar
        string sqlStatement = "insertaInformacion";
        SqlCommand cmd = new SqlCommand(sqlStatement, sqlConnection);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.Add("@monto", SqlDbType.Int).Value = monto;
        cmd.Parameters.Add("@noTelefono", SqlDbType.VarChar).Value = celular;
        // Se inicia la conexion a la base de datos
        sqlConnection.Open();
        // Se ejecuta el StoredProcedure
        output = (int)cmd.ExecuteScalar();
        // Se termina la conexion a la base de datos
        sqlConnection.Close();
        sqlConnection.Dispose();
        // Se interpreta el resultado
        return (output == 1 ? true : false);
    } catch (Exception ex) {
        // Se verifica que el estado de la conexion
        if (sqlConnection.State == ConnectionState.Open) {
            sqlConnection.Close();
            sqlConnection.Dispose();
        }
        // Se registra la excepcion en la bitacora
        logger.Log(LogLevel.Trace, string.Format("Error al ejecutar el StoredProcedure insertaInformacion: {0}", ex.StackTrace.ToString()));
        return false;
    }
}

```

Para el diseño de este proyecto se diseñaron dos bases de datos las dos cuales cumplen con dos propósitos, una de ellas para almacenar las configuraciones de producto y la otra para almacenar las transacciones o ventas que se realicen del nuevo producto.

Figura No. 6. Diagrama entidad relación del configurador de productos

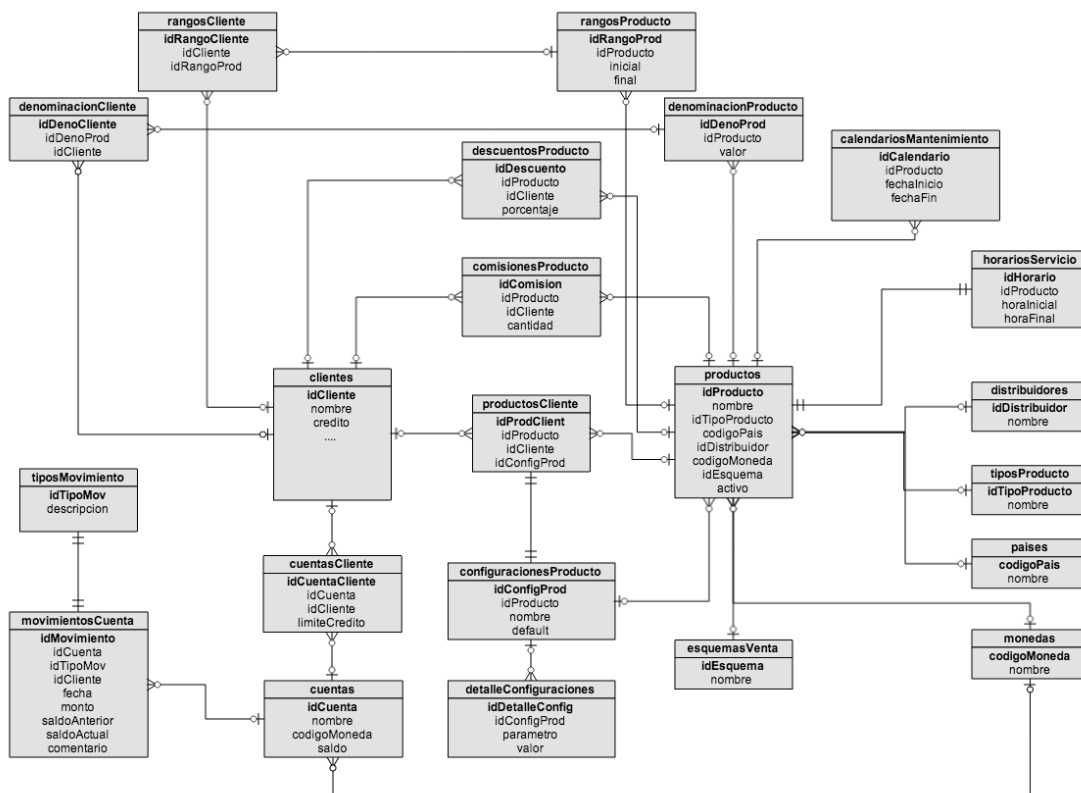
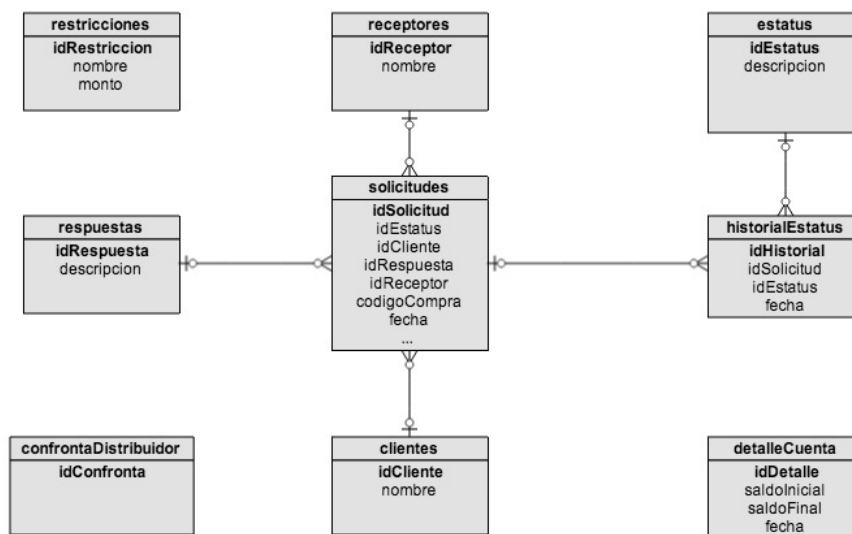


Figura No. 7. Diagrama entidad relación de la base de datos transaccional de producto nuevo



C. Servicios web

1. Servicio web para el cliente. Este servicio web es utilizado por los clientes para la compra y para la cancelación de solicitudes. Este servicio cuenta con dos métodos los cuales hacen uso del driver del producto como del modelo del mismo. Los métodos de este servicio web son:

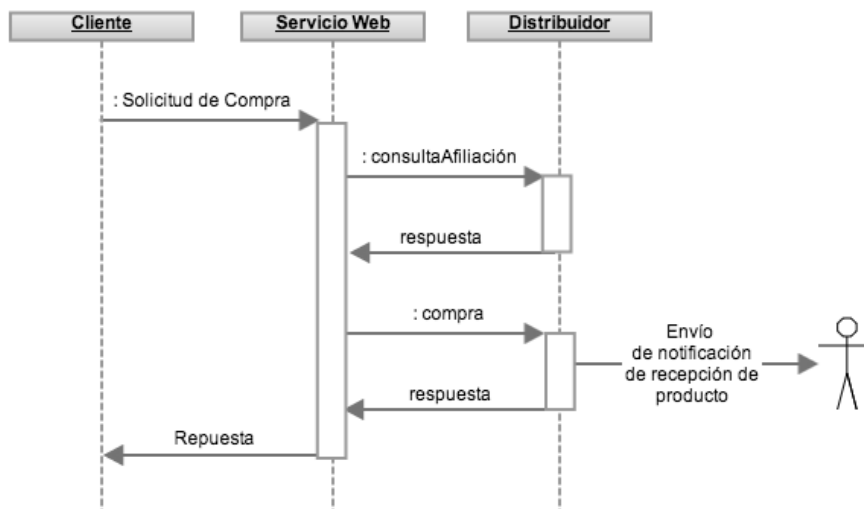
- compra: recibe como parámetros el identificador del cliente, el código de compra (el cual debe de ser único por cliente), el identificador, nombre y apellido del receptor, y el valor de la compra. Este método se encarga de enviar las solicitudes a despachar.
- cancelación: recibe como parámetros el identificador del cliente y el código de compra. Este método se encarga de enviar a cancelar una solicitud determinada.

Al recibir una nueva solicitud se realizan diferentes validaciones para que esta pueda ser procesada, las cuales son:

- El cliente debe de tener el producto asignado y este debe de estar activo.
- El valor de la compra debe de estar entre los rangos permitidos del cliente.
- El código de compra no debe de existir previamente en la base de datos.
- Si la solicitud puede ser procesada debido al horario de servicio o si existe algún mantenimiento programado.
- Si los campos requeridos fueron recibidos.
- En el caso que el receptor ya este afiliado con el distribuidor se compara que uno de los apellidos coincida con los que están registrados con el distribuidor.

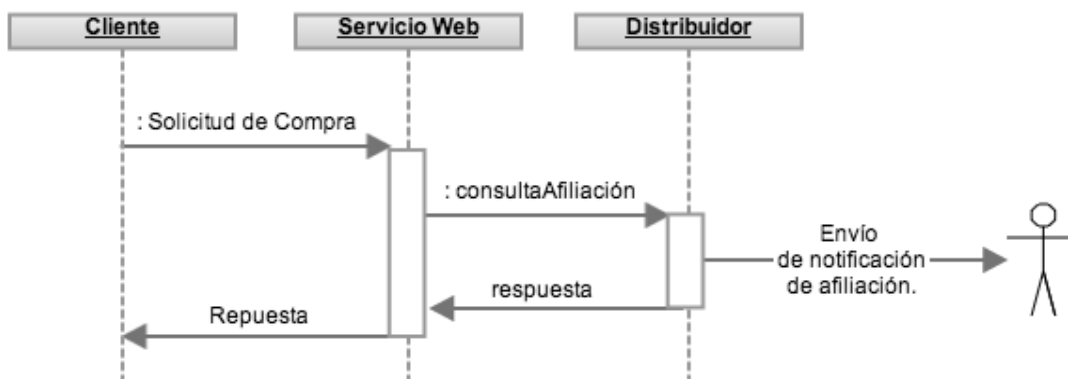
Al realizar las validaciones se consulta si el receptor está afiliado al distribuidor para que pueda recibir su producto en el caso que lo este, se envía la solicitud de compra al distribuidor para que sea procesada, el distribuidor notifica al receptor en el caso que se haya procesado exitosamente, el distribuidor notifica al receptor que ha recibido el producto y se devuelve la respuesta a cliente. Para esta tarea se sigue el siguiente diagrama de secuencia.

Figura No. 8. Diagrama de secuencia de compra



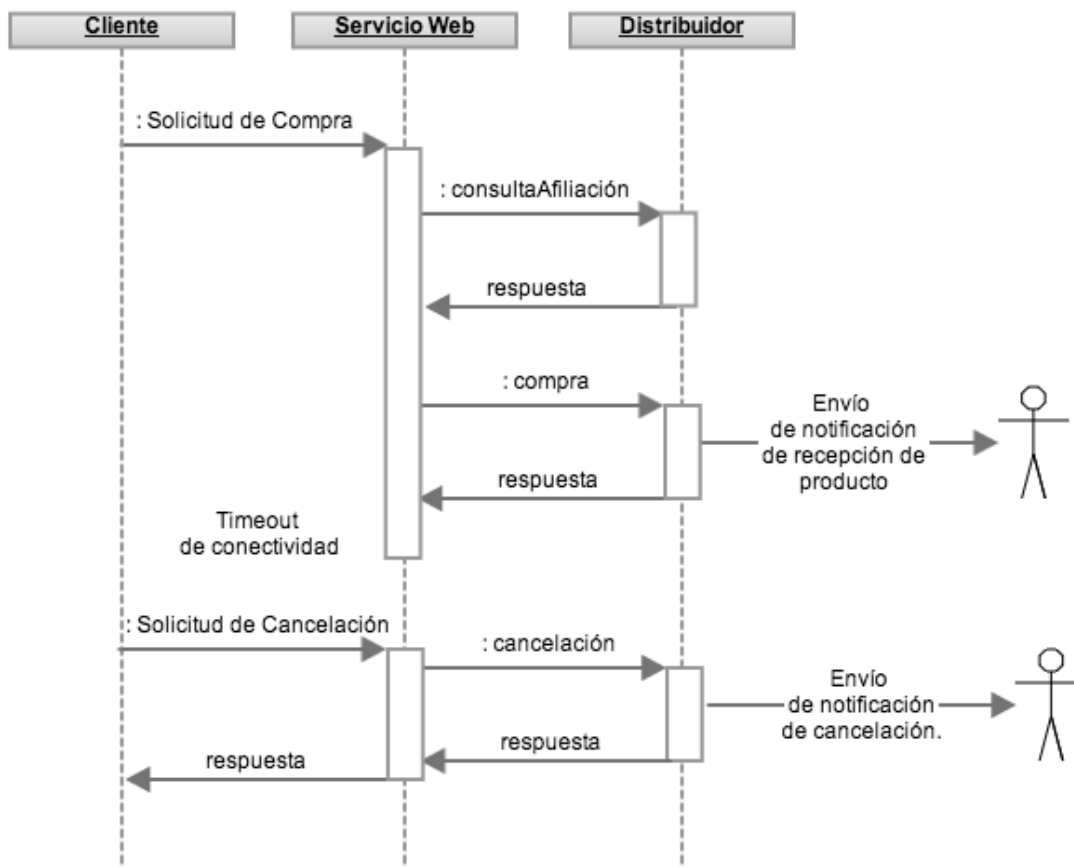
En el caso que el receptor no este afiliado con el distribuidor se realizan las mismas validaciones que en el caso anterior, con la diferencia que en esta caso el cliente es notificado que se afilie con el distribuidor. Para este caso se sigue el siguiente diagrama de secuencia

Figura No. 9. Diagrama de secuencia de compra, receptor no afiliado.



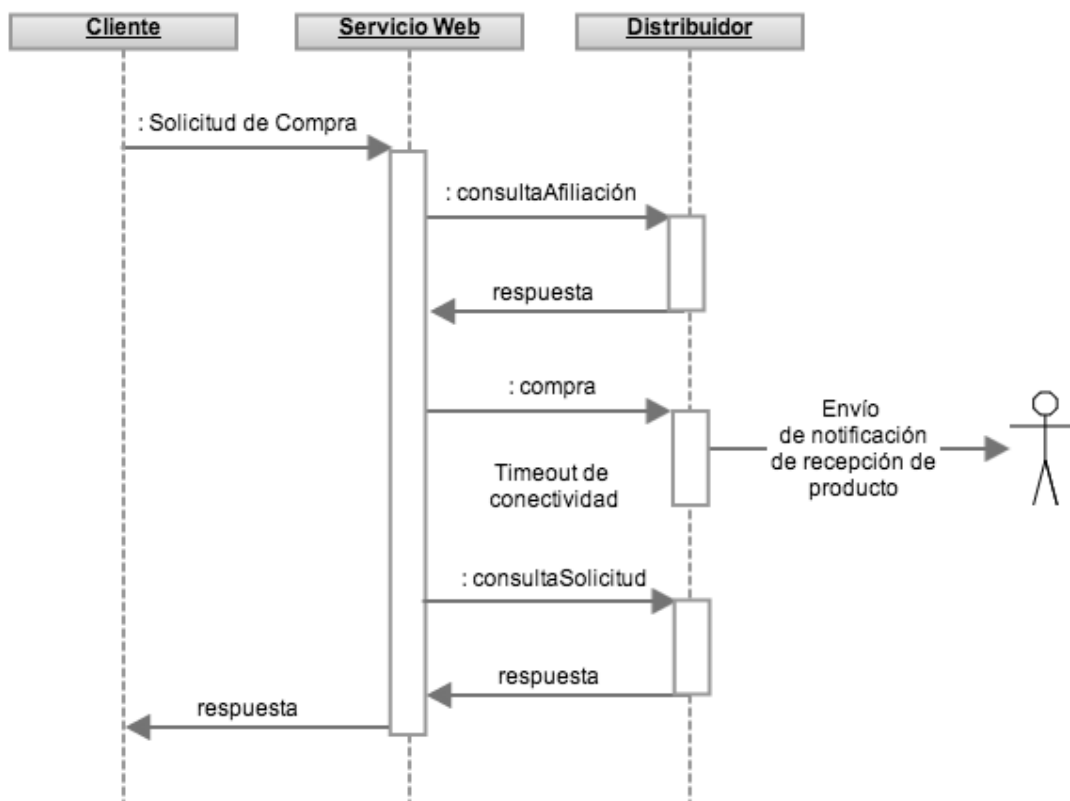
Debido a que pueden existir pérdidas de conexión al momento de que el cliente realice una compra se definió un procedimiento para mitigar este tipo de error. El cual consiste de que si el cliente experimenta una pérdida de conexión hacia el servicio web este deberá invocar el método de cancelación, para realizar esto también se envía una solicitud de cancelación al distribuidor. Para esta tarea se sigue el siguiente diagrama de secuencia.

Figura No. 10. Diagrama de secuencia de compra, pérdida de conexión entre cliente y el servicio web.



Debido a que se puede experimentar una pérdida de conectividad entre el cliente y servicio web, esto también puede suceder entre el driver del producto y el servicio web del distribuidor, para poder mitigar de que la compra si se efectuó pero no se recibió la respuesta del servicio, se decidió que en el caso que se obtenga una pérdida de conexión se deberá consultar el estado de la compra para determinar si se procesó exitosamente. Para esta tarea se sigue el siguiente diagrama de secuencia.

Figura No. 11. Diagrama de secuencia de compra, pérdida de conexión entre el servicio web y el distribuidor



2. Servicio web para el distribuidor. Este servicio web es utilizado por el distribuidor para dos funcionalidades específicas, una de ellas es la confirmación de entrega del producto y otra de ellas es la confirmación de afiliación de un determinado receptor. Los métodos de este servicio web son:

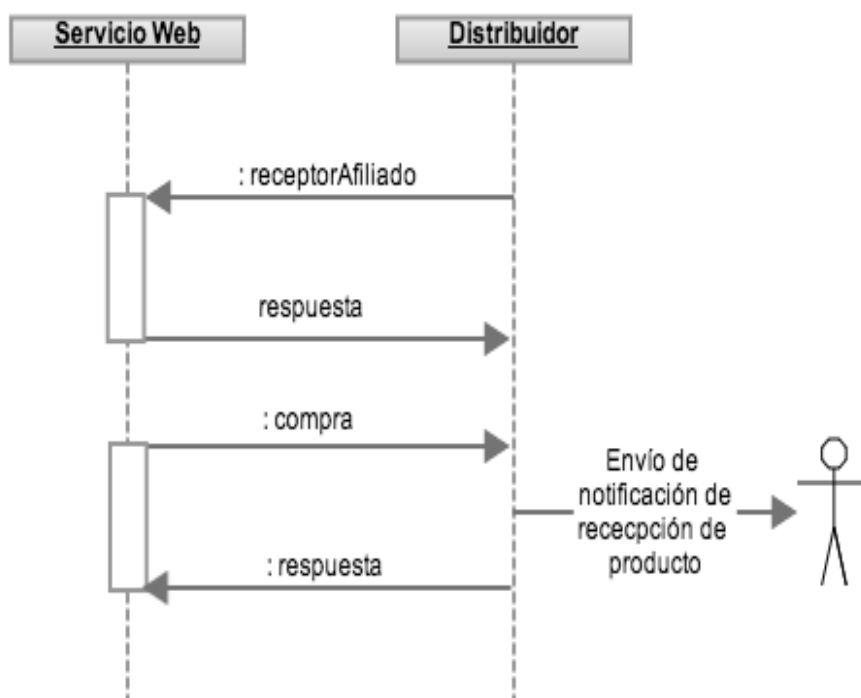
- confirmación entrega: recibe como parámetro el código de compra con la referencia del distribuidor indicando que el producto ya fue entregado al receptor.
- receptor afiliado: recibe como parámetro el identificador del cliente indicando que el receptor ya ha sido afiliado.

En el momento en que el receptor se afilia con el distribuidor este automáticamente envía una solicitud al servicio web indicando que este ya se encuentra afiliado, para esto se valida la estructura de la solicitud XML, como también de que el receptor efectivamente exista en el sistema, que este está esperando la confirmación de afiliación y por último se realiza una

comparación de apellidos del receptor para determinar que si es quien le enviaron la compra.

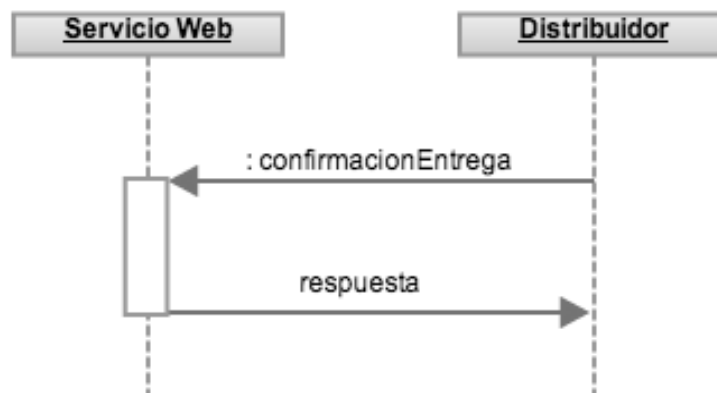
Al recibir la confirmación de que el receptor ya ha sido afiliado exitosamente se procede a enviar la solicitud o solicitudes que estaban pendientes de enviar debido a que el receptor no estaba afiliado. Tal y como se manejó para el caso en que el receptor ya estaba afiliado, si se obtiene una pérdida de conectividad al momento de realizar la compra con el distribuidor, se realiza una consulta de la compra para poder determinar el estado actual de esta solicitud. Para esta tarea se sigue el siguiente diagrama de secuencia.

Figura No. 12- Diagrama de secuencia de afiliación de receptor de producto.



El distribuidor debe de notificar en el momento en que el producto ya fue entregado al cliente y esto se hace mediante una solicitud al servicio web, las validaciones que se realizan es que la estructura de la solicitud XML sea valida como también de que el código de compra exista. Para esta tarea se sigue el siguiente diagrama de secuencia.

Figura No. 13. Diagrama de secuencia de confirmación de entrega del producto.



3. Códigos de respuesta de los servicios. Los servicios web se diseñaron de tal forma que devuelvan un código de respuesta para poder indicar el resultado de la operación y este pueda ser interpretado por el sistema que hace uso del servicio web.

Tabla No. 1. Códigos de respuesta del servicio web.

Código de respuesta	Descripción
1000	Éxito
1001	Fuera de servicio o en mantenimiento
1002	Error general en el sistema
1003	La estructura de la solicitud no es valida
1004	Identificador del receptor no existe
1005	Identificador del cliente no existe
1006	Código de compra ya existe
1007	Código de compra no existe
1008	El valor de la compra no se encuentra dentro de los rangos permitidos
1009	La compra ya ha sido confirmada
1010	El cliente no tiene asignado el producto
1011	La compra ya ha sido cancelada previamente
1012	No se han recibido los campos obligatorios
1013	Los apellidos del receptor no coinciden

El servicio web del distribuidor también maneja un listado de códigos de respuesta que nos permite determinar que fue lo que sucedió con una determinada compra e interpretar el resultado de la operación.

D. Driver y modelo del producto nuevo

Estos dos elementos del proyecto son fundamentales para el despacho del nuevo producto ya que ambos cumplen con diferentes funcionalidades específicas. El *driver* del producto se encarga de modelar toda la lógica del negocio, interacciones con la base de datos a través de los procedimientos almacenados, registro en las bitácoras, como también hace uso del modelo.

El modelo es el encargado de modelar las solicitudes que se envían al servicio web del distribuidor como también de las solicitudes que se reciben de parte del distribuidor. Estas solicitudes son enviadas en formato XML, para hacer el modelado de estas solicitudes se definieron clases que manejan las solicitudes y respuestas de cada una de las acciones y otra para poder realizar el envío de cada una de las solicitudes. Lo que indica que el driver hace uso del modelo para la construcción y envío de solicitudes al servicio web del distribuidor, las solicitudes se modelaron como clases para así poder manipular de mejor forma la información que se envía o se recibe. Para poder realizar la construcción de las solicitudes XML se hizo uso de la librería de XML de C#.

Figura No. 14. Uso de la librería de C# para la construcción de XML

```
public string buildXML() {
    // Se crea el documento XML
    XmlDocument request = new XmlDocument();

    // Se crea el elemento root del XML en este caso es "ROOT"
    XmlElement root = request.CreateElement("ROOT");
    request.AppendChild(root);

    // Se crea el elemento XML de node_a
    XmlElement node_a = request.CreateElement("NODE_A");
    node_a.InnerText = NODE_A;

    // Se crea el elemento XML de node_b
    XmlElement node_b = request.CreateElement("NODE_B");
    node_b.InnerText = NODE_B;

    // Se crea el elemento XML de node_c
    XmlElement node_c = request.CreateElement("NODE_C");
    node_c.InnerText = NODE_C;

    // Se agregan los nodos hijo al elemento root del XML
    root.AppendChild(node_a);
    root.AppendChild(node_b);
    root.AppendChild(node_c);

    // Se devuelve el XML
    return request.OuterXml;
}
```

E. Estados de una solicitud

Cada una de las solicitudes de compra del nuevo producto que provienen del cliente tienen un estado el cual permite determinar que ocurrió con esta solicitud. Los diferentes estados que puede tener una solicitud son:

Tabla No. 2. Estados de una solicitud del nuevo producto.

Estado	Descripción
1	Nueva
2	Pendiente de Afiliación
3	Éxito
4	Entregada
5	Cancelada
6	Error

Todas las solicitudes que ingresan al sistema mediante el servicio web ingresan con un estado de *Nueva* que significa que acaba de ingresar al sistema y va a ser procesada, luego la solicitud puede cambiar a tres estados *Pendiente de Registro*, *Éxito* o *Error*.

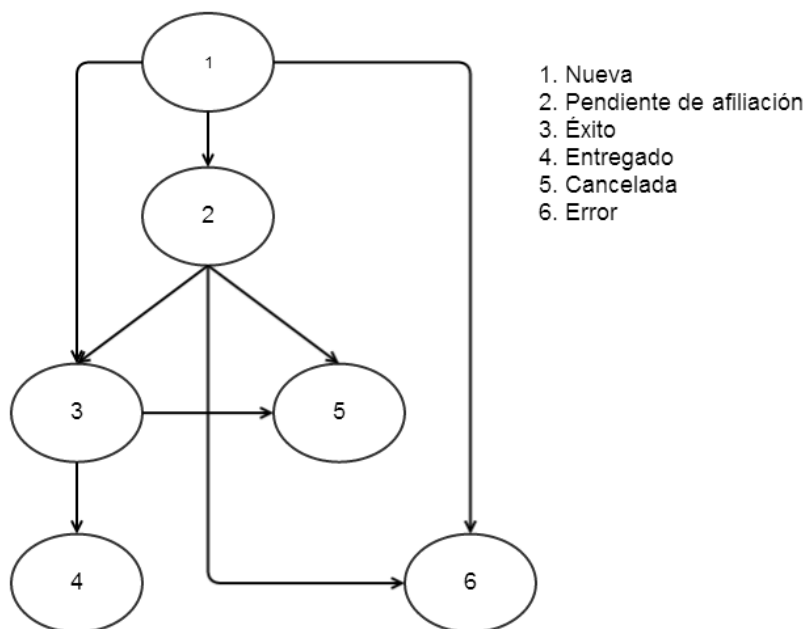
Si la solicitud pasa de *Nueva* a *Éxito* se interpreta como que la compra ha sido procesada exitosamente por el distribuidor y el receptor ha sido notificado para que se le pueda ser entregado el producto. En el caso que la solicitud pase de *Nueva* a *Error* se interpreta como que la solicitud no ha podido ser procesada exitosamente con el distribuidor. En el caso que la solicitud pase de *Nueva* a *Pendiente de Registro* se interpreta como que el receptor aún no está afiliado con el distribuidor y esta solicitud espera pendiente hasta que se reciba la confirmación de afiliación del distribuidor.

Una solicitud que está en estado de pendiente de afiliación puede pasar de este estado a *Éxito*, lo cual indicaría que al momento que el receptor se afilió la solicitud fue enviada y procesada exitosamente con el distribuidor y el receptor ya fue notificado y se le puede hacer entrega del producto, también puede pasar a un estado de *Cancelada*, lo cual indicaría que esta ya no sería enviada al distribuidor en el momento que se reciba la confirmación de afiliación ya que esta fue cancelada y por último la solicitud puede pasar de *Pendiente de Afiliación* a *Error* en el caso que al ser procesada la solicitud ocurra un error durante este procedimiento.

Una solicitud que está en estado de éxito puede pasar únicamente a dos estados a cancelada y a entregado, en el caso de cancelada indicaría que el cliente quiso cancelar esta

solicitud y para el caso de entregado indicaría que el producto ya fue entregado al receptor, por lo que en este último caso ya no se podría cancelar una solicitud. El siguiente diagrama representa los posibles cambios de estado que puede tener una solicitud.

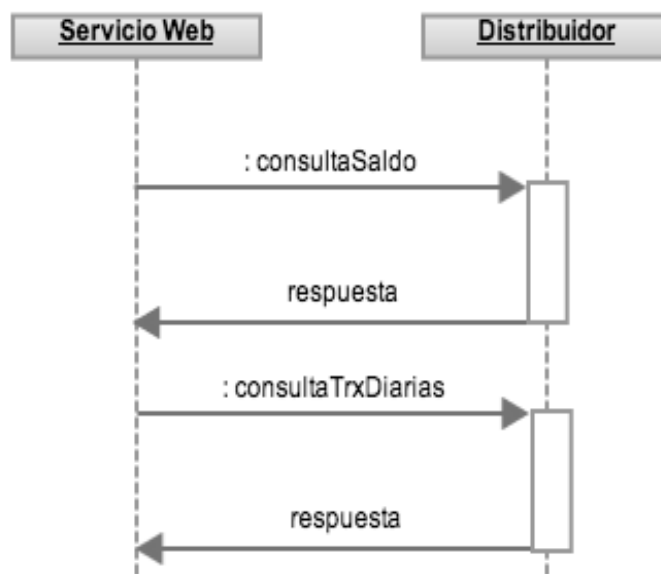
Figura No. 15. Diagrama de estados de solicitudes.



F. Consultas al distribuidor

Uno de los requerimientos del nuevo producto es que se realicen consultas sobre el saldo final del día anterior y de las transacciones realizadas el día anterior para así realizar una comparación entre las ventas que registro el distribuidor y las ventas que realizo la empresa. Para llevar a se realizó diseño como una tarea ejecutable que hace uso del driver del producto para poder enviar las consultas de saldo y de transacciones al servicio web del distribuidor, esta tarea será ejecutada diariamente a las 4:00 am. Para realizar esta tarea se sigue el siguiente diagrama de secuencia.

Figura No. 16. Diagrama de secuencia de consulta de saldo y transacciones diarias



G. Comparación fonética

Uno de los requerimientos para el despacho del nuevo producto es que se debe de realizar una comparación de apellidos en diferentes casos. Por lo que para esto se diseñó un algoritmo que se encarga de comparar dos apellidos, para esto se realizó una investigación sobre comparación de cadenas y se llegó a la conclusión de que la comparación se debería de realizar fonéticamente y no por medio de la longitud de la cadena o que determinados caracteres coincidieran, esto es porque se puede hacer el ingreso de un apellido incorrectamente ya que es algo muy común en el ingreso de datos. Para poder lograr la comparación se utilizaron dos algoritmos los cuales son:

1. **Double Metaphone:** este algoritmo se encarga de devolver una llave a partir de una palabra que representa sus propiedades fonéticas. Una de las ventajas de este algoritmo sobre otros es que puede devolver la llave que representa sus propiedades fonéticas en diferentes idiomas entre uno de ellos el español. A continuación se presentan algunos resultados del algoritmo

Tabla No. 3 Resultados del algoritmo Double Metaphone

Cadena	Resultado
Juan	JN y AN
Maria	MR
Mendoza	MNTS
Ramirez	RMRS
Ramires	RMRS
De la Vega	TLFK
De la Beja	TLPK

2. Distancia de Levenshtein: este algoritmo se encarga de determinar la distancia entre cadenas o la cantidad de caracteres necesaria para transformar una cadena a otra. A continuación se presentan algunos resultados del algoritmo.

Tabla No. 4. Resultados del algoritmo distancia Levenshtein

Cadena A	Cadena B	Resultado
Juan	Guan	1
Juan	Diego	5
TLFK	TLPK	1
Mendosa	Mendoza	1
Luis	Luisao	2

Al tener ambos algoritmos también se definió una función la cual se encargaría de eliminar el espacio que separa a los apellidos compuestos, como por ejemplo Del Bosque la función devuelve DelBosque. Los apellidos que aplican son todos aquellos que tengan Del, De Los, De Las, De La y De. Esto se realiza debido a que puede que el cliente al momento de realizar la compra únicamente envíe uno de los apellidos y el distribuidor tenga registrado los dos y así separar fácilmente los apellidos.

Finalmente se llegó a definir el siguiente algoritmo que hace uso de los tres mencionados anteriormente. El algoritmo definido tiene una tolerancia de distancia de Levenshtein de 1 es decir si el resultado del Double Metaphone no es idéntico se realiza la comparación de distancia de Levenshtein.

Figura No. 17. Algoritmo implementado para la comparación de cadenas

```

public bool compareLastNames(string lastNameA, string lastNameB) {
    // Objeto de la clase LevenshteinDistance que se utilizara para determinar
    // la distancia entre dos palabras. El minimo que se acepta para la comparacion
    // de apellidos es 1.
    LevenshteinDistance levensthein = new LevenshteinDistance();
    // Objeto de la clase DoubleMetaphone que se utilizara para determinar la
    // similitud fonetica de los apellidos
    DoubleMetaphone doubleM = new DoubleMetaphone();

    // Declaracion de variables
    string doubleMetaphoneA = "";
    string doubleMetaphoneB = "";

    // Declaracion de listas, que se construyen a partir de separar cada apellido despues de haber
    // removido los articulos como de, del, de los, de las, de la
    List<string> listA = dropLastNameArticles(lastNameA.Trim().ToLower()).Split(' ').ToList();
    List<string> listB = dropLastNameArticles(lastNameB.Trim().ToLower()).Split(' ').ToList();

    // Ciclo foreach que se encarga de comparar todos los apellidos de la lista A contra los
    // de la lista B, si se obtiene una igualdad/similitud se devuelve true, de lo contrario
    // se finaliza el ciclo y se devuelve false
    foreach (string a in listA) {
        foreach (string b in listB) {
            doubleMetaphoneA = doubleM.GenerateDoubleMetaphone(a);
            doubleMetaphoneB = doubleM.GenerateDoubleMetaphone(b);
            if (doubleMetaphoneA.Equals(doubleMetaphoneB) == true) {
                listB.Remove(b);
                return true;
            } else if (levensthein.Compute(doubleMetaphoneA, doubleMetaphoneB) == 1) {
                listB.Remove(b);
                return true;
            }
        }
    }
    return false;
}

```

Tabla No. 5. Resultados del algoritmo implementado para la comparación fonética

Cadena A	Cadena B	Resultado
Dighero	Diguero	True
Ramirez	Ranires	True
Barragan	Barajan	True
Diaz	De Los Diaz	False

H. Manejo de bitácoras

Para el manejo de bitácoras o de registros de las solicitudes y respuesta de los servicios web como también del modelo y del driver del producto se utilizó es un framework gratis y de código libre para el manejo de bitácoras que ofrece muchas ventajas sobre los algoritmos que manipulan los archivos de texto, como por ejemplo:

- Maneja la concurrencia de escritura de los archivos, lo que quiere decir que si se utiliza NLog se podrá registrar en la bitácora sin importar que tan seguidos son los sucesos.
- No se necesita de un desarrollo extra para utilizarlo más que la inclusión de la librería en el proyecto.

Debido a que el proyecto cuenta con una base de datos transaccional se corre el riesgo que esta base de datos se pueda corromper o se pueda perder la información y esto es un problema que ha sucedido, por lo cual se pretende mitigar registrando en estas bitácoras todas las peticiones realizadas a los servicios web como también cualquier excepción que ocurra en el flujo de las solicitudes, para que si existe la necesidad de reconstruir la información se pueda realizar por medio de la bitácora como también determinar específicamente que fue lo ocasionó una excepción.

Para poder hacer uso de NLog se debe de incluir la librería y luego crear un archivo de configuración, en este archivo de configuración se puede configurar la plantilla que va a tener la bitácora como también, el directorio en donde se van a almacenar las bitácoras. A continuación se presenta un ejemplo básico de un archivo de configuración de NLog.

Figura No. 18. Ejemplo de archivo de configuración NLog

```
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      throwExceptions="true" internalLogFile ="C:\Logs\nlog.log">
  <variable name="baseDir" value="C:\Logs\${shortdate}\" />
  <targets>
    <target name="traceFile" xsi:type="AsyncWrapper" queueLimit="5000" overflowAction="Discard">
      <target name="traceFile" xsi:type="File"
              layout="${longdate}|${processname}|${message}"
              keepFileOpen="false"
              fileName="${baseDir}Trace.log"/>
    </target >
  </targets>
  <rules>
    <logger name="Trace" minlevel="Trace" writeTo="traceFile" final="true" />
  </rules>
</nlog>
```

Al haber realizado el archivo de configuración se puede hacer uso del mismo declarando el objeto de NLog y luego utilizando la función Log para realizar el registro en la bitácora

Figura No. 19. Ejemplo de uso de NLog

```
using NLog;

.....

Logger traceLogger = NLog.LogManager.GetLogger("Trace");

.....

traceLogger.Log(LogLevel.Trace, string.Format("Error al ejecutar la funcion x: {0}", exception.ToString()));
```

I. Interfaz gráfica del configurador de productos

A pesar de que el desarrollo de la interfaz gráfica del configurador de producto estuviera fuera del ámbito del proyecto se realizó el diseño de las diferentes pantallas del sistema, esto se realizó a través de prototipos de papel o paper prototyping, los cuales fueron de utilidad ya que se pudo diseñar y modificar varias veces el prototipo para poder llegar a la versión final de cada una de las pantallas y así lograr satisfacer las necesidades de las partes interesadas del proyecto.

Aparte de que se desarrollaron los prototipos de papel también se propuso utilizar para la interfaz gráfica el Yii framework, este se propuso a su reciente crecimiento y a la rápida curva de aprendizaje que uno puede llegar a tener al experimentar con el mismo, otro factor importante es que tiene una documentación bastante amplia y la comunidad en internet es bastante grande lo cual facilita la resolución de dudas, por ultimo también utiliza el patrón de arquitectura modelo vista controlador el cual da un orden al desarrollo del producto el cual separa los datos, la lógica del negocio y la vista o interfaz de usuario.

VII. Discusión

El uso de la metodología de cascada se pudo aplicar muy bien al proyecto ya que las partes interesadas del proyecto conocían sus necesidades por lo que el camino a tomar estaba claro, aunque una buena toma de requerimientos y buen análisis fue el que permitió que se pudiera desarrollar y concluir el proyecto. Durante el desarrollo del proyecto se aplicaron diferentes conocimientos aprendidos durante la carrera, estos ayudaron a que se pudiera desarrollar de una forma.

El desarrollo de este proyecto proporciona las bases para que se puedan desarrollar nuevos proyectos como también logra resolver muchas de sus problemáticas que tienen previos al desarrollo de este producto, como también el desarrollo del despacho del nuevo producto permite ampliar su catálogo de productos.

Al desarrollar el despacho del nuevo producto de una forma en que las transacciones son procesadas en línea se deja atrás la forma en la que se despachaban las solicitudes en la empresa, que funcionaba de tal forma en las que estas se encolaban, lo cual mejora los tiempos de entrega como también el servicio al cliente algo que la empresa busca mejorar. También se reduce el riesgo que existía que al cliente que hace uso del servicio web se la haya dado una respuesta y luego esta solicitud no necesariamente pudo haber sido procesada con el distribuidor por alguna u otra razón de rechazo. Al cambiar este esquema de despacho se busca que todos los nuevos productos se despachen de esta forma como también los antiguos se migren a esta nueva forma de despacho.

El configurador de productos va a poder empoderar a los usuarios finales para poder realizar las configuraciones o los ajustes que ellos crean necesarios para sus clientes o productos, sin que este sea un proceso tardado y complicado, ya que se diseñó de una forma en la que se cumplan todas las necesidades de la empresa.

VIII. Pruebas e instalación

A. Pruebas

Se realizaron diferentes pruebas durante el desarrollo del proyecto que ayudaron a determinar que el funcionamiento del mismo era el correcto. Es importante resaltar que todas las pruebas se realizaron en un ambiente de desarrollo, es decir que este es un ambiente controlado y dedicado para realizar pruebas por lo que en el caso de cualquier inconveniente o mal funcionamiento no se afecte las operaciones de los ambientes de producción, principalmente se buscó no afectar el funcionamiento de los servidores web y de los servidores de bases de datos ya que la empresa operaba mientras se realizaban estas pruebas. El tipo de pruebas que se realizaron son:

1. **Pruebas unitarias.** Las pruebas unitarias permiten garantizar o asegurar que diferentes métodos o funciones de código funcionen tal y como se esperaba en diferentes casos. Para realizar estas pruebas se utilizaron las herramientas de Visual Studio 2010 para realizar las mismas.

Con estas pruebas se busca garantizar que las interacciones con la base de datos fueran realizadas exitosamente tanto lectura como escritura, las validaciones de las solicitudes con diferentes parámetros y por último el envío de solicitudes.

Al finalizar estas pruebas fue sencilla la integración y el uso de los métodos o funciones ya que cada uno estaba probado y estaba garantizado que su funcionamiento fuera el correcto, como también ayudó a que las pruebas de aceptación de usuario no tuvieran problemas de ejecución.

2. **Pruebas de aceptación de usuario.** Las pruebas de aceptación de usuario se realizaron con el distribuidor como con los clientes que van a hacer uso del servicio web que se encarga de despachar el nuevo producto. Estas pruebas están orientadas a verificar las funcionalidades que le interesan al negocio o al usuario final. Para definir las pruebas de aceptación se utilizó la plantilla presentada a continuación.

Tabla No. 6. Plantilla utilizada para las pruebas de aceptación de usuario

Nombre del proyecto								
Fecha de inicio								
Fecha de fin								
ID caso de prueba	Equipo de pruebas	Descripción	Precondiciones o información	Pasos a ejecutar	Resultados esperados	Resultado (Satisfactorio O Error)	Fecha	Observaciones
a.	b.	c.	d.	e.	f.	g.	h.	i.

El objetivo de esta plantilla es que las partes interesadas propongan sus casos de prueba de los que están interesados en realizar. Las propiedades de la plantilla se describen a continuación.

- a. ID caso de prueba: se utiliza para manejar un identificador único del caso de prueba para que la comunicación acerca sobre este caso sea más sencilla.
- b. Equipo de pruebas: el equipo de pruebas que estará iniciando o realizando el caso de prueba.
- c. Descripción: descripción de lo que pretende el caso de pruebas.
- d. Precondiciones: son las condiciones o información con las que se deben de contar antes de realizar la prueba.
- e. Pasos a ejecutar: es la secuencia de pasos que se deben de realizar.
- f. Resultados esperados: el resultado o resultados esperados de la prueba para determinar si el caso ha sido satisfactorio o erróneo.
- g. Resultado: resultado de la prueba ya sea satisfactorio o erróneo.
- h. Fecha: fecha en que se realizó la prueba.
- i. Observaciones: comentarios u observaciones que se dieron durante la prueba.

3. Pruebas de estrés. Las pruebas de estrés permiten determinar el punto de quiebre de la aplicación desarrollada. Estas pruebas se realizaron utilizando la aplicación llamada App Perfect Load Test la cual permitió realizar las pruebas de estrés sobre los servicios web para determinar si todos los componentes del servicio podían funcionar correctamente bajo estas condiciones. Con las pruebas se busca determinar los siguientes puntos:

- a. Determinar si registra en la bitácora del servicio todas las solicitudes.
- b. Determinar que la base de datos pueda soportar una cantidad alta de solicitudes.
- c. Determinar que el tiempo de respuesta no incrementa mediante el número de transacciones crece.
- d. Determinar si siempre se obtiene una respuesta del servicio.
- e. Determinar si se obtuvieron errores no esperados durante la ejecución de las pruebas.
- f. Determinar si los servicios web soportarán la cantidad de transacciones esperadas por el negocio.

B. Instalación

Al finalizar las pruebas en el ambiente de desarrollo se trasladaron las bases de datos y lo servicios web a los servidores de producción, al tener instalado todo en estos servidores se realizaron las configuraciones necesarias y se realizaron las pruebas piloto durante un periodo de una semana para determinar el funcionamiento correcto de los componentes del proyecto y así poder iniciar operaciones y no obtener errores inesperados.

IX. Impacto y próximos desarrollos

El proyecto provee muchos beneficios a la empresa ya que soluciona la problemática que tiene hoy en día que el usuario final no puede realizar configuraciones sobre los productos como tampoco sobre los clientes de una forma fácil, por último también se cambia la forma en la que se despachaban los productos en la empresa de pasar de un sistema de colas a procesarlas en línea.

El desarrollo de este proyecto deja la puerta abierta a nuevos proyectos y a nuevos en la empresa como lo son la interfaz gráfica de la configuración de productos y clientes, como también el desarrollo de la reportería de ventas del nuevo producto y por último también realizar el desarrollo de los servicios web de cada uno de los modelos de negocio de la empresa para que estos puedan funcionar de tal forma en que las solicitudes sean operadas en el momento o en línea para así ya no encolar solicitudes y no tener el problema de saturación y de tiempos altos de entrega o de despacho de producto lo cual ocasionaba mucho disgusto con los clientes y al tener el despacho de estos productos también se debe de desarrollar la reportería de ventas de estos modelos de negocios.

X. Conclusiones y recomendaciones

1. La definición de la estructura del configurador de productos permitirá a la empresa tener un mejor control sobre sus productos y clientes.
2. Los servicios web pueden ser utilizados para muchos propósitos hoy en día, ya que se pueden ofrecer diferentes funcionalidades a través de la web, como la venta de productos electrónicos.
3. El realizar pruebas permite a las partes interesadas del proyecto puedan garantizar el buen funcionamiento del proyecto.
4. Las VPN ofrecen seguridad sobre la transmisión de datos ya que los mismos viajan encriptados y resulta complicado obtener los datos que se transmiten sobre la misma.
5. El uso de bitácoras de la aplicación permite darle una trazabilidad a los eventos de la misma y en el caso de un error es más fácil identificar donde ocurrió el mismo.
6. La toma de requerimientos es vital para que el desarrollo de un proyecto sea exitoso y se pueda determinar si el proyecto ha sido concluido con éxito.
7. Los procedimientos almacenados son una herramienta muy útil ya que permiten la reutilización de rutinas SQL y agrega seguridad a las interacciones a la base de datos.
8. La metodología de desarrollo de cascada aplico muy bien en este proyecto ya que los requerimientos estaban claramente establecidos.
9. Sin lugar a dudas durante el desarrollo del proyecto se pudieron aplicar los conocimientos aprendidos durante la carrera.
10. Se recomienda que se le dé continuidad al desarrollo del proyecto ya que esta primera fase provee muchos beneficios a la empresa.

XI. Bibliografía

1. Agarwal & S. Tayal. 2010. *Software engineering & testing, an introduction*. 1ª ed. Sudbury: Jones and Bartlett Publishers. 516 págs.
2. Bordo, Vince. *Overview of user acceptance testing (UAT) for business analysts*. <http://www.develop.com/useracceptancetests> recuperado el 24 de enero de 2013.
3. García Molina, Héctor. 2008. *Database Systems: The Complete Book*. 2ª ed. New Jersey: Prentice Hall. 1248 págs.
4. Hass, Hugo. *Web Services Architecture*. <http://www.w3.org/TR/ws-arch/> recuperado el 24 de enero de 2013.
5. Jusiak, Jan. 2010. *A Manager's Guide to User Acceptance Testing*. Melbourne: IRM Training. 109 págs.
6. Microsoft. *Virtual Private Networking*. <http://technet.microsoft.com/en-us/library/bb742566.aspx> recuperado el 24 de enero de 2013.
7. Myers, Glenford J. 2004. *The art of software testing*. 2ª ed. New Jersey: John Wiley & Sons, Inc. 227 págs.
8. Ramikrishnan, Raghu. 2002. *Databases management systems*. 3ª ed. New York: McGraw-Hill Education. 1104 págs.
9. Scott, Charlie. 1999. *Virtual Private Networks*. 2ª ed. Sebastopol: O' Reilly. 230 págs.
10. Suda, Brian. 2003. *SOAP Web Services*. School of Informatics, University of Edinburgh. 75 págs.
11. VPN Consortium. *VPN technologies: definitions and requirements*. <http://www.vpnc.org/vpn-technologies.html> recuperado el 24 de enero de 2013.

XII. Glosario

- App Load Perfect: herramienta automatizada para realizar pruebas de estrés a servicios web, desarrollados por AppPerfect.
- C#: lenguaje de programación orientado a objetos desarrollado por Microsoft.
- Fonética: estudio acerca de los sonidos de uno o varios idiomas, sea en su fisiología y acústica.
- Gliffy: herramienta web gratis para el diseño de diagramas.
- Inyección SQL: forma de realizar un ataque a una base de datos mediante la inserción de código intruso desde el ingreso de datos de una aplicación.
- NLog: es una plataforma de registro de bitácoras de código libre para .NET.
- SQL Server: manejador de base datos relacional desarrollado por Microsoft.
- Visual Studio: es un IDE para el desarrollo de aplicaciones Windows en lenguajes de programación como C#, Visual Basic, C++, entre otros.
- Windows Server: sistema operativo desarrollado por Microsoft diseñado para la administración de servidores.