

Universidad Nacional Autónoma de Nicaragua

UNAN – LEON



Facultad de Ciencias y Tecnología

*Aplicación Web para automatizar la administración e inventario del negocio “El Chompipe”  
en el mercado “El Bisne” de Chinandega.*

Monografía para optar al Título de Ingeniero en Sistema de Información.

**Autores:**

- Br .Gustavo José Baca Téllez.
- Br. Gustavo Adolfo Chavarría Reyes.
- Br. Harvin Francisco LLanes López.

**Tutor:**

- M.Sc. Eduardo Santiago Molina Poveda

Marzo del 2011, León-Nicaragua



## **Agradecimiento**

Esta monografía, si bien ha requerido de esfuerzo y mucha dedicación por parte de los autores y su director, no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada una de las personas que a continuación citaremos y muchas de las cuales han sido un soporte muy fuerte en momentos de angustia y desesperación.

Primero y antes que nada, dar gracias a Dios, por estar con nosotros en cada paso que damos, por fortalecer nuestro corazón e iluminar nuestra mente y por haber puesto en nuestro camino a aquellas personas que han sido soporte y compañía durante todo el periodo de estudio.

A nuestros padres les agradecemos su apoyo, su guía y su confianza en la realización de nuestros sueños.

Gracias padres, por todo ese apoyo incondicional que hizo posible este resultado, por ser padres más que responsables, por legarnos lo que no nos pueden quitar, por gozar y sufrir con nuestros logros y fracasos.



## Dedicatoria

Este trabajo se lo dedicamos especialmente

a nuestros padres y hermanas:

Rosa Téllez y Georgina Téllez.

Mercedes Reyes y Gustavo Chavarría.

Tania López.

Por habernos brindado

siempre su apoyo

en todo momento



## Índice

<b>I. Introducción</b>	<b>1</b>
<b>II. Antecedentes</b>	<b>3</b>
<b>III. Justificación</b>	<b>4</b>
<b>IV. Objetivos</b>	<b>5</b>
<b>V. Metodología</b>	<b>6</b>
<b>5.1. Diseño Metodológico</b>	<b>6</b>
<b>5.2. Materiales utilizados</b>	<b>7</b>
<b>VI. Marco Teórico</b>	<b>8</b>
<b>6.1. Sistemas de Inventario</b>	<b>8</b>
6.1.1. Control.	8
6.1.2. Propiedades del inventario	9
6.1.3. Costos de no tener inventario	9
6.1.4. Modelo de inventario	9
6.1.5. Tipos de Sistemas de Inventario.	10
<b>6.2. Sistema de Administración.</b>	<b>11</b>
<b>6.3. Herramientas de Desarrollo</b>	<b>14</b>
6.3.1 Microsoft .NET	14
6.3.2. C Sharp	16
6.3.3. AJAX	20
<b>6.4. Bases de Datos (BD)</b>	<b>25</b>
<b>6.5. Sistemas de gestión de bases de datos (SGBD)</b>	<b>27</b>
<b>6.6. Servidor Web</b>	<b>28</b>
<b>6.7. Modelo de Programación por Capas</b>	<b>30</b>
6.7.1. Estilos Arquitecturales	30
6.7.2. Arquitectura N-Capas.	36
<b>VII. Análisis</b>	<b>38</b>
<b>7.1. Diagrama de Casos de Uso.</b>	<b>38</b>
<b>7.2. Diagrama de clases básico</b>	<b>40</b>
<b>7.3. Diagrama Entidad – Relación</b>	<b>41</b>



7.4. Especificación de Requisitos Software (Resumen).	42
VIII. Diseño y Programación	45
8.1. Diseño de datos.	45
8.2. Diseño arquitectónico	50
8.3. Diseño de Interfaz	52
8.4. Programación	62
IX. Evaluaciones	68
X. Conclusiones	69
XI. Recomendaciones	70
XII. Bibliografía	71
XIII. ANEXOS	73
13.1. Diagrama de Flujo de datos	73
13.2. Diagramas de Secuencia	74
13.3. Especificación de Requisitos Software Completo	90
13.4. Diccionario de datos	121



## Índice de Figuras

Figura 1: Ciclo de vida Clásico.....	7
Figura 2: Modelo de Toma de decisiones.....	12
Figura 3: Modelo tradicional frente al modelo de Ajax.....	20
Figura 4: Comparación entre las llamadas síncronas en las aplicaciones clásicas frente a las llamadas asíncronas con AJAX. Fuente: (Der Henst, Falla Aroche, & Hernández) .....	22
Figura 5: .Arquitectura del Entorno ASP.NET AJAX Fuente (Rodriguez) .....	24
Figura 6: Arquitectura cliente y servidor de ASP.NET AJAX. Fuente. (Wallace B. McClure, 2006) .....	25
Figura 7: Estilo arquitectural Cliente-Servidor. Fuente: (Cesar de la Torre, 2010) .....	31
Figura 8: Estilo arquitectural basado en componentes. Fuente: (Cesar de la Torre, 2010) .....	31
Figura 9: Estilo arquitectural Presentación Desacoplada. Fuente: (Cesar de la Torre, 2010) .....	32
Figura 10: Estilo arquitectural de N-Capas. Fuente: (Cesar de la Torre, 2010) .....	33
Figura 11: Estilo Arquitectural Orientado a Dominio. Fuente: (Cesar de la Torre, 2010) .....	34
Figura 12: Estilo Orientado a Objetos. Fuente: (Cesar de la Torre, 2010) .....	34
Figura 13: Orientado a Servicios Fuente: (Cuellar, 2010).....	35
Figura 14: Arquitectura N-Capas. Fuente: (Cuellar, 2010) .....	36
Figura 15: Casos de Usos Parte #1 .....	38
Figura 16: Casos de Uso Parte #2 .....	39
Figura 17: Diagrama de Clases .....	40
Figura 18: Diagrama E- R.....	41
Figura 19: Diseño Arquitectónico.....	51
Figura 20: Captura Ingreso al sistema .....	52
Figura 21: Captura de Registrarse en el sistema.....	52
Figura 22: Captura de Pantalla de Inicio.....	53
Figura 23: Captura de Proveedores.....	54
Figura 24: Captura de Agregar Proveedores.....	54
Figura 25: Captura de Buscar Proveedores.....	55
Figura 26: Capturas de Artículos .....	55
Figura 27: Captura de Agregar Artículos.....	56
Figura 28: Captura de Clientes .....	57
Figura 29: Capturas de Ventas .....	58
Figura 30: Captura de Ventas al Crédito.....	58
Figura 31: Captura de Calcular Cambio .....	58
Figura 32: Captura de Factura Generada .....	59
Figura 33: Captura de Compras .....	59
Figura 34: Capturas de Facturas .....	60
Figura 35: Captura de Pago de Facturas .....	60
Figura 36: Captura de Caja .....	61
Figura 37: Captura de Usuarios .....	61



## I. Introducción

El presente trabajo se refiere a uno de los aspectos más importantes dentro de cualquier empresa u organización que desee optimizar sus mecanismos operacionales internos aprovechando las técnicas informáticas para la correcta gestión de la información. En líneas generales, una organización es cualquier institución compuesta de recursos, cuya combinación, permite alcanzar una serie de objetivos. Debido a ello es de suma importancia aplicar técnicas de gestión de la información para llevar a cabo el aprovechamiento de estos recursos de forma correcta, del mismo modo para todas las operaciones de almacenamiento y recuperación de información y que ésta sea presentada de forma útil y productiva.

El interés de este trabajo de fin de carrera, está basado en resolver la problemática de gestión de información del establecimiento comercial “El Chompipe” ubicado dentro del mercado de Chinandega - “El Bisne”, reemplazar la forma actual (sistema no automatizado) bajo la cual se efectúan todas las operaciones que involucran tratamiento de la información; por un sistema automatizado que ciertamente representará un auxiliar invaluable y permanente en las operaciones que directamente impliquen hacer uso de la información como podría ser, a modo de ejemplo, la toma de decisiones.

Para analizar el problema presentado por el sistema no automatizado, es condición necesaria encontrar las causas que conllevan a la pérdida de beneficios, así como las necesidades puntuales que nos fueron presentadas a la hora de tomar el desarrollo del sistema. De la misma manera, se ha tomado como soporte el documento de especificación de requisitos de Software (ERS) que servirá para encontrar un consenso entre la implementación final del sistema y la satisfacción del cliente, quien será el que haga uso de dicho sistema.

A lo largo de este documento, se presenta el enfoque que se ha implementado para llevar a cabo el desarrollo del sistema de gestión de la información para el negocio “El Chompipe”, se optó por un modelo Cliente/Servidor utilizando como interfaz de comunicación entre el usuario final y el sistema una aplicación web, específicamente la tecnología Microsoft ASP.Net para el desarrollo de páginas dinámicas; por todas las ventajas que este modelo implica y en vista de que el desarrollo actual está orientado al modelo de computo en la nube<sup>1</sup>; consiguiendo de esta manera un sistema eficiente y escalable entre otras ventajas. Una característica importante dentro de cualquier sistema automatizado es la interfaz entre el ser humano y el sistema, la cual debe ser intuitiva pero al mismo tiempo agradable, evitando en lo posible que el usuario final del sistema perciba el postback<sup>2</sup> típico de cualquier aplicación web; para solventar esto e igualar en este sentido el sistema de gestión de información a sus contrapartes de escritorio, se utilizó la tecnología AJAX, obteniendo como producto final un sistema de gestión de información con

---

<sup>1</sup> Esquema que permite ofrecer servicios de computación a través de Internet.

<sup>2</sup> Recarga completa de una página web.



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

todas la ventajas que proporcionan las aplicaciones web y la facilidad de uso de una aplicación de escritorio.

Como paradigma de programación se escogió el modelo de Programación en Capas, lo cual conlleva una serie de ventajas a la hora de migrar un sistema a tecnologías futuras o a la hora de ampliar las funcionalidades existentes que el sistema de gestión presenta.



## II. Antecedentes

El tema de programas de inventario y administración se ha centrado mayormente en aplicaciones de escritorios y en aplicaciones en red implementadas como cliente-servidor. Hasta hace 4 o 3 años se incursionó en el diseño aplicaciones web, esta nueva tecnología permite a los dueños de los negocios ampliar sus horizontes, obteniendo múltiples ventajas como el acceso remoto al sistema, la escalabilidad, etc.

Los sistemas de inventario y administración en forma de aplicaciones web se han realizados utilizando lenguajes como PHP y JavaScript, en combinación de un servidor Web, pero han surgido nuevos métodos o herramientas como lo son los lenguajes .NET en conjunto al servidor IIS, estos últimos elementos con ayuda de las técnicas de desarrollo web para crear aplicaciones interactivas o RIA(AJAX), forman un sistema ágil, sencillo, poderoso y multiplataforma que a veces llegan a ser mejores que las aplicaciones de escritorios, por ejemplo GMAIL, Yahoo! Mail, Webnote.

En la actualidad existen diversos sistemas para la administración de un negocio, pero son pocos los programas en los cuales se mantenga bien informado al usuario del estado actual del negocio por ejemplo: el artículo más vendido, el cliente que más compra, los productos más baratos en base los distintos proveedores, etc., entre estos tenemos GDS 5, programa de inventario de ferretería, SECOP, sistema de control de inventario general.

Hoy en día en el mercado el BISNE de Chinandega existen numerosos negocios, como lo es la ferretería el CHOMPIPE en donde la información de los artículos, ventas, facturas al crédito, clientes y proveedores se llevan de forma manual y no precisamente ordenada. Dicha información no es muy accesible debido a que se llevan grandes registros en donde se tiene que buscar uno por uno para encontrar lo que se necesita. Actualmente en el negocio existe únicamente un ordenador con conexión a Internet sin utilizarse en las operaciones de administración.

Debido a esta situación no se tiene una clara visión de las existencias de artículos con los que se cuenta o la ganancia real que generan estos al ser vendidos.

Por ser un sistema no automatizado ocasiona problemas que afecta la gestión administrativa/financiera del tramo El CHOMPIPE, entre los cuales tenemos: falta de decisiones rápidas y acertadas, falta de precisión en la información y demoras a la hora de realizar ventas.



### **III. Justificación**

El éxito de cualquier negocio se ve afectada por la eficiencia y rapidez con que puede llevar a cabo las operaciones de administración e inventario. Este tipo de operaciones que tradicionalmente se realizan de manera manual traen consigo el inconveniente que exigen fiabilidad y velocidad.

El gran aporte que la computación ofrece al área de administración, es la realización de manera automática de las actividades administrativas (automatización), así también la fiabilidad que ofrecen las transacciones llevadas a cabo por un ordenador es mucho más fiables que las realizadas con recurso humanos. Así que se ha considerado importante elaborar una aplicación que se encargue de estas operaciones en el Comercio “El Chompipe”.

Entre alguna de las ventajas que nuestro sistema automatizado ofrecerá a este local están: la rapidez con que el negocio será capaz de actualizar su inventario, la seguridad de la información financiera, el alto grado organización de la información, la buena presentación y elegancia que ofrece un local que hace uso de la tecnología, eficiencia y velocidad en la atención de los clientes, renovación y actualización automática de la información de proveedores, la capacidad de respaldar y recuperar esta información en caso de pérdida y sobre todo la reducción del coste material y humano del negocio.

La singularidad que una aplicación web ofrece a una empresa, microempresa o cualquier negocio es que permite el acceso remoto a la información propia de la empresa, de manera que sea posible llevar a cabo cambios que de manera manual exigirían nuestra presencia, la programación en capas de este tipo de aplicaciones permite la flexibilidad en la adaptación de la aplicación a nuevas tecnología que puedan surgir. Este proyecto contribuirá a que el comercio “El Chompipe” cuente con una aplicación que alcance las características antes mencionadas.

Como último punto cabe destacar que la implementación de este trabajo es viable y se dispone con los recursos necesarios para llevarla a cabo.



## IV. Objetivos

### Objetivo General

- Desarrollar una aplicación web que automatice las actividades de administración e inventario del negocio “El Chompipe” en el mercado “El Bisne” en Chinandega.

### Objetivos Específicos

- ✓ Proveer al usuario la posibilidad de transferir a un soporte físico la información relevante.
- ✓ Desarrollar un sistema con alto rendimiento de respuesta para todas las operaciones generadas por los usuarios.
- ✓ Controlar las entradas y salidas de dinero de caja que no son obtenidos por compras o ventas.
- ✓ Elaborar un sistema que brinde seguridad en las transacciones y en el manejo de los datos.



## V. Metodología

### 5.1. Diseño Metodológico

Para llevar a cabo el desarrollo de este trabajo se implementó la metodología del ciclo de vida clásico o en cascada.

Entiéndase por ciclo de vida todas aquellas etapas que asocian una serie de tareas que se deberán realizar incluyendo los documentos que cada una de estas generan, y que serán las que servirán de entradas a las siguientes fases; por las cuales pasa el software, desde que el proyecto es concebido hasta que este es dejado de usarse.

#### Modelo en Cascada

Descompone el proceso de desarrollo en diferentes fases, constituyendo la salida de cada una la entrada requerida por la siguiente. En este modelo se supone que todos los requisitos son conocidos y comprendidos perfectamente al iniciar el desarrollo del software.

#### Actividades del ciclo de vida en cascada

- **Análisis:** Se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos).  
Es importante señalar que en esta etapa se deben reunir todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software.
- **Diseño:** Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo.
- **Codificación:** Es la fase de programación o implementación propiamente dicha. Aquí se implementa el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores.
- **Prueba:** El software debe ser probado para descubrir los defectos que puedan existir en la función, en la lógica y en la implementación.
- **Mantenimiento:** La fase de mantenimiento se centra en el cambio. Esta fase aplica los pasos de las fases de definición y de desarrollo, pero en el contexto del software ya existe. Durante la fase de mantenimiento se centran tres tipos de cambios: corrección, adaptación y mejora.

(Sommerville, 2005)



Gráficamente el modelo de ciclo de vida clásico se representa de la siguiente manera:

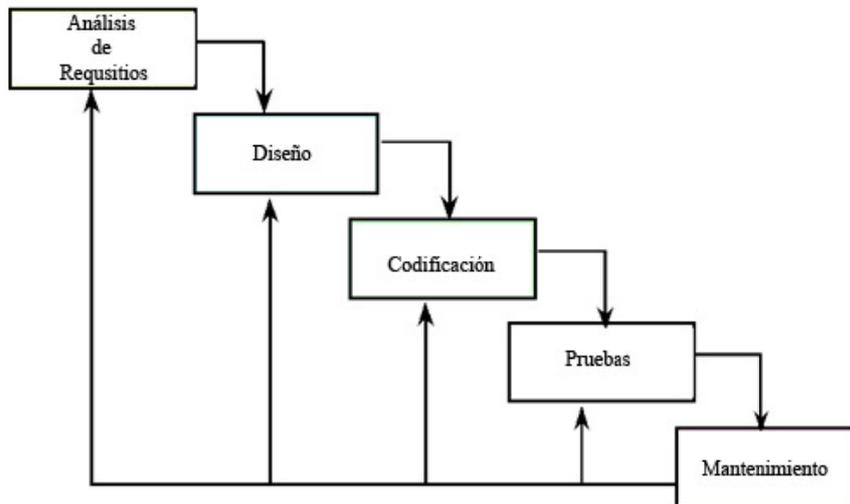


Figura1: Ciclo de vida Clásico.

## 5.2. Materiales utilizados

a. Hardware: Este proyecto a nivel de hardware se utilizaron las siguientes herramientas:

Clientes:

- CPU: 1.6Ghz . Intel Dual Core
- Ram: 1 GB.
- Disco Duro: 160 GB.

Servidor:

- CPU: 2.2Ghz. Dual Core 2 Duo.
- Ram: 4 GB.
- Disco duro: 500 GB

b. Software: Las herramientas software que se utilizaron fueron:

- Windows 7 (Sistema Operativo)
- Internet Information Server (IIS) v6.1
- Mysql versión: 3.2.4
- Microsoft Office Visio 2010
- Mozilla Firefox 3.6.13
- Visual Studios 2008



## VI. Marco Teórico

A largo de estasección se abordan los conceptos de inventario y administración, a la vez se dan a conocer las diversas herramientas para el desarrollo de este proyecto. También se abarca concepto de bases de datos y de los distintos gestores de bases de los cuales se menciona alguno, seguido de esto se entra en detalle de servidores web: concepto, funcionalidades, etc. Y por último se describe la programación en capas.

### 6.1. Sistemas de Inventario

Antes de empezar a trabajar en el desarrollo de un sistema de inventarios, es conveniente definir que es un inventario.

Según afirma (Tersine, 1999): Un inventario consiste en la existencia de productos físicos que se conservan en un lugar y momento determinado.

Una definición más precisa podría ser: Un sistema de inventarios es un conjunto de normas, métodos y procedimientos aplicados de manera sistemática para planificar y controlar los materiales y productos que se emplean en una organización. Este sistema puede ser manual o automatizado. Para el control de los costos, elemento clave de la administración de cualquier empresa, existen sistemas que permiten estimar los costos de las mercancías que son adquiridas y luego procesadas o vendidas.

La finalidad de un sistema de inventarios es encontrar las respuestas a las preguntas ¿Cuánto pedir? Y ¿Cuándo pedirlo? A manera de minimizar los costos.

En un sistema manual, la complejidad y dificultad para dar respuesta óptima a estas preguntas, crece considerablemente en la medida que se adquieren nuevos productos, por consiguiente un sistema manual para cualquier empresa que desee obtener el máximo beneficio es impensable, en este proyecto hemos trazado una línea sobre la cual migrar las operaciones a las tecnologías más actuales, con el fin de desarrollar un sistema automatizado de inventario.

#### 6.1.1. Control.

Uno de los aspectos más importantes en el desarrollo de este proyecto es el tratar de controlar. Pero que es control.

Se define en (Greene, 2001) que Control es la medición y corrección del desempeño con el fin de asegurar que se cumplan los objetivos de la empresa y los planes diseñados para alcanzarlos.



### **6.1.2. Propiedades del inventario**

**Demanda.**- Son las unidades requeridas que se toman del inventario en un determinado tiempo.<sup>5</sup>

**Productos.**-Los productos pueden ser uno o varios. Así mismo se pueden clasificar por unidad o por lote dependiendo del proceso; perecederos o duraderos dependiendo de su vida útil, divisibles o indivisibles, etc.

El objetivo primordial del control del inventario es tener la cantidad apropiada de materia prima u otros materiales y productos terminados en el lugar adecuado, en el tiempo oportuno y con el menor costo posible. Los costos excesivos en inventarios pueden ser por malas decisiones en el establecimiento de un sistema.

Es necesario hacer notar, que en un sistema manual, los errores humanos intencionados o no, debilitan enormemente la eficiencia de dichos sistemas; sin embargo con un sistema automatizado, se eliminan muchos de estos errores, también puede existir otro tipo de fallos propios de estos sistemas por ejemplo: pérdida de información por fallos de voltaje, sin embargo en este trabajo investigativo hemos tenido presente ese tipo de problemas y también las posibles soluciones; en el caso de pérdidas de información por fallos de voltaje, hemos diseñado un mecanismo fácil y automático para generar respaldos de la información (backup).

### **6.1.3. Costos de no tener inventario**

Como se explica en (Tersine, 1999), estos costos pueden tener su origen en faltantes externos cuando a un cliente no se le puede surtir una orden ocasionando órdenes pendientes, disminución en las ventas y pérdida de prestigio comercial, o internos cuando un departamento dentro de la organización no cuenta con materiales o artículos ocasionando pérdidas de producción, retraso en las fechas de entrega.

### **6.1.4. Modelo de inventario**

La naturaleza del problema de inventario consiste en hacer y recibir pedidos de determinados volúmenes, repetidas veces y a intervalos determinados. Una política de inventario responde a la siguiente pregunta.

*¿Cuánto se debe ordenar?*

El costo de compra se basa en el precio por unidad del artículo. Puede ser constante, o se puede ofrecer con un descuento que depende del volumen del pedido.



El costo de faltante es la penalidad en la cual se incurre cuando nos quedamos sin existencias. Incluye la pérdida potencial de ingresos, así como el costo más subjetivo de la pérdida de la buena voluntad de los clientes.

*¿Cuándo se deben hacer los pedidos?*

Depende del tipo de sistema de inventario que tenemos. Si el sistema requiere una revisión periódica (por ejemplo, semanal o mensual), el momento para hacer un nuevo pedido coincide con el inicio de cada periodo.

El administrador financiero debe tener la información pertinente que le permita tomar decisiones sobre el manejo que se le debe dar a la empresa, en este proyecto hemos tenido en cuenta este punto, como uno de los más importantes, ya que el hecho de disponer de información útil y actualizada acerca del estado de los artículos, representa la diferencia entre tomar una buena decisión o no.

### **6.1.5. Tipos de Sistemas de Inventario.**

#### *6.1.5.1. Sistema de inventario periódico*

Mediante este sistema, los comerciantes determinan el valor de las existencias de mercancías mediante la realización de un conteo físico en forma periódica, el cual se denomina inventario inicial o final según sea el caso.

**Inventario inicial:** Es la relación detallada y minuciosa de las existencias de mercancías que tiene una empresa al iniciar sus actividades, después de hacer un conteo físico.

**Inventario final:** Es la relación de existencias al finalizar un periodo contable.

#### *6.1.5.2. Sistema de inventario permanente*

Por medio de este sistema la empresa conoce el valor de la mercancía en existencia en cualquier momento, sin necesidad de realizar un conteo físico, porque los movimientos de compra y venta de mercancías se registran directamente en el momento de realizar la transacción a su precio de costo.

Actualmente se utiliza el siguiente método para determinar las cantidades físicas de un inventario, muchos de los cuales requieren o permiten la utilización de un equipo de trabajo. En el método más sencillo un miembro de un grupo de trabajo cuenta, pesa, mide o hace el estimado de la cantidad a otro miembro del grupo quien se encargara de anotarlo en una hoja de inventario. A continuación se le presenta la flotación de un inventario físico en la hoja de control de Inventario o Libros de Inventarios que es donde se registran todos los movimientos de



las mercancías, al final de cada periodo deben registrarse en el libro de inventarios las existencias de las mercancías, así como el contenido de los estados financieros de manera resumida.

El cálculo de costos de las mercancías vendidas es un concepto muy importante que requiere de una cuidadosa atención.

El costo de compra (p) es el precio unitario de un artículo si este fue adquirido de fuente externa o proveedor, y debe ser registrado en nuestro costo de inventario como tal.

## **6.2. Sistema de Administración.**

El término sistema de administración es un tanto ambiguo, sin embargo en el contexto del desarrollo de un software, se entenderá como aquel sistema que estará pensado para cubrir áreas tales como el control de inventarios, la gestión de clientes, presupuestos, pedidos y ventas, entre otras operaciones con lo que se podrá realizar una mejor gestión para de esta forma sacar un mayor provecho a los recursos que ya poseen y lograr una mayor competitividad con el mínimo de inversión.

Una organización o empresa debe ser vista bajo la óptica de un sistema abierto, El sistema abierto interactúa constantemente con el ambiente en forma dual, o sea, lo influye y es influenciado. El sistema cerrado no interactúa.

Existen diferencias entre los sistemas abiertos (como los sistemas biológicos y sociales, a saber, células, plantas, el hombre, la organización, la sociedad).

Es propio del sistema abierto competir con otros sistemas, no así el sistema cerrado. Para que una empresa pueda competir con otras empresas es necesario, saber aprovechar de manera práctica y útil la información que posee, organizar de manera adecuada sus elementos (productos, capital, etc.).

Al igual que los organismos vivos, las empresas tienen seis funciones primarias, estrechamente relacionadas entre sí: (Vozes, 2004)

**Ingestión:** las empresas hacen o compran materiales para ser procesados. Adquieren dinero, máquinas y personas del ambiente para asistir otras funciones, tal como los organismos vivos ingieren alimentos, agua y aire para suplir sus necesidades.

- **Procesamiento:** los animales ingieren y procesan alimentos para ser transformados en energía y en células orgánicas. En la empresa, la producción es equivalente a este ciclo. Se procesan materiales y se desecha lo que no sirve, habiendo una relación entre las entradas y salidas.



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

- **Reacción al ambiente:** el animal reacciona a su entorno, adaptándose para sobrevivir, debe huir o si no atacar. La empresa reacciona también, cambiando sus materiales, consumidores, empleados y recursos financieros. Se puede alterar el producto, el proceso o la estructura.
- **Provisión de las partes:** partes de un organismo vivo pueden ser suplidas con materiales, como la sangre abastece al cuerpo. Los participantes de la empresa pueden ser reemplazados, no son de sus funciones sino también por datos de compras, producción, ventas o contabilidad y se les recompensa bajo la forma de salarios y beneficios. El dinero es muchas veces considerado la sangre de la empresa.
- **Regeneración de partes:** las partes de un organismo pierden eficiencia, se enferman o mueren y deben ser regeneradas o relocalizadas para sobrevivir en el conjunto. Miembros de una empresa envejecen, se jubilan, se enferman, se desligan o mueren. Las máquinas se vuelven obsoletas. Tanto hombres como máquinas deben ser mantenidos o relocalizados, de ahí la función de personal y de mantenimiento.
- **Organización:** de las funciones, es la requiere un sistema de comunicaciones para el control y toma de decisiones. En el caso de los animales, que exigen cuidados en la adaptación. En la empresa, se necesita un sistema nervioso central, donde las funciones de producción, compras, comercialización, recompensas y mantenimiento deben ser coordinadas. En un ambiente de constante cambio, la previsión, el planeamiento, la investigación y el desarrollo son aspectos necesarios para que la administración pueda hacer ajustes.

Un esquema general del modelo final a la hora de tomar decisiones aprovechando las ventajas de un sistema de administración, son reflejadas en el siguiente gráfico.

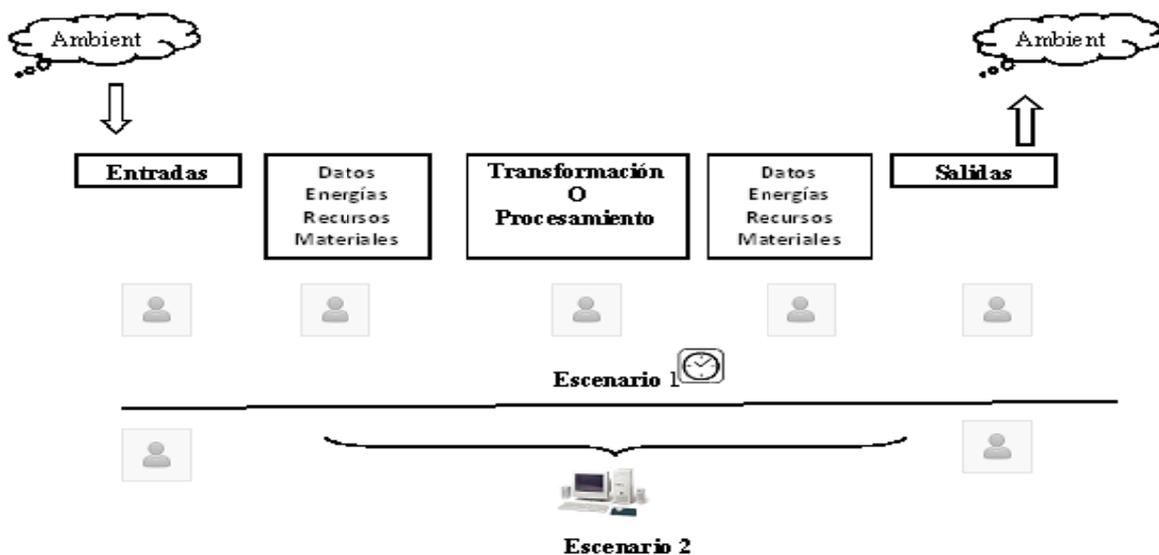


Figura2: Modelo de Toma de decisiones.



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

Entrada o insumo o impulso (input): es la fuerza de arranque del sistema, que provee el material o la energía para la operación del sistema.

Salida o producto o resultado (output): es la finalidad para la cual se reunieron elementos y relaciones del sistema. Los resultados de un proceso son las salidas, las cuales deben ser coherentes con el objetivo del sistema.

Ambiente: es el medio que envuelve externamente el sistema. Está en constante interacción con el sistema, ya que éste recibe entradas, las procesa y efectúa salidas. La supervivencia de un sistema depende de su capacidad de adaptarse, cambiar y responder a las exigencias y demandas del ambiente externo. Aunque el ambiente puede ser un recurso para el sistema, también puede ser una amenaza.

El primer escenario corresponde al esquema actual bajo el cual se efectúan todas las operaciones de administración y organización de todas las funciones.

En este esquema vemos como todas las operaciones implican la participación directa de una entidad humana, en todo lo referente a cuestiones administrativas. Más allá del tedio y de ser sensible a errores humanos, este sistema implica un gran gasto de energía y tiempo, que podrían ser ocupados para operaciones que verdaderamente requieran la participación humana, como es la toma de decisiones.

El segundo escenario, corresponde a un esquema automatizado, en el las operaciones mecánicas, quedan destinadas al software que se diseñara para tal fin. Esto redundara en productividad y eficiencia.

En pocas palabras, el software de administración se encarga de todas aquellas tarea que impliquen aplicar un pensamiento algorítmico, mientras que el operador humano se encarga de aplicar mayormente su pensamiento heurístico, en la toma de decisiones.

Se denomina heurística a la capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos, desde cuyo punto de vista puede describirse como el arte y la ciencia del descubrimiento y de la invención o de resolver problemas mediante la creatividad y el pensamiento lateral o pensamiento divergente.

En (Pensamiento Complejo y computadoras, Matthew Lipman) se define pensamiento algorítmico se entiende a aquel tipo de pensamiento, que puede ser computado, programado, y que es determinado bajo una serie finita de pasos, en este proyecto se determinarán ambos tipos de pensamientos dentro el último estará destinado en mayor medida al software.



### 6.3. Herramientas de Desarrollo

#### 6.3.1 Microsoft .NET

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Sun Microsystems y a los diversos framework de desarrollo web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones –o como la misma plataforma las denomina, soluciones– permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.(Microsoft, 2009)

La plataforma .NET de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows.

.NET Framework se incluye en Windows Server 2008, Windows Vista y Windows 7. De igual manera, la versión actual de dicho componente puede ser instalada en Windows XP, y en la familia de sistemas operativos Windows Server 2003. Una versión "reducida" de .NET Framework está disponible para la plataforma Windows Mobile, incluyendo teléfonos inteligentes.

La norma (incluido en ECMA-335, ISO/IEC 23271) que define el conjunto de funciones que debe implementar la biblioteca de clases base (BCL por sus siglas en inglés, tal vez el más importante de los componentes de la plataforma), define un conjunto funcional mínimo que debe implementarse para que el marco de trabajo sea soportado por un sistema operativo. Aunque Microsoft implementó esta norma para su sistema operativo Windows, la publicación de la norma abre la posibilidad de que sea implementada para cualquier otro sistema operativo existente o futuro, permitiendo que las aplicaciones corran sobre la plataforma independientemente del sistema operativo para el cual haya sido implementada. El Proyecto Mono emprendido por Ximian pretende realizar la implementación de la norma para varios sistemas operativos adicionales bajo el marco del código abierto.

Los principales componentes del marco de trabajo son:

- El conjunto de lenguajes de programación
- La *Biblioteca de Clases Base* o *BCL*
- El *Entorno Común de Ejecución para Lenguajes* o *CLR* por sus siglas en inglés.



Debido a la publicación de la norma para la **infraestructura común de lenguajes** (*CLI* por sus siglas en inglés), el desarrollo de lenguajes se facilita, por lo que el marco de trabajo .NET soporta ya más de 20 lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos, lo que elimina las diferencias que existían entre lo que era posible hacer con uno u otro lenguaje.

Algunos de los lenguajes desarrollados para el **marco de trabajo .NET** son: C#, Visual Basic, Delphi (Object Pascal), C++, J#, Perl, Python, Fortran, Prolog (existen al menos dos implementaciones, el P#y el Prolog.NET, Cobol y PowerBuilder.

### 6.3.1.1. CommonLanguageRuntime (*CLR*)

Este es el lenguaje insignia de .NET Framework (*marco de trabajo .NET*) y pretende reunir las ventajas de lenguajes como C, C++ y Visual Basic en uno solo. El CLR es el verdadero núcleo del framework de .NET, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo (W2k y W2003). (Microsoft, 2009)

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio, el MSIL (*Microsoft IntermediateLanguage*), similar al BYTECODE de Java. Para generarlo, el compilador se basa en la especificación CLS (*CommonLanguageSpecification*) que determina las reglas necesarias para crear el código MSIL compatible con el CLR.

Para ejecutarse se necesita un segundo paso, un compilador JIT (*Just-In-Time*) es el que genera el código máquina real que se ejecuta en la plataforma del cliente. De esta forma se consigue con .NET independencia de la plataforma de hardware. La compilación JIT la realiza el CLR a medida que el programa invoca métodos. El código ejecutable obtenido se almacena en la memoria caché del ordenador, siendo recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente.

### 6.3.1.2. Características

Es el encargado de proveer lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta. Los servicios son variados:

- Cargador de clases: permite cargar en memoria las clases.
- Compilador MSIL a nativo: transforma código intermedio de alto nivel independiente del hardware que lo ejecuta a código de máquina propio del dispositivo que lo ejecuta.
- Administrador de código: coordina toda la operación de los distintos subsistemas del CommonLanguageRuntime.
- Recolector de basura: elimina de memoria objetos no utilizados.
- Motor de seguridad: administra la seguridad del código que se ejecuta.



- Motor de depuración: permite hacer un seguimiento de la ejecución del código aun cuando se utilicen lenguajes distintos.
- Verificador de tipos: controla que las variables de la aplicación usen el área de memoria que tienen asignado.
- Administrador de excepciones: maneja los errores que se producen durante la ejecución del código.
- Soporte de multiproceso (hilos): permite ejecutar todo el código en forma paralela.
- Empaquetador de COM: coordina la comunicación con los componentes COM para que puedan ser usados por el .NET Framework.
- Soporte de la Biblioteca de Clases Base: interfaz con las clases base del .NET Framework. Esto quiere decir que existen tipos de estructuras como es la de java y la .NET

### 6.3.2. C Sharp

C# (pronunciado *si sharp* en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO (Microsoft, 2009)

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET, similar al de Java aunque incluye mejoras derivadas de otros lenguajes (entre ellos Delphi).

La creación del nombre del lenguaje, C#, proviene de dibujar dos signos positivos encima de los dos signos positivos de "C++", queriendo dar una imagen de salto evolutivo, del mismo modo que ocurrió con el paso de C a C++.

C#, como parte de la plataforma.NET, está normalizado por ECMA desde diciembre de 2001 (C# LanguageSpecification "Especificación del lenguaje C#"). El 7 de noviembre de 2005 salió la versión 2.0 del lenguaje, que incluía mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. El 19 de noviembre de 2007 salió la versión 3.0 de C#, destacando entre las mejoras los tipos implícitos, tipos anónimos y LINQ (*LanguageIntegratedQuery* -consulta integrada en el lenguaje).

Aunque C# forma parte de la plataforma.NET, ésta es una interfaz de programación de aplicaciones (API), mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco de DotGNU - Mono que genera programas para distintas plataformas como Win32, UNIX y Linux.

### 6.3.2 ASP.NET

ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el CommonLanguageRuntime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework. (Microsoft, 2009)

Cualquier persona que está familiarizada con el desarrollo de aplicaciones web sabrá que el desarrollo web no es una tarea simple. Ya que mientras que un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo, la programación web es una mezcla de varios lenguajes de etiquetas, un gran uso de lenguajes de *script* y plataformas de servidor. Por desgracia para el programador de nivel intermedio, el conocimiento y habilidades que se necesitan para desarrollar aplicaciones web tienen muy poco en común con las que son necesarias en el desarrollo tradicional de aplicaciones.

Microsoft introdujo la tecnología llamada Active Server Pages en diciembre de 1996. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente. La definición contextual de Microsoft es que *"Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web"*.

Después del lanzamiento del Internet Information Services 4.0 en 1997, Microsoft comenzó a investigar las posibilidades para un nuevo modelo de aplicaciones web que pudiera resolver las quejas comunes sobre ASP, especialmente aquellas con respecto a la separación de la presentación y el contenido y ser capaz de escribir código "limpio". A Mark Anders, un administrador del equipo de IIS y Scott Guthrie, quien se había unido a Microsoft en 1997 después de graduarse de la Universidad Duke, se les dio la tarea de determinar cómo debería ser ese modelo. El diseño inicial fue desarrollado en el curso de dos meses por Anders y Guthrie, y Guthrie codificó los prototipos iniciales durante las celebraciones navideñas de 1997.

### **6.3.2.1 Páginas**

Las páginas de ASP.NET, conocidas oficialmente como "*web forms*" (formularios web), son el principal medio de construcción para el desarrollo de aplicaciones web. Los formularios web están contenidos en archivos con una extensión ASPX; en jerga de programación, estos archivos típicamente contienen etiquetas HTML o XHTML estático, y también etiquetas definiendo *Controles Web* que se procesan del lado del servidor y *Controles de Usuario* donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web.

Adicionalmente, el código dinámico que se ejecuta en el servidor puede ser colocado en una página dentro de un bloque `<% -- código dinámico -- %>` que es muy similar a otras tecnologías de desarrollo como PHP, JSP y ASP, pero esta práctica es, generalmente,



desaconsejada excepto para propósitos de enlace de datos pues requiere más llamadas cuando se genera.

ASP.NET Es limitado a solo funcionar con IIS, por lo que su uso es cuestionado por la mayoría de los programadores web quienes prefieren otros lenguajes de programación del lado del servidor como por ejemplo PHP, Perl, Java Etc.

### **6.3.2.2 El modelo Code-behind**

Microsoft recomienda que para realizar programación dinámica se use el modelo code-behind, o de respaldo, que coloca el código en un archivo separado o en una etiqueta de script especialmente diseñada. Los nombres de los archivos *code-behind* están basados en el nombre del archivo ASPX tales como *MiPagina.aspx.cs* o *MiPagina.aspx.vb* (esta práctica se realiza automáticamente en Microsoft Visual Studio y otras interfaces de desarrollo). Cuando se usa este estilo de programación, el desarrollador escribe el código correspondiente a diferentes eventos, como la carga de la página, o el clic en un control, en vez de un recorrido lineal a través del documento.

El modelo *code-behind* de ASP.NET marca la separación del ASP clásico y alienta a los desarrolladores a construir aplicaciones con la idea de presentación y contenido separados en mente. En teoría, esto permite a un diseñador web, por ejemplo, enfocarse en la creación del diseño con menos posibilidades de alterar el código de programación mientras lo hace. Esto es similar a la separación en el Modelo Vista Controlador

### **6.3.2.3.Estado de la sesión**

El estado de la sesión (Sessionstate) es una colección de variables definidas por el usuario, las cuales persisten durante la sesión de un usuario. Estas variables son únicas para diferentes instancias de una sesión de usuario, y son accedidas usando la colección `Session`. Las variables de sesión pueden ser preparadas para ser automáticamente destruidas después de un determinado tiempo de inactividad, incluso si la sesión no ha terminado. Del lado del cliente, una sesión de usuario es identificada por una cookie o codificando el ID de la sesión en la misma URL.

ASP.NET proporciona tres modos de persistencia para variables de sesión



## **InProc**

Las variables de sesión son mantenidas dentro del proceso. Sin embargo, en este modo, las variables son destruidas cuando el proceso ASP.NET es reciclado o terminado.

## **StateServer**

En este modo, ASP.NET ejecuta un servicio de Windows separado que mantiene las variables de estado. Como esta administración de estado ocurre fuera del proceso ASP.NET, tiene un impacto negativo en el rendimiento, pero permite a múltiples instancias de ASP.NET compartir el mismo estado del servidor, permitiendo que una aplicación ASP.NET pueda tener su carga balanceada y escalada en múltiples servidores. También, como el servicio de administración del estado se ejecuta independiente de ASP.NET, las variables pueden persistir a través de las finalizaciones del proceso ASP.NET.

## **SqlServer**

En este modo, las variables de estado son almacenadas en un servidor de base de datos, accesible usando SQL. Las variables de sesión pueden persistir a través de finalizaciones de procesos también en este modo.

En el modelo de desarrollo web basado en páginas activas, la programación ASP actual tiene diversas limitaciones:

- Para que todo ocurra en una página web, es habitual escribir una gran cantidad de código para resolver necesidades sencillas. ASP.NET incorpora un modelo declarativo a la programación web: los controles de servidor funcionan en una página Web simplemente declarándolos. Cuando se carga la página ASP.NET, se instancian los controles listados en la página ASP y es responsabilidad del control emitir código HTML que el navegador pueda entender.
- ASP clásico es un tanto desorganizado. En una página ASP podemos incluir casi todo: HTML plano, código script, objetos COM y texto. No hay una distinción formal entre el contenido de una página y su comportamiento: simplemente, insertamos código en la página, y a ver qué pasa. ASP.NET impone un cierto orden sobre el modelo de programación estándar ASP. En cierto modo, esta "desorganización" puede evitarse fácilmente usando el sentido común y algunas de las nuevas tecnologías. Por ejemplo, podemos escribir en nuestras páginas ASP únicamente código VBScript. Dicho código generaría un mensaje XML, que luego sería interpretado por un archivo XSLT. De esta forma conseguimos evitar el llamado "código spaguetti", aumentando la claridad del código y la velocidad de ejecución de las páginas ASP.



- La tercera limitación en el desarrollo con ASP es que con el tradicional utilizamos lenguajes de scripting no tipados como VBScript o JScript. Podemos instalar otros motores de scripting que impongan verificación de tipos; sin embargo, no son universalmente conocidos o utilizados como los anteriores. ASP.NET claramente separa la porción basada en script de una página web de su contenido.

### 6.3.3. AJAX

Acrónimo de **A**synchronous**J**ava**S**cript**a**nd**X**ML por sus siglas en Ingles (JavaScript Asíncrono y XML). Ajax no es en sí una tecnología sino la unión de muchas tecnologías (Garret), que trabajan de manera conjunta para dotar a una aplicación Web de características interactivas.

Las aplicaciones desarrolladas a principios de la evolución de la Web trabajaban según el siguiente modelo: una acción realizada en la interfaz dispara una petición al servidor web, el servidor realiza algún procesamiento y devuelve una página HTML al cliente, mientras el servidor realizaba este procesamiento, el cliente esperaba una respuesta viendo una ventana en blanco en su navegador, haciendo de este un proceso tedioso y vulgar.

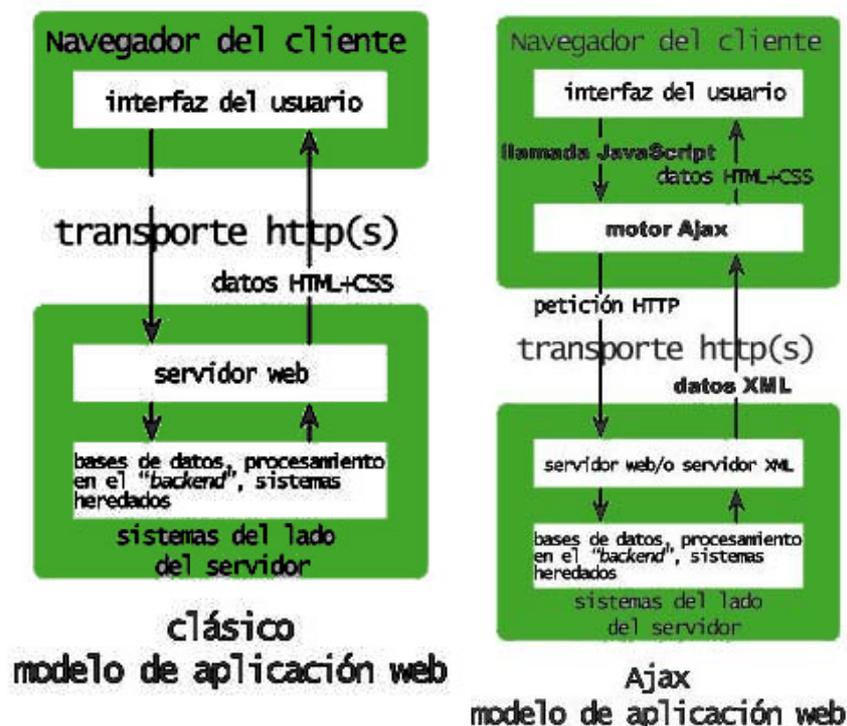


Figura3: Modelo tradicional frente al modelo de Ajax.

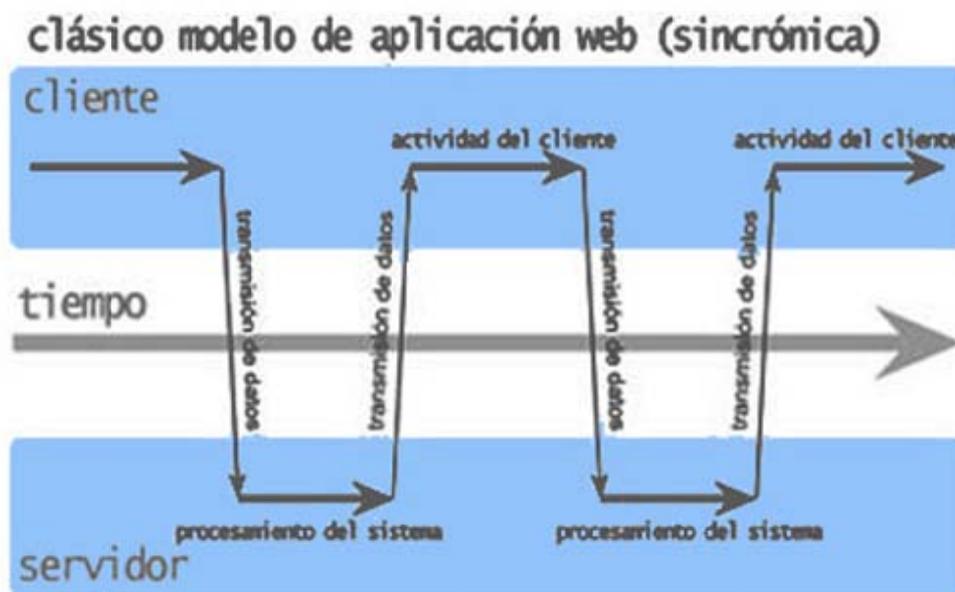
Fuente:(Der Henst, Falla Aroche, & Hernández)



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

Contrario a este comportamiento estático, las aplicaciones desarrolladas con AJAX eliminan el inconveniente de petición-espera de las aplicaciones primitivas. En lugar de cargar una página web al inicio de la sesión, el navegador carga un motor AJAX escrito en Java Script y usualmente escrito en el marco oculto. Este motor se encarga de renderizar la interfaz que el usuario ve y se comunica con el servidor en nombre del usuario. El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente - independiente de comunicación con el servidor. Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena, esperando a que el servidor para hacer algo.(Garret).

AJAX maneja cada acción que realiza un usuario sobre la aplicación como una llamada JavaScript al motor AJAX, que normalmente generaría una solicitud HTTP, de esta manera cualquier repuesta que no requiera un viaje de vuelta al servidor es manejado por esta. En caso de que el motor AJAX necesite datos del servidor se hace una llamada asíncrona evitando así que la aplicación deje de interactuar con el usuario.



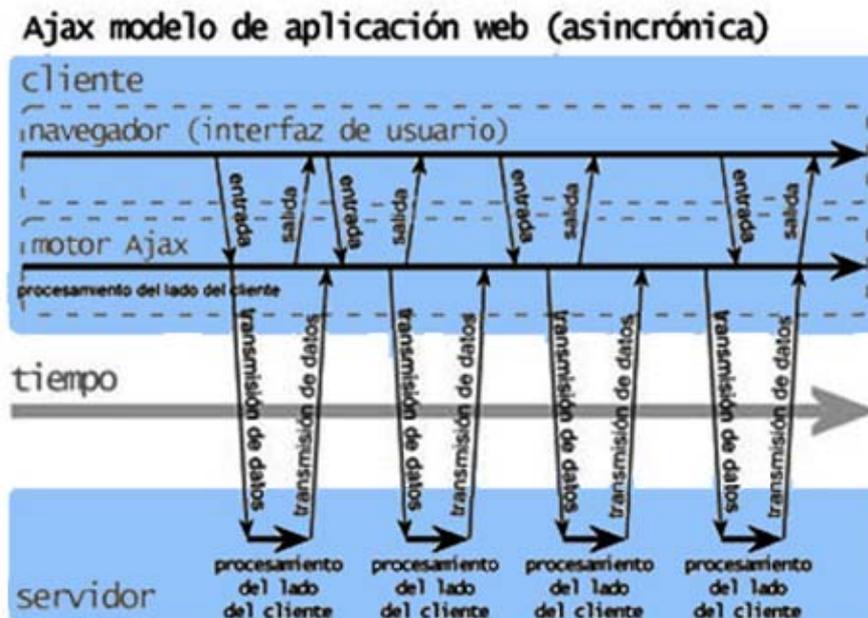


Figura 4: Comparación entre las llamadas síncronas en las aplicaciones clásicas frente a las llamadas asíncronas con AJAX. Fuente:(Der Henst, Falla Aroche, & Hernández)

#### 6.3.3.1. Ventajas de AJAX

- ✓ **Asíncrono** –Ajax ofrece la funcionalidad de hacer llamadas asíncronas al servidor web. Evitando al cliente esperar por la llegada de los datos.
- ✓ **Transferencia mínima de los datos**-Evita hacer un postback completo y enviar todos los datos al servidor, el uso de la red es minimizado y el procesamiento ocurre rápidamente. En redes en la cuales hay restricciones del ancho de banda esta es una gran característica.
- ✓ **Reducción del procesamiento en el servidor**- Debido a que únicamente los datos necesarios son enviados al servidor, el servidor no requiere procesar todos los elementos evitando sobrecargar al servidor.
- ✓ **Respuestabilidad**-Gracias a que las aplicaciones en AJAX son asíncronas en el cliente, este percibe una alta Respuestabilidad.
- ✓ **Permanencia del Contexto**-Con una solicitud completa de la página el usuario puede perder el contexto en que se encontraba. Con AJAX el usuario puede estar en la parte inferior de la página, hacer una solicitud y la página no es recargada completamente permitiendo al usuario mantener su localización.

(Wallace B. McClure, 2006)



### 6.3.3.2. *Tecnologías que hacen realidad AJAX*

El termino AJAX fue establecido por Jesse Jame Garret de AdaptivePath en un artículo publicado en febrero de 2005 y ampliamente aceptado en la comunidad de desarrolladores de software. El termino como tal es una combinación de las tecnologías en uso, Java Script Asíncrono y XML.

- ✓ **XMLHttpRequest**-La historia de AJAX está íntimamente ligada con este objeto. Este objeto permite al navegador conversar con el servidor sin requerir de una solicitud postback. En Internet Explorer esta habilidad es provista por los componentes MSXML ActiveX. En Mozilla Firefox esta característica es provista por un objeto literalmente llamado XMLHttpRequest. Las librerías JavaScript del lado del cliente ocultan la diferencia entre los diferentes ambientes de los navegadores.
- ✓ **JavaScript**-JavaScript provee la capacidad de comunicar al cliente con el servidor de manera imperceptible al usuario. La versión requerida es la 1.5. JavaScript no es específicamente necesario, su utilidad radica desde el punto de vista del cliente en el soporte para el manejo de los script en la mayoría de navegadores modernos.
- ✓ **Soporte DHTML/DOM**-Un navegador debe de poseer la habilidad de actualizar dinámicamente sus elementos, esta habilidad es la que ofrece DocumentObjectModel (DOM).
- ✓ **Transporte de Datos con XML o JSON**- Usando XML permite la capacidad de comunicarse con el servidor en un mecanismo estándar. Hay situaciones en las que Java Script ObjectNotation(JSON), el cual es un lenguaje liviano de comunicación, es usado como el mecanismo entre el cliente y el navegador.

(Wallace B. McClure, 2006)

A continuación una lista de algunas aplicaciones que usan AJAX:

- ✓ **Buscador de Google:** conocido como Google Suggest es una característica incluida en este buscador la cual sugiere de manera dinámicamente en un drop-downlist una búsqueda asociada a la introducida por el usuario.
- ✓ **Google Maps:** La habilidad de hacer un acercamiento en un área del mapa sin hacer un postback es sorprendente.
- ✓ **Google Gmail:** El gestor de correo de Google comparte las funcionalidades incluidas en el navegador.
- ✓ **Live.com:** El servicio de correo de Microsoft hace uso del framework Atlas para dotar de muchas funcionalidades propias de AJAX.

### 6.3.3.3. *AJAX y ASP.NET*

ASP.NET AJAX es una nueva tecnología de Microsoft para el desarrollo Web que facilita el uso y extiende los beneficios de la metodología AJAX sobre la plataforma ASP.NET(Rodríguez). ASP.NET 2.0 introduce una nueva técnica para implementar paginas AJAX denominada



*ScriptCallbacks*(Ceballos, 2007). El objetivo que se perseguía era el mismo que en AJAX, generar peticiones asíncronas desde el lado del cliente al servidor cuando sea necesario refrescar parcialmente la página; sin embargo hay una diferencia significativa entre esta técnica y los objetivos de AJAX y es que *script callbacks* nos permite prescindir del manejo de la clase XMLHttpRequest. En resumen es una técnica para realizar postback en segundo plano utilizando el modelo de eventos del servidor .NET.

(Ceballos, 2007)

No obstante el desarrollo de aplicaciones Web con programación del lado del cliente requiere del conocimiento de los lenguajes JavaScript, manejo de objetos DOM y demás.

Para evitar estas limitaciones Microsoft desarrollo una paquete de extensiones para ASP.NET encargadas facilitar el uso de AJAX y extender los beneficios sobre la plataforma Punto NET. Este paquete es compuesto por el AJAX Control Toolkit, Microsoft AJAX Library, ASP.NET AJAX Extensions y ASP.NET AJAX Server Control.

Los objetivos que persigue este proyecto son:

- ✓ Dotar de compatibilidad con la mayoría de navegadores las aplicaciones que se desarrollen.
- ✓ Soporte de bibliotecas JavaScript mediante ensamblados o en base a ficheros .js independientes.
- ✓ Servicios Web encargados de la validación, extensibilidad de control y otros.
- ✓ Controles del lado del servidor capaces de integrarse con el cliente para producir un comportamiento igual al de AJAX.

#### 6.3.3.4. *Arquitectura de ASP.NET AJAX*

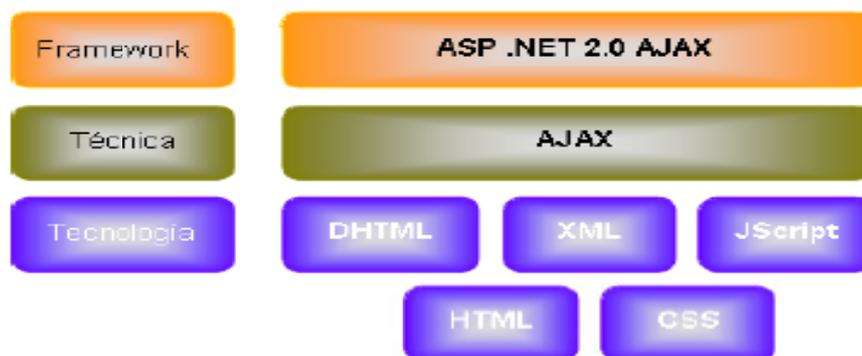
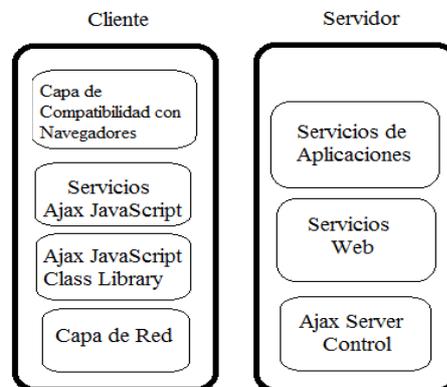


Figura 5: .Arquitectura del Entorno ASP.NET AJAX Fuente (Rodríguez)



**Figure 6: Arquitectura cliente y servidor de ASP.NET AJAX.** Fuente.(Wallace B. McClure, 2006)

#### 6.4. Bases de Datos (BD)

Una base de datos o banco de datos, es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. (Wikipedia, wikipedia.org, 2010)

Actualmente las bases de datos son unos de los elementos más utilizados a la hora de guardar grandes cantidades de información, debido a que ofrecen un amplio rango de características, como son:

- Independencia de los datos: es decir que los datos no dependen del programa y por tanto cualquier aplicación puede hacer uso de los datos.
- Reducción de la redundancia: llamamos redundancia a la existencia de duplicación de los datos, al reducir esto conseguimos un mayor aprovechamiento del espacio.
- Acceso concurrente por parte de múltiples usuarios: cada gestor o motor de base de datos tienen distintos tipos de técnicas de bloqueo o cerrado de datos para poder controlar la concurrencia; mejorando así la integridad de los datos y reduciendo la redundancia.
- Distribución espacial de los datos: la independencia lógica y física facilita la posibilidad de sistemas de bases de datos distribuidas. Los datos pueden encontrarse en otro habitación, otro edificio e incluso en otro país. El usuario no tiene por qué preocuparse de la localización espacial de los datos a los que accede.
- Integridad de los datos: Se refiere a las medidas de seguridad que impiden que introduzcan datos erróneos. Como se había dicho anteriormente esto puede ser causado por un mal control de concurrencia, motivos físicos (defectos hardware, actualizaciones incompleta debido a causas externas), como de operación (introducción de datos incoherentes).



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

- Consultas complejas optimizadas: la optimización de consultas permite la rápida ejecución de las mismas.
- Seguridad de acceso y auditoría: se refiere al derecho de acceso a los datos contenidos en la base de datos por parte de personas y organismos. En esta parte permite generar diferentes tipos de permisos para el acceso a secciones de información dando como resultado una organización jerárquica de usuarios. El sistema de auditoría mantiene el control de acceso a la base de datos, con el objeto de saber qué o quién realizó una determinada modificación y en qué momento.
- Respaldo y recuperación: dicha característica se refiere a la capacidad de un sistema de bases de datos de recuperar su estado previo a la pérdida de datos.
- Acceso a través de lenguajes de programación estándar: consiste en la posibilidad ya mencionada de acceder a los datos de una base de datos mediante lenguajes de programación ajenos al sistema de base de datos propiamente dicho.

Una vez conocidas las diferentes características de las BD, entraremos en detalle de los distintos modelos de bases de datos que existen según el libro (Paré, Casillas Santillán, Costa Costal, Ginesta, Escofet, & Mora, 2005):

- Modelo Jerárquico: Sus estructuras son registros interrelacionados en forma de árboles. El sistema gestor de BD clásico de este modelo lo es el IMS/DL1 de IBM.
- Modelo en Red: Como en el modelo jerárquico hay registros e interrelacionados, pero un registro ya no está limitado a ser “hijo” de un solo registro.
- Modelo Relacional: Dicho modelo se basa en el concepto matemático de relación, que aquí podemos considerar de momento equivalente al término tabla.
- Modelo relacional con objetos: Se trata de ampliar el modelo relacional, añadiéndole la posibilidad de que los tipos de datos sean tipos abstractos de datos.

Todo modelo de bases de datos nos proporciona tres tipos de herramientas:

- a) Estructuras de datos con las que se puede construir la base de datos: tablas, árboles, etc.
- b) Diferentes tipos de restricciones (o reglas) de integridad que el sistema gestor de BD tendrá que hacer cumplir a los datos: dominios, claves, etc.
- c) Una serie de operaciones para trabajar con los datos: Un ejemplo de ello en el modelo relacional, es la operación SELECT, que sirve para seleccionar las filas que cumplen alguna condición.



### **6.5. Sistemas de gestión de bases de datos (SGBD)**

Los SGBD son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan (Wikipedia, Wikipedia.org, 2010).

Existen distintos objetivos que deben cumplir los SGBD:

**Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos.

**Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

**Consistencia.** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente.

**Seguridad.** Los SGBD deben garantizar que esta información se encuentra segura de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

**Manejo de transacciones.** Una transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.

**Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Como todo sistema consta de ventajas y desventajas las cuales se mencionaran a continuación:

**Ventajas:**

- Proveen facilidades para la manipulación de grandes volúmenes de datos.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

**Desventajas:**

1. Típicamente, es necesario disponer de una o más personas que administren de la base de datos, en la misma forma en que suele ser necesario en instalaciones de cierto porte disponer de una o más personas que administren los sistemas operativos. Esto puede llegar a incrementar los costos de operación en una empresa. Sin embargo hay que balancear este aspecto con la calidad y confiabilidad del sistema que se obtiene.



2. Si se tienen muy pocos datos que son usados por un único usuario por vez y no hay que realizar consultas complejas sobre los datos, entonces es posible que sea mejor usar una planilla de cálculo.
3. Complejidad: los programas muy complejos y las personas que vayan a usarlo deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.
4. Tamaño: la complejidad y la gran cantidad de funciones que tienen hacen que sea un software de gran tamaño, que requiere de gran cantidad de memoria para poder correr.
5. Coste del hardware adicional: los requisitos de hardware para correr un SGBD por lo general son relativamente altos, por lo que estos equipos pueden llegar a costar gran cantidad de dinero.

Luego de tener una idea de lo que hace un SGBD y de sus ventajas y desventajas podremos mencionar algunos, populares en el mercado: SQLserver, PostgreSQL, MySQL, Oracle, etc.

Todos estos sistemas son de gran ayuda para el manejo de información, de forma rápida, segura y accesible; que combinados con un lenguaje de programación forman una herramienta útil y poderosa para cualquier empresa o persona.

## 6.6. Servidor Web

En la actualidad el Internet ha pasado de ser un lujo a una necesidad, todo usuario o empresa necesita acceder a la web por distintas razones, pero la principal es compartir información, para hacer esto existen diversas maneras de hacerlo, ya sea mediante blogs, redes sociales, etc. Es aquí donde sobresalen los servidores web, que es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El programa implementa el *protocolo HTTP (HyperText Transfer Protocol)* que pertenece a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa (Wikipedia, 2010).

Dadas las funcionalidades que presentan dichos servidores, su uso ha venido incrementando gradualmente con base a las nuevas computadoras, anteriormente la mayoría de usuarios que utilizaban esta herramienta eran las empresas y los recursos con que contaban no eran muchos, pero con el avance de la tecnología y de la web ahora se pueden crear servidores web caseros para uso personal, de forma fácil y sencilla, sin invertir un solo centavo.

El funcionamiento de un servidor web se puede observar de forma sencilla así: El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una *página web* que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. A modo de ejemplo, al teclear *www.direccionweb.com* en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma

Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java "applets" o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas *scripts*). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje *javascript* y *java*, aunque pueden añadirse más lenguajes mediante el uso de *plugins*.
- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

Hoy en día existen diversos servidores web, como lo es APACHE, IIS, CHEROCKEE, etc; estos pueden ser gratuitos o privados, pero ambos funcionan de forma poderosa combinados con un lenguaje de programación con el que se vaya a implementar, cabe mencionar que cada servidor web está asociado a una serie de lenguajes de programación pero no a todos, por lo que hay saber cuál utilizar en base a lo que se desea realizar y los componentes a utilizar.

Estos servidores web se han vuelto tan populares que ahora ya vienen combinados con una serie de programas o utilidades, que son fáciles de instalar y sin costo, por ejemplo: WAMP, XAMPP, etc.

Una última ventaja con que cuentan algunos servidores web es que son multiplataforma, brindando su uso a todo tipo de usuario.



## **6.7. Modelo de Programación por Capas**

La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.(es.Wikipedia.org, 2010)

La idea de la programación en capas se propone facilitar el desarrollo del diseño de la arquitectura la cual se entiende como: "... el proceso por el cual se define una solución para los requisitos técnicos y operacionales de un sistema"(Cesar de la Torre, 2010)

Respondiendo a esta necesidad de agilizar el proceso de desarrollo de la arquitectura el estilo de desarrollo de aplicaciones ha cambiado dando a luz a lo que se conoce como programación en capas.

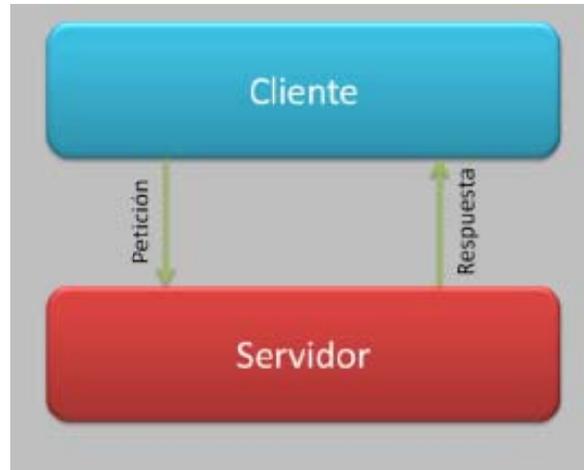
Además, este modelo permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles. Es por esta razón que la mayoría de las compañías de software han optado por este nuevo estilo de programación, muestra de ello son la estrategia de Sun con su proyecto SunTone o la de Microsoft con el proyecto DotNet (.Net)

### **6.7.1. Estilos Arquitecturales**

Un estilo arquitectural se define como un conjunto de principios que definen a alto nivel un aspecto de la aplicación, los principales aspectos son: comunicación, despliegue, dominio, interacción, relación y estructura.(Cesar de la Torre, 2010), entre los estilos arquitecturales más relevantes destacan.

#### **❖ Cliente-Servidor.**

Este modelo define una relación entre dos aplicaciones de las cuales una es cliente y otra es servidor, es el modelo por excelencia difundió en los inicios de la Internet.



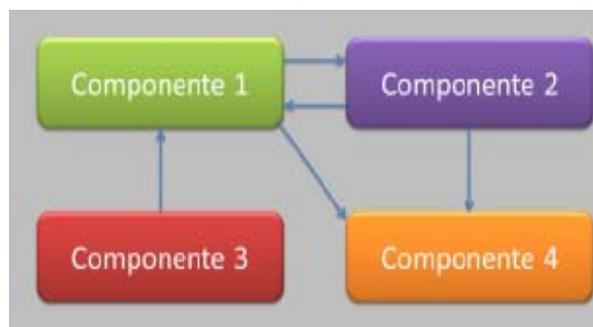
**Figura7: Estilo arquitectural Cliente-Servidor. Fuente: (Cesar de la Torre, 2010)**

Entre sus características destacan

- ✓ Permite el uso de un amplio rango de protocolos y formatos de datos.
- ✓ El esquema del modelo divide el sistema en un cliente un servidor y una red.
- ✓ El servidor no realiza ninguna petición al cliente.
- ✓ El servidor envía los datos.
- ✓ La identidad del cliente normalmente debe ser verificada.

#### ❖ Basado en Componentes

Este estilo describe un acercamiento al diseño de sistemas como un conjunto de componentes que exponen interfaces bien definidas y que colaboran entre si para resolver el problema.



**Figura 8: Estilo arquitectural basado en componentes. Fuente: (Cesar de la Torre, 2010)**



### Características

- ✓ Un estilo para diseñar aplicaciones a partir de componentes individuales.
- ✓ Enfatiza la descomposición del sistema en componentes con interfaces bien definidas.
- ✓ Se logra una aproximación a través de componentes que se comunican mediante interfaces.

### ❖ **Presentación Desacoplada**

El estilo de presentación desacoplada indica cómo debe realizarse el manejo de las acciones del usuario, la manipulación de la interfaz y los datos de la aplicación. Este estilo separa los componentes de la interfaz de flujo de datos de la manipulación.

#### Características.

- ✓ Un estilo, para diseñar aplicaciones basadas en patrones de diseño.
- ✓ Separa la lógica para el manejo de la interacción de la representación de los datos con que trabaja el usuario.
- ✓ Permite a los diseñadores crear una interfaz gráfica mientras los desarrolladores escriben el código para su funcionamiento.



**Figura9: Estilo arquitectural Presentación Desacoplada. Fuente: (Cesar de la Torre, 2010)**

### ❖ **Arquitectura de N-Capas (N Layer)**

El estilo arquitectural en capas se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y forma de interacción con otras capas y las responsabilidades la funcionalidad que implementan.



**Figura10: Estilo arquitectural de N-Capas. Fuente: (Cesar de la Torre, 2010)**

#### Características

- ✓ Descomposición de los servicios de forma que la mayoría de las interacciones ocurren solo entre capas.
- ✓ La capa de aplicación puede residir en la misma máquina o pueden estar distribuidos entre varios equipos.
- ✓ Los componentes de cada capa se comunican con los componentes de otras capas a través de interfaces bien conocidos.
- ✓ Cada nivel agrega las responsabilidades y abstracciones del nivel inferior.
- ✓ Muestra una vista completa del modelo y a la vez proporciona suficientes detalles para entender las relaciones entre capas.
- ✓ Separa de forma clara la funcionalidad de cada capa.

#### ❖ **Orientada al Dominio**

Orientado para diseñar e implementar aplicaciones empresariales complejas donde es fundamental definir un Modelo de Dominio expresado en el propio lenguaje de los expertos del dominio de negocio real.

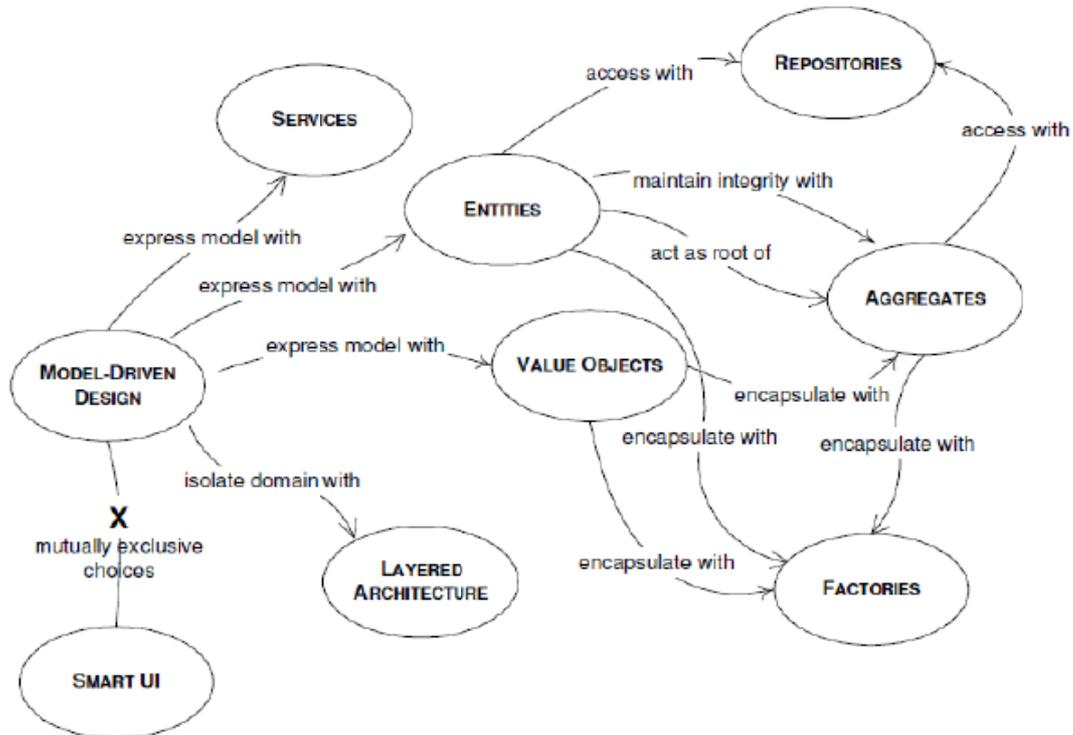


Figura 11: Estilo Arquitectural Orientado a Dominio. Fuente: (Cesar de la Torre, 2010)

Características.

- ✓ Arquitecturas N-Capas (Capas específicas tendencias arquitectura DDD).
- ✓ Patrones de Diseño.
- ✓ Es fundamental el desacoplamiento entre componentes.

### ❖ Orientado a Objetos

Es un estilo que define el sistema como un conjunto de objetos que cooperan entre sí en lugar de como un conjunto de procedimientos. Los objetos son discretos, independientes y poco acoplados, se comunican mediante interfaces.

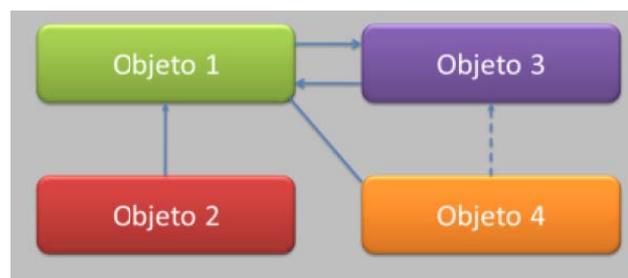


Figura 12: Estilo Orientado a Objetos. Fuente: (Cesar de la Torre, 2010)



### Características.

- ✓ Es un estilo para diseñar aplicaciones basadas en un número de unidades lógicas y código reusable.
- ✓ Describe el uso de objetos que contienen los datos y el comportamiento para trabajar con esos datos y además tiene un rol o responsabilidad distinta.
- ✓ Se hace hincapié en la reutilización a través de la encapsulación, la modularidad y los polimorfismos y la herencia.
- ✓ Su enfoque es completamente distinto al procedimental ya que en este estilo los objetos interactúan unos con otros para realizar tareas.

### ❖ Orientado a Servicios

Este estilo permite ofrecer su funcionalidad como un conjunto de servicios para que sean consumidos. Los servicios utilizan interfaces estándares que pueden ser invocadas, publicadas y descubiertas.

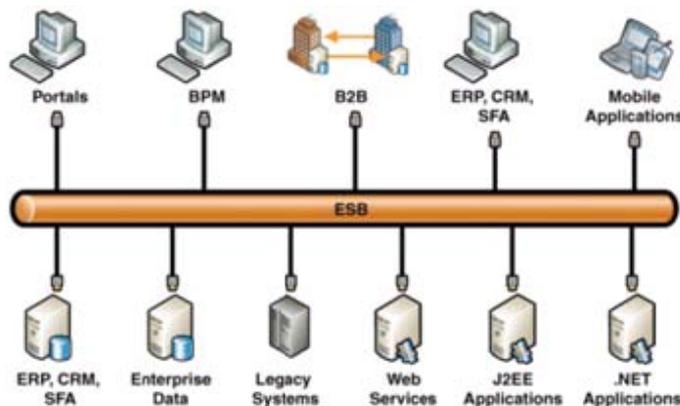


Figura13: Orientado a Servicios Fuente: (Cuellar, 2010)

### Características.

- ✓ La interacción con los servicios están muy desacopladas.
- ✓ Los procesos del negocio se pueden empaquetar como servicios.
- ✓ Los clientes y otros servicios pueden acceder a los servicios locales en el mismo nivel.
- ✓ Los clientes y los servicios pueden hacer contacto con los servicios remotos mediante la red.
- ✓ Permite el uso de un amplio rango de protocolos y formato de datos.

Fuente.(Cesar de la Torre, 2010)



### 6.7.2. Arquitectura N-Capas.

Pese a la gran variedad de estilos arquitectónicos, el estilo que se ha impuesto es el estilo de N-Capas, dentro de sus múltiples versiones la más utilizada es el modelo de Presentación Desacopladas. Dicha implementación destaca en aplicaciones con cierta complejidad.

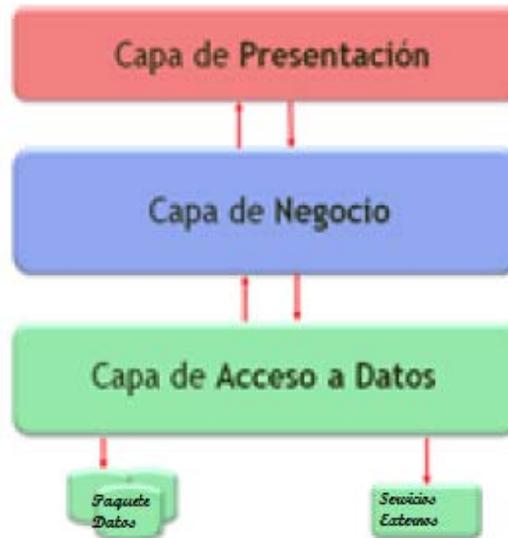


Figura14: Arquitectura N-Capas. Fuente: (Cuellar, 2010)

1. **Capa de presentación:** es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.
2. **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
3. **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Fuente.(es.Wikipedia.org, 2010)



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que reside la capa de negocio.



## VII. Análisis

### 7.1. Diagrama de Casos de Uso.

A continuación se mostrará todas las operaciones que podrá realizar el usuario:

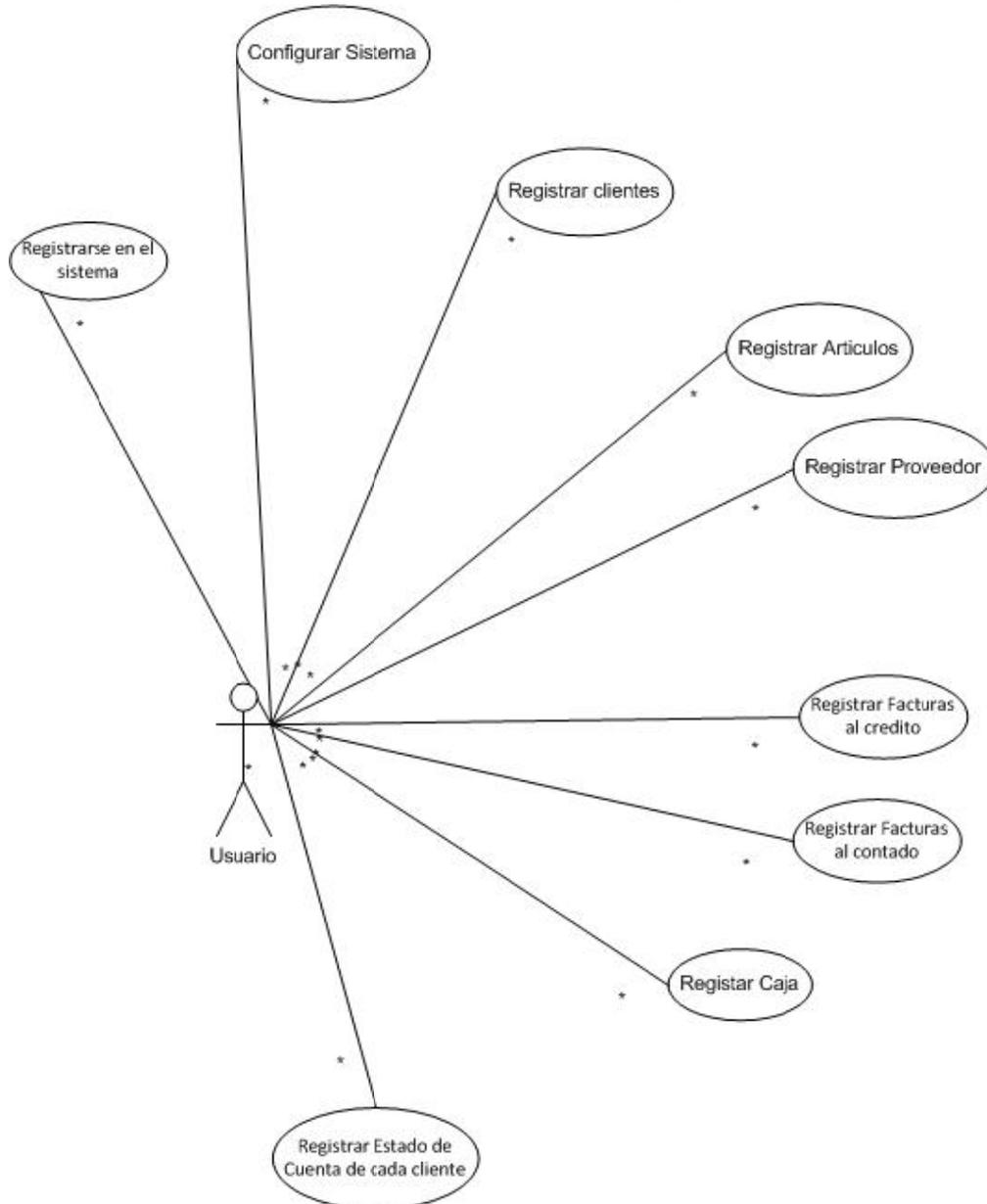


Figura15: Casos de Usos Parte #1

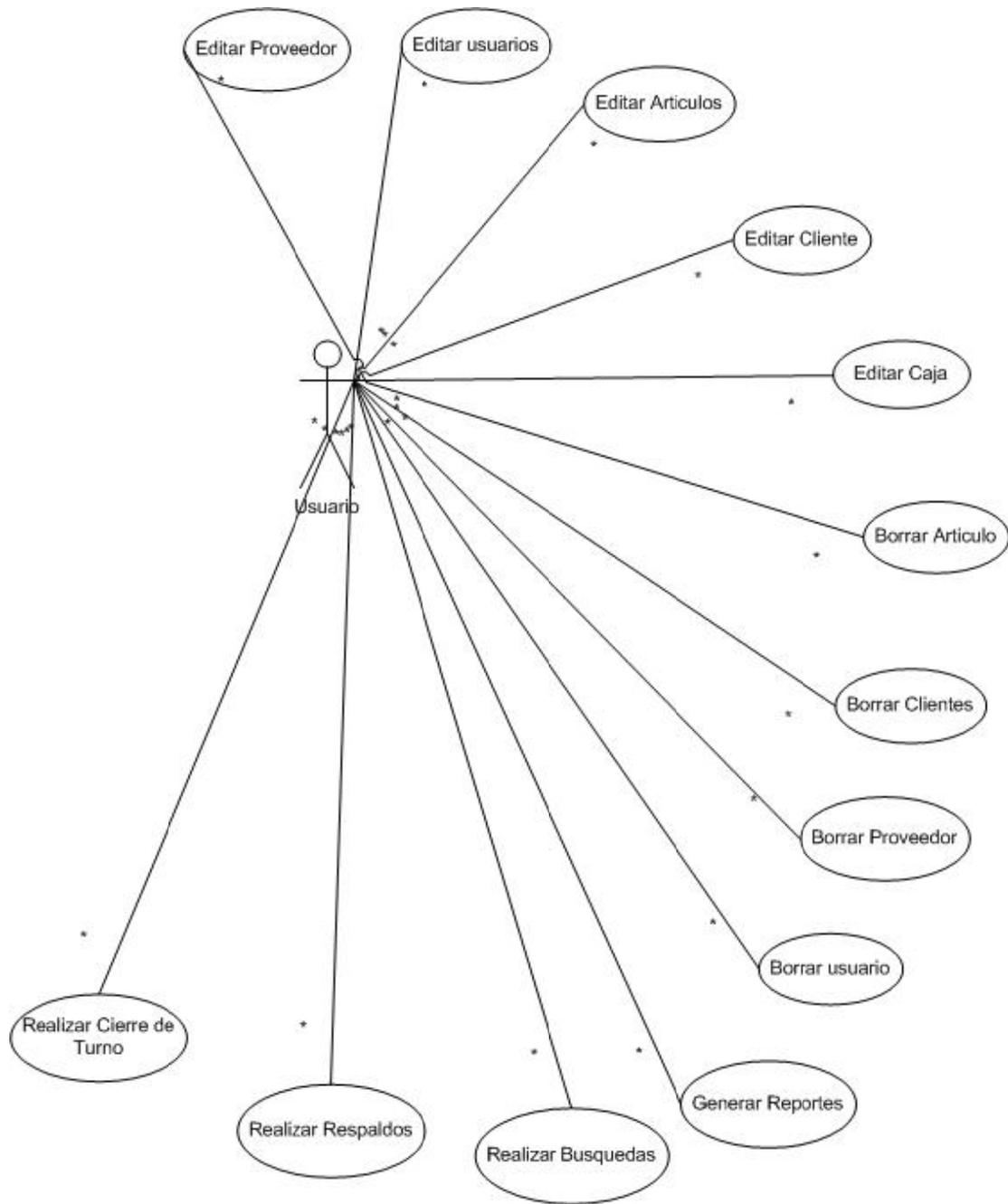


Figura16: Casos de Uso Parte #2



## 7.2. Diagrama de clases básico

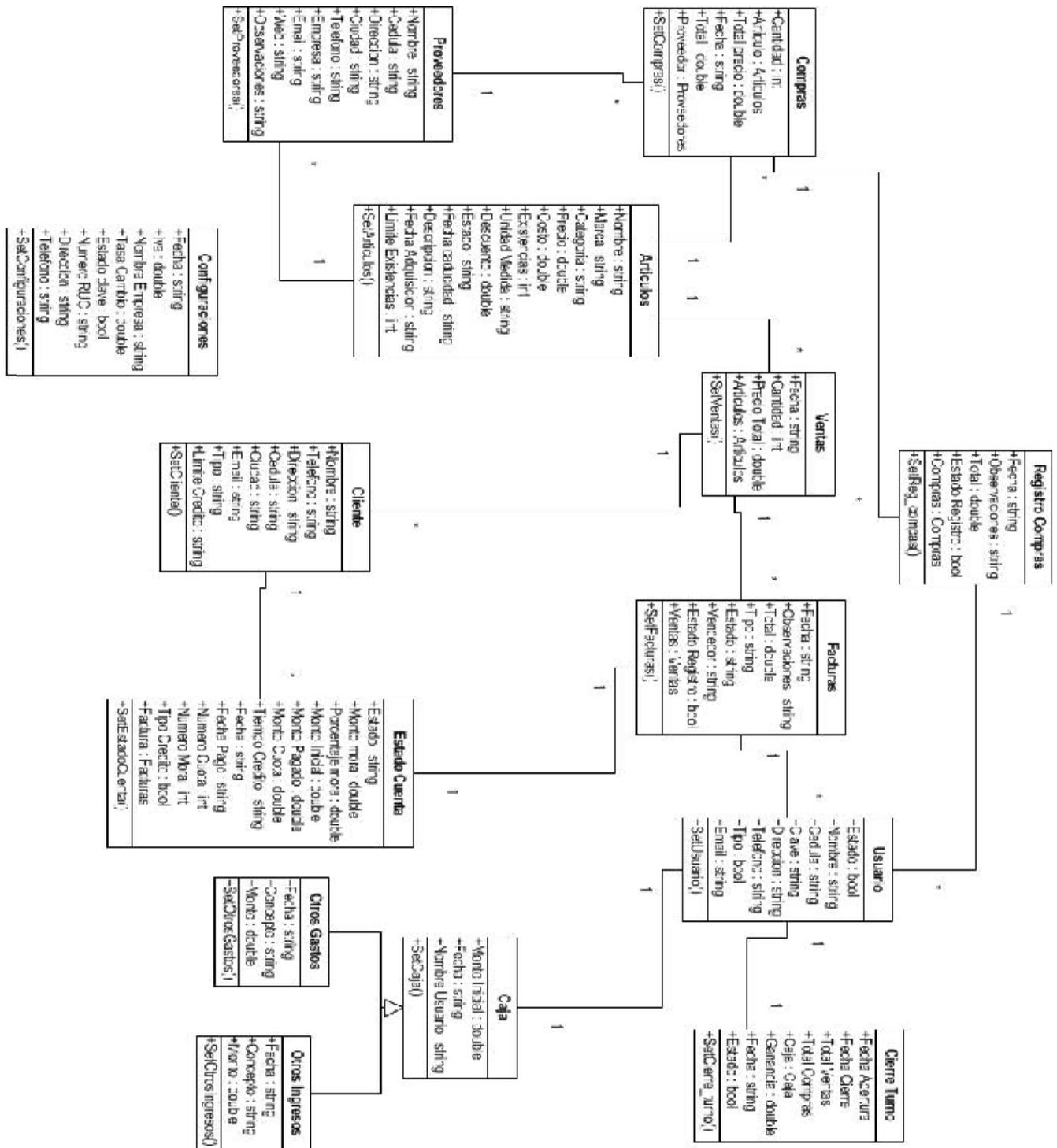


Figura17: Diagrama de Clases





## **7.4. Especificación de Requisitos Software (Resumen).**

### **1. Introducción**

#### Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales y del sistema para la implementación de una aplicación WEB que permitirá llevar a cabo las operaciones de inventario del negocio “El Chompipe” en el mercado el Bisne del Chinandega.

Este documento va dirigido a los propietarios del negocio y a los usuarios finales que deberán estudiarlo para su aprobación o desacuerdo antes de abordar la fase de análisis.

#### Alcance

Diseño, desarrollo e implementación del Sistema de Inventario del negocio “El Chompipe” El Sistema de Inventario será una aplicación que funcionara en entorno Web que permitirá realizar las actividades de administración e inventario del negocio. Esta aplicación realizara las siguientes funciones.

- ✓ Registrar Usuarios en el Sistema.
- ✓ Configurar el Sistema.
- ✓ Captura y registro de los datos de un Cliente.
- ✓ Captura y registro de los datos de un Proveedor.
- ✓ Registro de Artículos.
- ✓ Captura de Facturas al Crédito y al Contado.
- ✓ Registro de Caja.
- ✓ Captura del estado de cuenta de cada cliente.
- ✓ Edición de los datos de un Proveedor.
- ✓ Edición de los datos de un Usuario.
- ✓ Edición de los datos de un Artículo.
- ✓ Edición de los datos de Caja.
- ✓ Eliminar un Artículo.
- ✓ Eliminar un Cliente.
- ✓ Eliminar un Proveedor.
- ✓ Eliminar un Usuario.
- ✓ Generar los siguientes Reportes:
- ✓ Reporte de Ventas por rango de fechas.
- ✓ Reportes de Facturas por cobrar, por rango de fechas o por usuario.
- ✓ Reporte de Existencia de Productos y Productos próximos a agostarse.
- ✓ Reporte de Listado de Compras.
- ✓ Realizar búsqueda de Artículos, Proveedores, Clientes.
- ✓ Cierre de turno del negocio.
- ✓ Realizar una copia de respaldo de toda la información de la empresa.



## 2. Descripción general

### 2.1. Perspectiva del producto

El Sistema de Inventario será una aplicación diseñada para trabajar en entorno WEB, lo que permitirá su uso descentralizado, es además un producto independiente de otros sistemas.

### 2.2. Restricciones

#### Políticas de Software.

EL Sistema de Inventario se desarrollara usando software que trae el sistema operativo por lo que no se deberá pagar por el uso de: servidor WEB (IIS), Sistema de Gestión de base de datos (MySQL) y lenguaje de programación C# en conjunto con la tecnología ASP.NET. El uso de este software estará regido por las políticas de licenciamiento de software.

#### Limitaciones de Hardware.

Para el funcionamiento de la aplicación es necesario un computador servidor en el cual se instalara el servidor Web IIS, MySQL, el complemento de AJAX par ASP.NET y la aplicación de Sistema de Inventario.

#### Interfaces con otras aplicaciones.

La aplicación será un sistema autónomo y no interactuara con otras aplicaciones por lo que no se desarrollara interfaces con otras aplicaciones. Las conexiones necesarias para la conexión del servidor web, MySQL, Asp.Net , se hará por medio de la configuración de estos programas.

#### Requisitos de Fiabilidad.

Los datos correspondiente con los clientes, proveedores debe de verificarse. Los datos correspondientes a caja deben de estar ajustados a una moneda estándar y a una tasa de cambio uniforme.

#### Credibilidad de la aplicación.

Para garantizar la credibilidad de la aplicación al usuario final, el sistema será sometido a una serie de pruebas de funcionamiento para establecer que se encuentra acorde a los requerimientos que se plasman en este documento así como la consistencia de los datos, redimiendo de la aplicación y tiempo de respuesta.



#### Consideraciones de Seguridad.

Cada usuario debe de autenticarse y su acceso verificado por una sola terminal para su correcto desempeño según sus rol. Todas las claves de seguridad deben de estar en su defecto encriptados en la base de datos para ofrecer un correcto funcionamiento.

#### 2.3. Suposiciones y dependencias

- ✓ La red interna deberá de estar configurada para el manejo de protocolos TCP/IP, HTTP, DNS, principalmente todo lo relacionado en cuanto a desempeño y seguridad.
- ✓ Debe realizarse una capacitación adecuada y acorde a lo que cada usuario va a realizar. Su capacitación de hará en el momento que sea necesaria y a las persona indicada.

### 3. Requisitos no funcionales

#### Requisitos de rendimiento

El sistema trabajará en una red compuesta de cuatro computadoras, donde no todas se conectarán simultáneamente. El tiempo de cada transacción no será mayor a 5 segundos. Donde el servidor será una pc dedicada a este servicio, de características normales.

#### Seguridad

El sistema constará de contraseñas para el acceso a este, encriptados en alguno de los algoritmos de encriptación conocidos, al igual la base de datos también constará de clave de acceso encriptado.

También constará con diferentes niveles de acceso según el tipo de usuario protegiendo información de gran importancia.

El sistema llevará un registro de las operaciones realizadas por cada usuario.

La aplicación hará uso del protocolo HTTPS para establecer una conexión segura.

#### Portabilidad

La aplicación podrá ser usada en cualquier sistema operativo que tenga un navegador web. El servidor puede ser exportado solo si se cambia al servidor apache por otro lado se manejara en la plataforma Windows



## VIII. Diseño y Programación

En este capítulo del trabajo se mostrará algunos de diagramas realizados para llevar a cabo el sistema, imágenes de la aplicación y porciones de código más importantes.

### 8.1. Diseño de datos.

**Tabla 1: artículos**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
Idarticulos	autoincremento(int)	11	Código con el que se identifica a el articulo
Nombre	varchar	60	Nombre del Articulo
Marca	varchar	40	Marca del Articulo
Categoría	varchar	100	Categoría en la que se clasifica el articulo
precio_unitario	double	11	Precio por unidad de medida del articulo
Costo			Costo de compra del articulo
Existencias	int	11	Número de artículos en existencias
unidad_medida	varchar	25	Unidad de Medida del articulo
Descuento	doublé	16	Descuento sobre el articulo
Estado	enum('bueno', 'retira', 'regular', devuelto')		Estado en que se encuentra el articulo
fecha caducidad	varchar	18	Fecha de Vencimiento
Descripción	Text	65535	Descripción sobre el articulo
fecha adquisición	varchar	18	Fecha de compra
Idproveedor	integer	11	Código del proveedor

**Tabla 2: proveedores**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
Idproveedor	autoincremento(int)	11	Código del Proveedor
nombre_proveedor	varchar	40	Nombre del Proveedor
Cedula	varchar	18	Cedula de identidad del Proveedor
Dirección	varchar	100	Dirección del Proveedor
Ciudad	varchar	30	Ciudad de residencia del Proveedor
Teléfono	varchar	15	Número de teléfono
Empresa	varchar	30	Nombre de la Empresa del Proveedor
Email	varchar	40	Correo electrónico del Proveedor
Web	varchar	40	Página Web de la Empresa del Proveedor
Observaciones	Text	65535	Observaciones sobre el proveedor

**Tabla 3. Clientes**



Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idclientes	autoincremento (int)	11	Código del Cliente
nombre	varchar	40	Nombre del Cliente
telefono	varchar	15	Número de teléfono del Cliente
direccion	varchar	100	Dirección del Cliente
ciudad	varchar	30	Ciudad de residencia del Cliente
cedula	varchar	18	Cédula de identidad del Cliente
email	varchar	40	Correo electrónico del Cliente
limite_credito	doble	16	Límite de crédito del Cliente
tipo	varchar	40	Tipo de Cliente

**Tabla 4: usuario**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idusuario	autoincremento (int)	11	Código del Usuario
Fecha	timestamp		Fecha de registro del Usuario
estado	enum('desconectado','conectado')		Estado de conectividad del Usuario
nombre	varchar	40	Nombre del Usuario
cedula	varchar	40	Cédula de identidad del Usuario
clave	varchar	15	Contraseña del Usuario
direccion	varchar	100	Dirección del Usuario
telefono	varchar	15	Número de teléfono del Usuario
tipo	enum('administrador','normal')		Tipo de Usuario

**Tabla 5: facturas**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
Idfactura	autoincremento (int)	11	Código de la Factura
Fecha	date	10	Fecha de registro de la factura
Observaciones	text	65535	Observaciones
Total	double	16	Total en la Factura
Tipo	enum('credito','contado')		Tipo de Factura
Estado	enum('cancelada','vigente','pagada')		Estado de la Factura
Vendedor	varchar	40	Vendedor que registra la Factura
estado_registro	boolean	1	Indica si una Factura al Contado está o no pagada

**Tabla 6: estado\_cuenta**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idestado_cuenta	autoincremento (int)	11	Código del Estado de Cuenta



Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

estado	enum('solvente', 'mora', 'baja')		Estado del Estado de Cuenta
monto_mora	double	16	Monta de mora a aplicar
porcentaje_mora	double	16	Porcentaje a aplicar en caso de mora
monto_inicial	double	16	Monto inicial del Estado de Cuenta
monto_cuota	double	16	Monto a pagar por cuota
monto_pagado	double	16	Monto total pagado hasta última cuota
tiempo_credito	varchar	20	Tiempo de credito del Estado de Cuenta
Idclientes	int	11	Codigo del Cliente al que pertenece el Estado de Cuenta
fecha	varchar	18	Fecha de inicio del Estado de Cuenta
idfactura	idfactura	11	Codigo de la factura indicada en el Estado de Cuenta
fecha_pago	varchar	18	Ultima fecha de pago
num_mora	int	11	Numero de mora a aplicar
num_cuota	int	11	Numero de cuota pagadas
tipo_credito	enum('Mensual', 'Quincenal')		Tipo de crédito del Estado de Cuenta

**Tabla 7: configuraciones**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idconfiguraciones	autoincremento (int)	11	Codigo de la Configuracion
Fecha	Timestamp	18	Fecha y hora del establecimiento de la Configuracion
Iva	double	16	Porcentaje a aplicar de IVA
nombre_empresa	varchar	50	Nombre de la Empresa
tasa_cambio	double	16	Tasa de cambio del dólar
Idusuario	int	11	Código del usuario que estableció la configuración
estado_clave	boolean	1	Indica si la configuración esta aplicada o no
num_ruc	varchar	25	Código que identifica a la empresa
direccion	varchar	100	Dirección de la empresa
Teléfono	varchar	12	Teléfono de la empresa

**Tabla 8: cierre\_turno**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
Idcierre	autoincremento (int)	11	Código del cierre de turno
fecha_apertura	varchar	18	Fecha de apertura del turno
fecha_cierre	varchar	18	Fecha de cierre del turno
total_ventas	int	11	Total obtenido en ventas en el turno
total_compras	int	11	Total utilizado en compras durante el turno



Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

caja_inicial	double	16	Monto en caja al iniciar el turno
total_otrosingreso	int	11	Total obtenido de otros ingresos
total_otrosgastos	int	11	Total utilizado en otros gastos
ganancia	Double	16	Ganancia obtenida durante el turno
fecha	timestamp	18	Fecha actual del sistema
estado	boolean	1	Indica si se ha aplicado o no el cierre
idusuario	Int	11	Código del cliente que hace el cierre

**Tabla 9: compras**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idcompras	autoincremento (int)	11	Código de la Compra
idarticulos	int	11	Código del artículo a Comprar
marca	varchar	40	Marca del artículo a comprar
idproveedor	int	11	Código del Proveedor que vende el artículo
Cantidad	int	11	Cantidad de artículos a comprar
precio	double	16	Precio por artículo a comprar
Fecha	varchar	18	Fecha de la compra
idreg	int	11	Código del registro de la Compra
Total	double	16	Monto total de la compra



**Tabla 10: reg\_compras**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idreg	autoincremento (int)	11	Código del registro de la compra
Fecha	varchar	18	Fecha de la Compra
observaciones	text	65535	Observaciones acerca de la compra
nombre_usuario	varchar	25	Nombre del Usuario que realizo la compra
total	double	16	Total gastado en la compra
estado_registro	boolean	1	Indica si se realizó el registro o no

**Tabla 11: otros\_gastos**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
id_otrosgastos	autoincremento (int)	11	Código del registro de otros gastos
fecha	varchar	18	Fecha del registro
detalle	varchar	100	Detalle del registro de otros gastos
monto	double	16	Monto total de otros gastos
idcaja	int	11	Código de la caja del que se retiran otros gastos

**Tabla 12: otros\_ingresos**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
id_otrosingresos	int(autonumerico)	11	Código del registro de otros ingresos
fecha	varchar	18	Fecha del registro
detalle	varchar	100	Detalle del registro de otros ingresos
monto	double	11	Monto total de otros ingresos
idcaja	int	11	Código de caja en que se depositan otros ingresos

**Tabla 13: Caja**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idcaja	autoincremento (int)	11	Código de la Caja
caja_inicial	double	16	Monto inicial en Caja
fecha	varchar	18	Fecha en que se define el monto inicial de la Caja
nombre_usuario	varchar	25	Nombre del usuario que define la caja

**Tabla 14: ventas**

Nombre del Campo	Tipo de Datos	Longitud	Descripción
idventas	autonumerico(int)	11	Código de la Venta



fecha	Date	18	Fecha de la Venta
idarticulo	int	11	Código del articulo a vender
idcliente	int	11	Código del cliente al que se vende
cantidad	int	11	Cantidad del articulo a vender
precio	double	16	Precio por unidad del articulo
total_precio	double	16	Precio total de la Venta
idfactura	int	11	Código de factura

## 8.2. Diseño arquitectónico

A continuación se muestra la arquitectura que presentará el sistema, en la cual se emplea un modelo de N – Capas, eligiendo cuatro capas debido a que satisfacía nuestras necesidades de separar la lógica de negocios de los datos y también separar las entidades de la lógica de negocios.

### Capas de Datos.

Esta capa contiene la conexión con la base de datos y las transacciones realizadas a la misma.

### Capa de Entidades.

Contiene todas las clases mostradas en el diagrama de clases y todos los dataset. Donde los dataset son usados para generar los reportes y por lo tanto cada clase del sistema solo posee un método, ya que todos los demás métodos están contenidos dentro de los dataset.

### Capa de Negocios.

Contiene las operaciones de cálculo, validación y verificación.

### Capa de Presentación

Contiene todas las interfaces del sistema y las hojas de estilos, también todos los archivos de crystalreport.



### Diseño arquitectónico

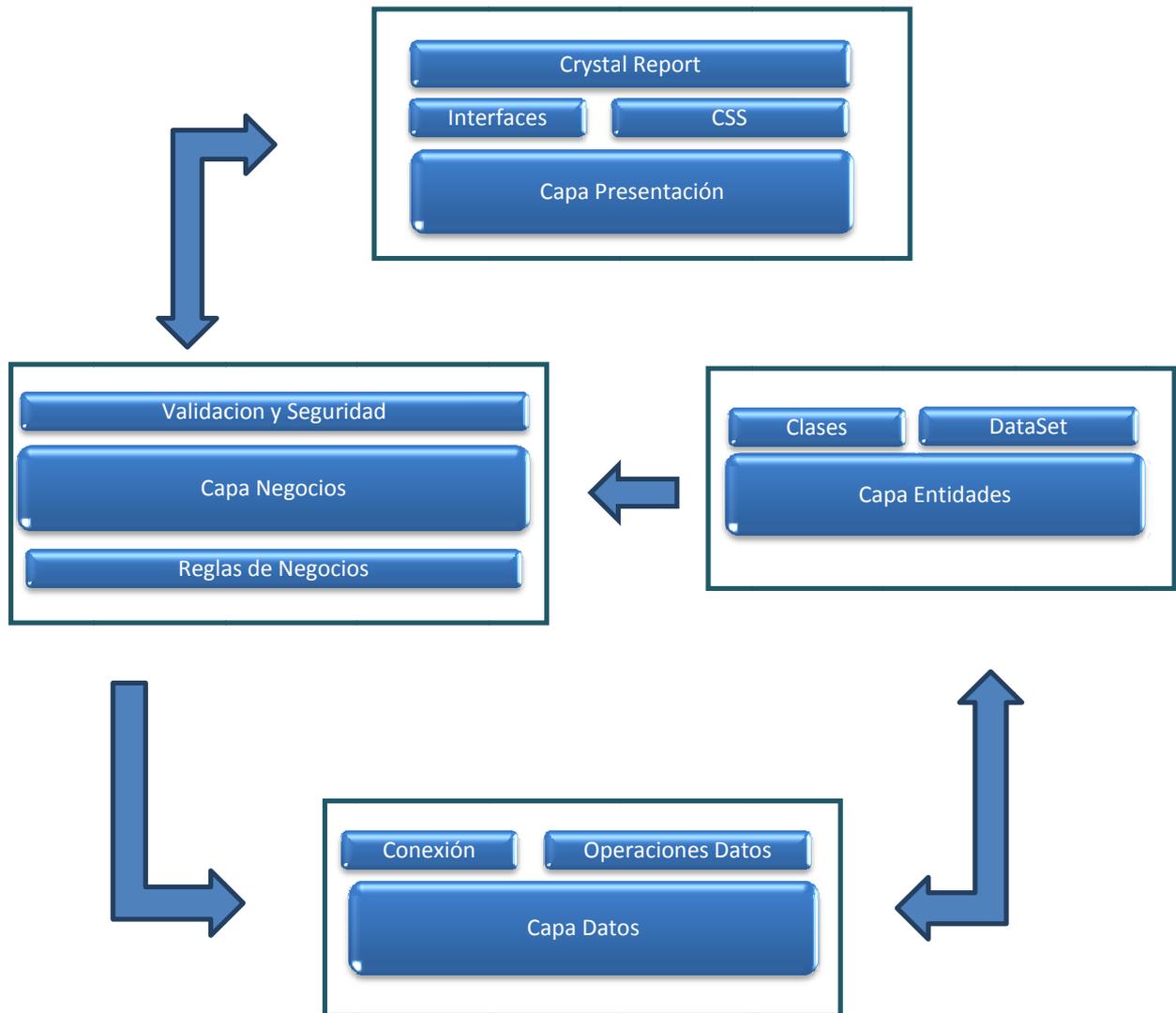


Figura19: Diseño Arquitectónico



### 8.3. Diseño de Interfaz

Esta interfaz se utiliza para el registro, ingreso al sistema o recuperación de la contraseña.

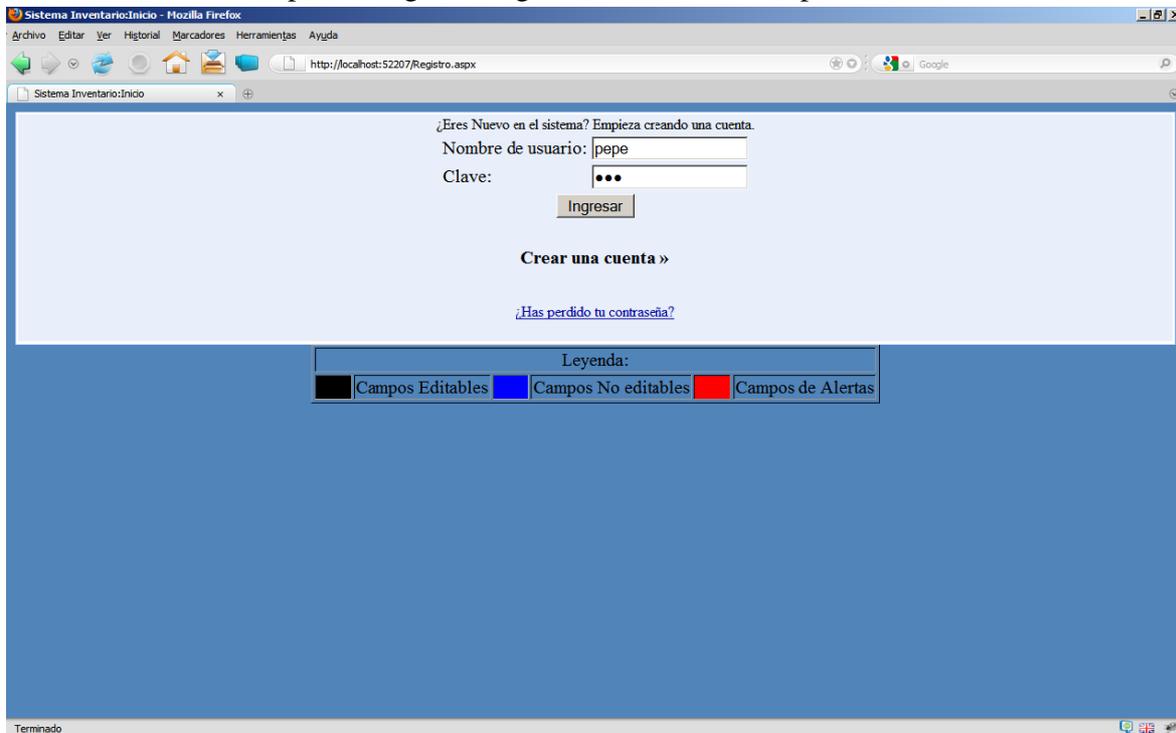


Figura20: Captura Ingreso al sistema

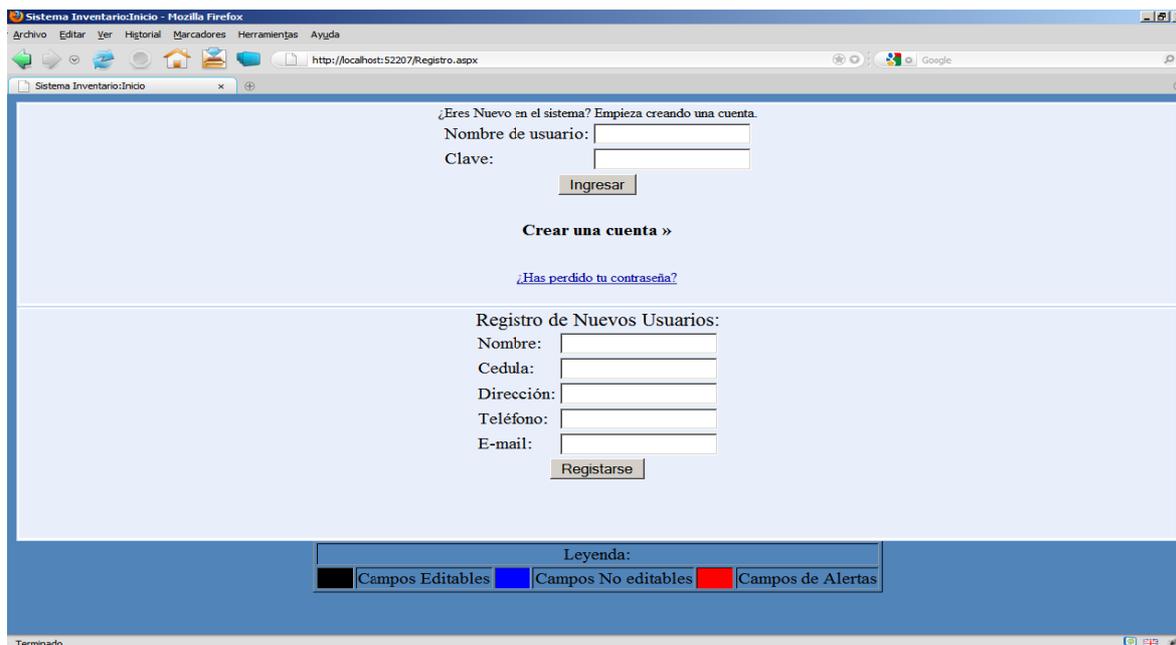
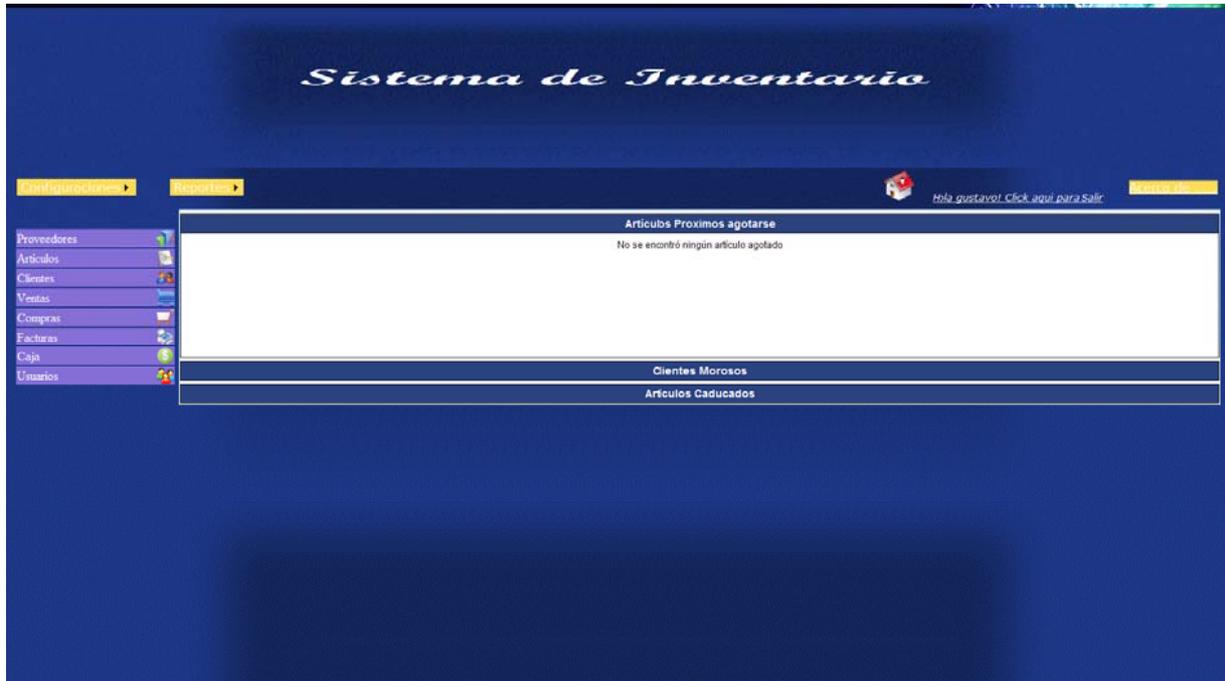


Figura 21: Captura de Registrarse en el sistema



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

Una vez dentro del sistema se muestra una pantalla inicial con información acerca de: artículos próximos agotarse, clientes morosos y artículos caducados.

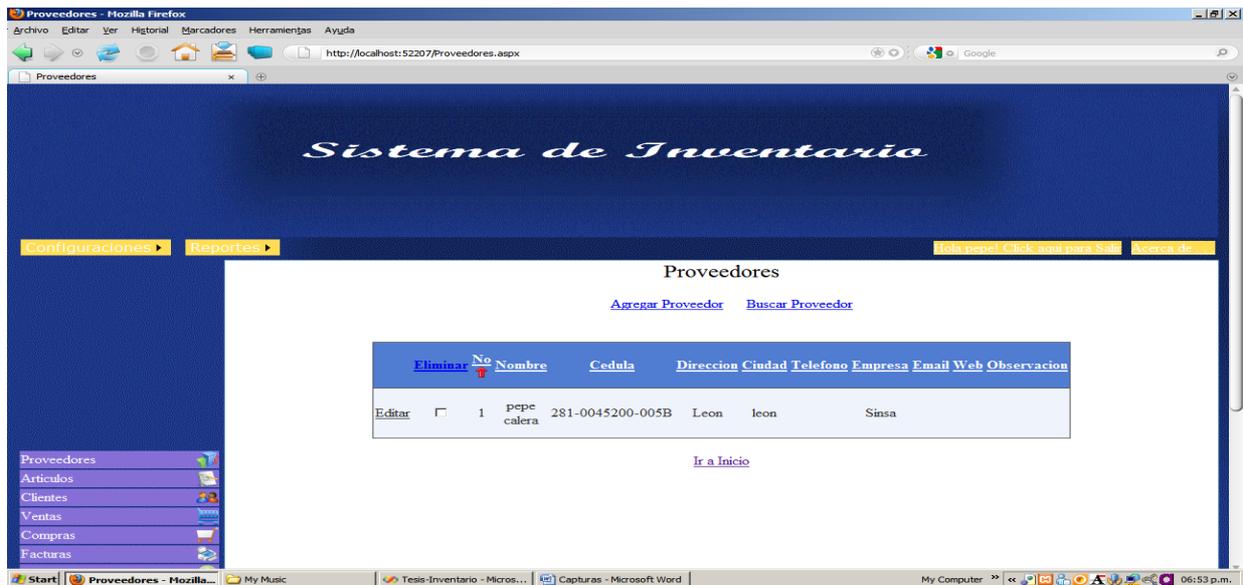


**Figura22: Captura de Pantalla de Inicio**



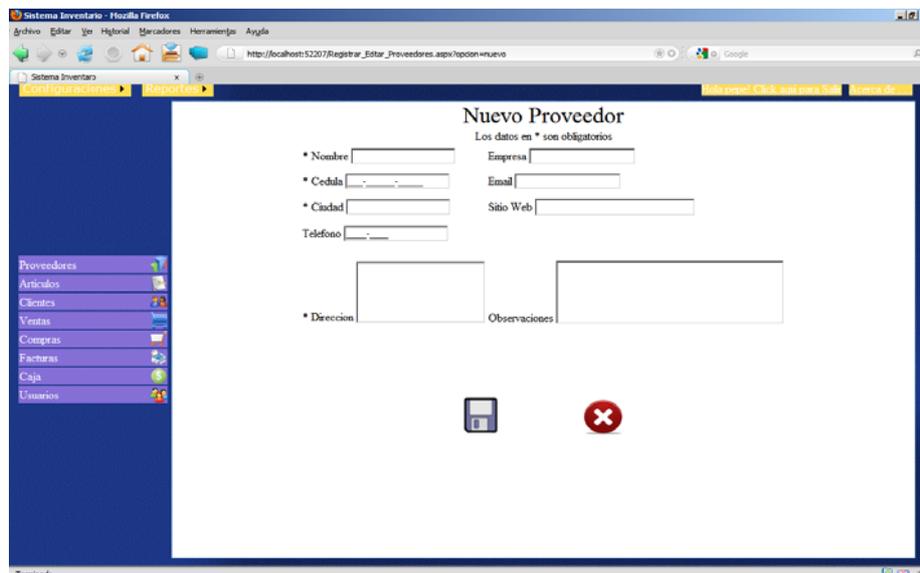
## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

Desde la interfaz anterior se puede acceder a cualquiera de las funciones del sistema, por ejemplo la de registro de proveedores. Dicha interfaz muestra información detallada de los proveedores, permite las opciones de editar, guardar, buscar y borrar.



**Figura23: Captura de Proveedores**

Dentro de la opción de agregar nuevo o editar proveedor podemos observar lo siguiente:



**Figura24: Captura de Agregar Proveedores**



Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

Otra función dentro de la interfaz de proveedor es la de realizar búsquedas que es invocada por el link “Buscar Proveedor“:



Figura25: Captura de Buscar Proveedores

La siguiente interfaz muestra el listado de todos los artículos y las opciones de agregar, buscar, ordenar, editar y borrar artículos.



Figura26: Capturas de Artículos



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

Una vez dentro de la interfaz artículos podemos agregar o editar alguno, como se muestra a continuación:

*Sistema de Inventario*

**Nuevo Artículo**  
Los campos con asteriscos son obligatorios!

**Datos Generales**

* Nombre	* Unidad de Medida	Unidad
* Estado	Bueno	Categoría
Proveedor	Ninguno	Marca

**Costos y Existencias**

* Precio por Unidad	Costo
Descuento	* Cantidad
Limite Existencias	

**Fechas**

Deshabilitar <input type="checkbox"/>	Fecha Adquisición	Fecha Caducidad
	08/12/2010	08/12/2010

Descripción

**Figura27: Captura de Agregar Artículos**



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

Al igual que la interfaz anterior, esta muestra un listado de todos los clientes y sus opciones de agregar, buscar, editar, borrar y ordenar.

*Sistema de Inventario*

Configuraciones ▶ Reportes ▶ [Ma para Click aquí para salir](#) [Verca de](#)

**Cientes**

[Agregar Cliente](#) [Buscar Cliente](#)

Eliminar	No	Nombre	Telefono	Direccion	Ciudad	Cedula	Email	Limite Credit	Tipo
Editar <input type="checkbox"/>	1	mario	2311-1555			222-222222-2222b		0	
Editar <input type="checkbox"/>	2	carlos montez	5555-5555		Leon	218-364796-6666n		0	

[Ir a Inicio](#)

Proveedores  
Articulos  
Clientes  
Ventas  
Compras  
Facturas  
Caja  
Usuarios

**Figura28: Captura de Clientes**



Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

Una de las interfaces más importantes de este sistema es la de ventas en la cual se pueden realizar al contado o al crédito.

**Figura29: Capturas de Ventas**

Para realizar las ventas al crédito basta con cambiar el tipo de factura y aparecerá una nueva interfaz como la siguiente:

**Figura30: Captura de Ventas al Crédito**

Una interfaz útil es la de “cambio” que sirve como calculadora para calcular el vuelto que se le deba de dar al cliente, este puede ser en dos tipos de moneda (córdoba y dólar).

**Figura31: Captura de Calcular Cambio**



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

A continuación se muestra la factura generada por la venta realizada:

The screenshot shows a web browser window with the following content:

**Ferretería La Bombina**

Dirección: Super mercado salinas 21124 al norte      Teléfono: 2311-0892

No. Rec: 475-889      Fecha: 12/12/10      No. Factura: 7

Vendedor: pepe      Tipo Factura: Comprobante      Nombre Cliente: none

Detalle	Descripcion	Cantidad	Valor	Impuesto	Total
Totales:		1	40.00	0.00	40.00

Subtotal:	40.00
IVA:	0.00
Total:	40.00

**Figura32: Captura de Factura Generada**

Otra de las interfaces importantes es la de compras, que a la vez permite agregar un nuevo artículo y proveedor, y ver la información de los artículos.

The screenshot shows the 'Sistema de Inventario' web application interface. It includes a navigation menu on the left with options like Proveedores, Artículos, Clientes, Ventas, Compras, Facturas, Caja, and Usuarios. The main area displays a purchase form with the following fields:

Artículos:         Proveedor:

Cantidad:       Precio:

ID	Nombre	Marca	Categoría	Precio U	Costo	Existencias	Unidad Medida	Descuento	Estado	Fecha Caducidad	Descripción	Fecha Adquisición	ID proveedor
2	martillo	stanley		1000	100	31	unidad	5	bueno			2010-07-12	1
3	artic			1	0	111111	Unidad	0	bueno	25/11/2010		25/11/2010	0
4	Martillos	especial		200	100	100	Docena	10	bueno	08/12/2010		08/12/2010	1

Observaciones:       Total:

Comprador: pepe      Fecha: 8/12/2010

**Figura33: Captura de Compras**



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

En la siguiente interfaz podemos ver un listado de las facturas realizadas y las operaciones que se pueden realizar.

No	Fecha	Observaciones	Total	Tipo	Estado	Vendedor
22	08/12/2010		0	Contado	Vigente	pepe

Figura34: Capturas de Facturas

Para realizar un pago de una factura al crédito se utiliza la siguiente interfaz siempre y cuando haya facturas pendientes:

**Pagar Facturas**

No Factura Pendiente:  Fecha: 08/12/2010 Cliente:

Monto Inicial:

No Cuotas:  Monto Cuota:

Mora:  Monto a Pagar:

Por Pagar:  Total Pagado:

**Pago**

Efectivo:  Moneda:

Cambio:

Figura35: Captura de Pago de Facturas



Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

La interfaz de caja permite ver las ganancias, realizar los cierres de turno y añadir gastos extras y ganancias extras.

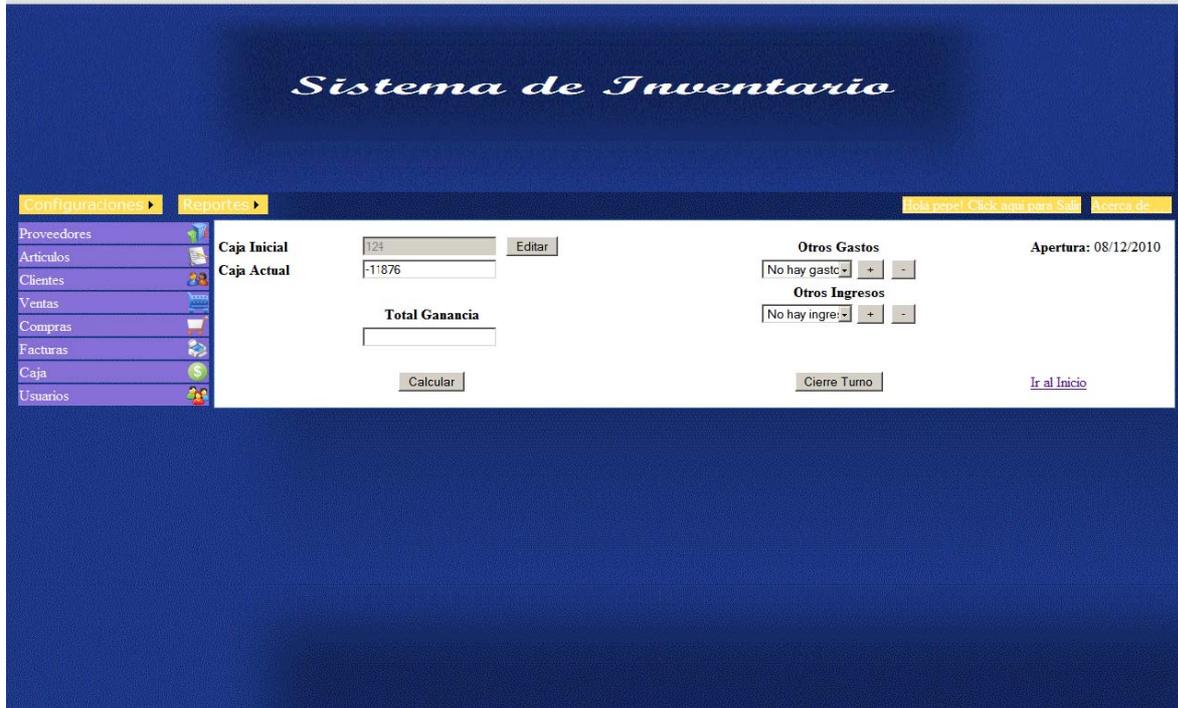


Figura36: Captura de Caja

Y por último la interfaz de usuario que permite editar, buscar y eliminar usuario. Pero a la vez a tiene la opciones de editar nuestra propia cuenta y subir de nivel a otros usuarios.

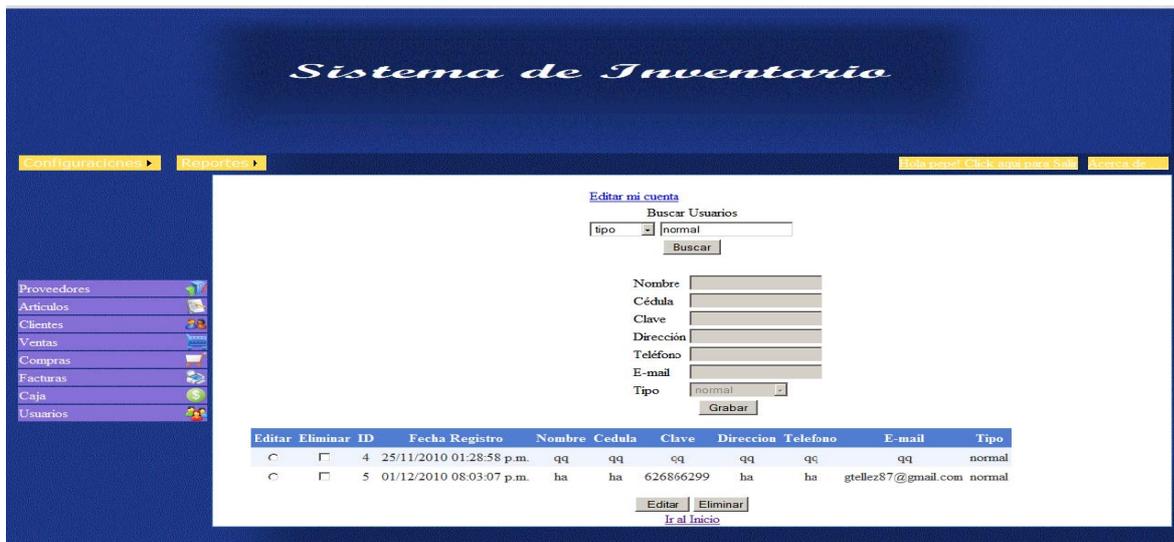


Figura37: Captura de Usuarios



## 8.4. Programación

En esta sección se muestra porciones de códigos que destacan dentro del Sistema. A continuación se muestra el cálculo de las cuotas quincenales o mensuales:

```
public string calculo_fecha_fin(string tiempo_c, bool rdmen, bool rdquin, string fech)
{
    string fecha = ""; int dia_act, mes_act, anyo_act, df = 1, mf = 1, yf = 0;
    dia_act = valor_fecha(1, fech);
    mes_act = valor_fecha(2, fech);
    anyo_act = valor_fecha(3, fech);
    if (rdmen)
    {
        int i = 0;
        while (i < (int.Parse(tiempo_c)))
        {
            if (mf < 12)
            {
                mf = ++mes_act;
                yf = anyo_act;
            }
            else
            {
                mf = 1;
                mes_act = 1;
                yf = ++anyo_act;
            }
            i++;
        }
        fecha = dia_act.ToString() + "/" + mf.ToString() + "/" + yf.ToString(); ;
    }
    elseif (rdquin)
    {
        int num_quincena = 0;
        int num_día = 0;
        while (num_quincena < (int.Parse(tiempo_c)))
        {
            switch (mes_act)
            {
                case 01:
                case 03:
                case 05:
                case 07:
                case 08:
                case 10:
                case 12:
                if (df < 31)
                {
                    if (num_día == 15)
                    {
                        num_quincena++;
                        num_día = 0;
                    }
                }
            }
        }
    }
}
```



```
num_dia++;
    df = ++dia_act;
    yf = anyo_act;
    mf = mes_act;
}
elseif (mf == 12)
{
    mf = 1;
    mes_act = 1;
    yf = ++anyo_act;
}
else
{
    dia_act = 1;
    df = dia_act;
    mf = ++mes_act;
}
break;
case 04:
case 06:
case 09:
case 11:
if (df < 30)
{
if (num_dia == 15)
{
    num_quincena++;
    num_dia = 0;
}
num_dia++;
df = ++dia_act;
    yf = anyo_act;
    mf = mes_act;
}
elseif (mf == 12)
{
    mf = 1;
    mes_act = 1;
    yf = ++anyo_act;
}
else
{
    dia_act = 1;
    df = dia_act;
    mf = ++mes_act;
}
break;
```



```

                                case 02:
if (df < 28)
                                {
if (num_dia == 15)
                                {
                                num_quincena++;
                                num_dia = 0;
                                num_dia++;
df = ++dia_act;
                                yf = anyo_act;
                                mf = mes_act;
                                }
elseif (mf == 12)
                                {
                                mf = 1;
                                mes_act = 1;
                                yf = ++anyo_act;
                                }
else
                                {
                                dia_act = 1;
                                df = dia_act;
                                mf = ++mes_act;
                                }
break;
                                }
                                }
fecha = df.ToString() + "/" + mf.ToString() + "/" + yf.ToString();
return fecha;
}
```



El siguiente segmento de código muestra el cálculo del cierre de turno:

```
public void Calcular_Cierre_Turno(int idcaja, string caja_inicial, int idusuario)
{
    float contado = 0;
    float credito = 0;
    float total_venta = 0;
    float total_comp = 0;
    float total_gastos = 0;
    float total_ingresos = 0;
    string f = "";
    cierre nuevo_cierre = new cierre();
    fact = operdatos.ObtenerFacturas();
    for (int i = 0; i < fact.Length; i++)
    {
        if (fact[i].estado_reg == 0 && fact[i].estado != "Cancelada")
        {
            contado += fact[i].total;
        }
    }
    es = operdatos.consultaEstado_cuenta();
    for (int i = 0; i < es.Length; i++)
    {
        credito += es[i].monto_pagado * es[i].num_cuota;
    }
    total_venta = contado + credito; //total_ventas()

    reg = operdatos.consultaRegCompras_ER0();

    for (int i = 0; i < reg.Length; i++)
    {
        total_comp += reg[i].total;
    }
    o_gastos = operdatos.obtener_otros_gastos(idcaja);
    for (int i = 0; i < o_gastos.Length; i++)
    {
        total_gastos += o_gastos[i].monto;
    }
    o_ingresos = operdatos.obtener_otros_ingresos(idcaja);
    for (int i = 0; i < o_ingresos.Length; i++)
    {
        total_ingresos += o_ingresos[i].monto;
    }
    f = operdatos.fecha_apertura();
    if (operdatos.verificar_cierre())
    {
        return;
    }
    nuevo_cierre.fecha_apertura = f; nuevo_cierre.fecha_cierre = DateTime.Today.ToShortDateString();
    nuevo_cierre.total_ventas = total_venta; nuevo_cierre.total_compras = total_comp;
    nuevo_cierre.caja_inicial = float.Parse(caja_inicial); nuevo_cierre.total_otrosingresos = total_ingresos;
    nuevo_cierre.total_otros_gastos = total_gastos; nuevo_cierre.ganancia = float.Parse(calcular_ganancia());
    nuevo_cierre.idusuario = idusuario; nuevo_cierre.estado = false; operdatos.InsertarCierre(nuevo_cierre);
    operdatos.actualizar_estado();
}
```



Este código permite eliminar múltiples artículos de un gridview:

```
protectedvoid Eliminar_Click(object sender,EventArgs e)
{
    bool cargar = false;
    Inicio();
    System.Web.UI.WebControls.CheckBox chk = null;
    for (int i = 0; i < GridArticulos.Rows.Count; i++)
    {
        chk =
        (System.Web.UI.WebControls.CheckBox)GridArticulos.Rows[i].Cells[1].FindControl("chk");
        if (chk.Checked)
        {
            reglas.BorrarArticulo(GridArticulos.Rows[i].Cells[2].Text);
            cargar = true;
        }
    }
    if (cargar == true)
    {
        reglas.ArticulosOrdenados(Session["campoarto"].ToString());
        CargarDatos();
    }
}
```



Este código es un ejemplo de cómo está estructurada una entidad:

```
namespace CapaEntidades.cls
{
publicclass articulos
{
publicint num_articulo;
publicstring nombre;
publicstring marca;
publicstring categoria;
publicfloat precio;
publicfloat costo;
publicint existencias;
publicstring unidad;
publicfloat descuento;
publicstring estado;
publicstring caducidad;
publicstring descripcion;
publicstring adquisicion;
publicint num_proveedor;
publicint limit_exis;
publicvoid setarticulo(int n_a, string nomb, string marc, string cat, float p,
float cost, int exist, string und, float desct, string est, string cad, string
desc, string adq, int n_p, int lim_ex)
{
    num_articulo = n_a;
    nombre = nomb;
    marca = marc;
    categoria = cat;
    precio = p;
    costo = cost;
    existencias = exist;
    unidad = und;
    descuento = desct;
    estado = est;
    caducidad = cad;
    descripcion = desc;
    adquisicion = adq;
    num_proveedor = n_p;
limit_exis = lim_ex;
}
}
}
```



## IX. Evaluaciones

Se presentan los resultados de las pruebas realizadas al sistema, los equipos y los programas utilizados en ellas.

Características de las Pc's:

Clientes:

- ✓ Acer Aspire One D250. .
  - CPU:1.6Ghz . Intel Atom
  - Ram: 1 GB.
- ✓ Toshiba Satellite
  - CPU:1.6Ghz. Intel CoreDuo.
  - Ram: 1GB.

Servidor:

- ✓ Sony Vaio.
  - CPU:2.2Ghz. Dual Core 2 Duo.
  - Ram: 4 GB.

Herramienta Software para medir el Rendimiento.

- ✓ Internal Monitoring in Monitis (mon.itor.us)
- ✓ Mozilla Firefox 3.6.13.

Los resultados obtenidos para la realización de las operaciones en los distintos escenarios que se muestran a continuación.

No. Usuarios	Escenario	Operación	Tiempo de Respuesta(seg)
3	1	Registro Simultaneo	4
3	2	Ingreso Simultaneo al Sistema	1
3	3	Guardado Simultaneo de datos	0.5
3	4	Paralelo(Borrado, Guardado, Editado)	0.5
3	5	Búsquedas y Reportes Simultaneo	1
3	6	Respaldo Simultaneo	0.5
		Promedio de Respuesta	1.25



## X. Conclusiones

Para concluir podemos decir, que se han cumplidos los objetivos planteados en la elaboración de este proyecto, de la siguiente manera:

- ✓ Se construyó un sistema con la posibilidad de generar reportes de la información que el usuario deseaba (artículos, compras, ventas, facturas, clientes, proveedores), brindándole la ventaja de poder imprimir dichos reportes y dando como solución a nuestro primer objetivo.
- ✓ Se logró desarrollar una aplicación con alto rendimiento de respuesta ante las operaciones de los usuarios, esto se puede comprobar en la sección evaluaciones de este documento, donde se obtuvo como rendimiento promedio 1.25 segundos para todos los escenarios planteados.
- ✓ Se controló el flujo de entrada y salida de dinero, independientes a la compra y venta, agregando las tablas de otros gastos y otros ingresos a la base de datos permitiendo registrar cualquier operación relacionado al efectivo.
- ✓ Se implementó el Protocolo seguro de transferencia de hipertexto (HTTPS) para establecer la seguridad entre las transacciones y también se utilizó el algoritmo de encriptación AES para las contraseñas de usuarios en conjunto a la contraseña de la base de datos.



## **XI. Recomendaciones**

- Es necesario capacitar a los usuarios en la seguridad de aplicaciones web para evitar cualquier pérdida de información o exposición de la misma.
- Es importante realizar respaldos de la información de la base de datos de forma periódica.
- Instalar el sistema en un servidor web, de forma que pueda ser accedido desde cualquier maquina con acceso a este.
- Una vez instalada en internet utilizar servicios web para realizar de forma automática algunas configuraciones del sistema (tasa de cambio).



## XII. Bibliografía

- Microsoft. (2009). Recuperado el 14 de Septiembre de 2010, de <http://www.microsoft.com/NET>
- Ceballos, F. J. (2007). *Enciclopedia de C# 2da Edición*. Alfaomega.
- Cesar de la Torre, U. Z. (2010). *Guía de Arquitectura N-Capas orientada al Dominio con .Net 4.0*. Krasis Consulting.
- Cuellar, J. (02 de agosto de 2010). *josecuellar.net*. Recuperado el 15 de octubre de 2010, de [josecuellar.net: www.josecuellar.net](http://josecuellar.net)
- Der Henst, C. V., Falla Aroche, S., & Hernández, D. (s.f.). *Maestros del web*. Obtenido de Maestros del web: <http://www.maestrosdelweb.com/editorial/ajax/>
- es.Wikipedia.org. (18 de Octubre de 2010). *Wikipedia*.
- Garret, J. J. (s.f.). *Ajax: A New Approach to Web Applications*. Obtenido de <http://adaptivepath.com/ideas/essays/archives/000385.php>
- Greene, J. H. (2001). *Control de la Producción*. Diana.
- Microsoft. (2009). *Microsoft*. Recuperado el 10 de Septiembre de 2010, de Microsoft: <http://www.microsoft.com/net/>
- Paré, R. C., Casillas Santillán, L. A., Costa Costal, D., Ginesta, M. G., Escofet, C. M., & Mora, P. O. (2005). *Bases de datos*. Barcelona: Eureka Media, SL.
- Rodriguez, M. V. (s.f.). *AJAX y ASP.NET*.
- Sommerville, I. (2005). *Ingeniería del Software* (Septima ed.). Madrid: Pearson Educacion.
- Tersine, R. J. (1999). *Principios de Inventario y manejo de materiales*. North-Holland.
- Vozes, P. (2004). *Teoría General de Sistemas*.
- Wallace B. McClure, S. C. (2006). *Begining Ajax with Asp.Net*.
- Wikipedia. (10 de 10 de 2010). *wikipedia.org*. Recuperado el 11 de 10 de 2010, de [http://es.wikipedia.org/wiki/Base\\_de\\_datos](http://es.wikipedia.org/wiki/Base_de_datos)
- Wikipedia. (6 de Octubre de 2010). *Wikipedia.org*. Recuperado el 11 de Octubre de 2010, de [http://es.wikipedia.org/wiki/Sistema\\_de\\_gesti%C3%B3n\\_de\\_bases\\_de\\_datos](http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos)



Aplicación Web para Automatizar la administración e inventario del negocio el  
Chompipe en el mercado el BISNE.

---

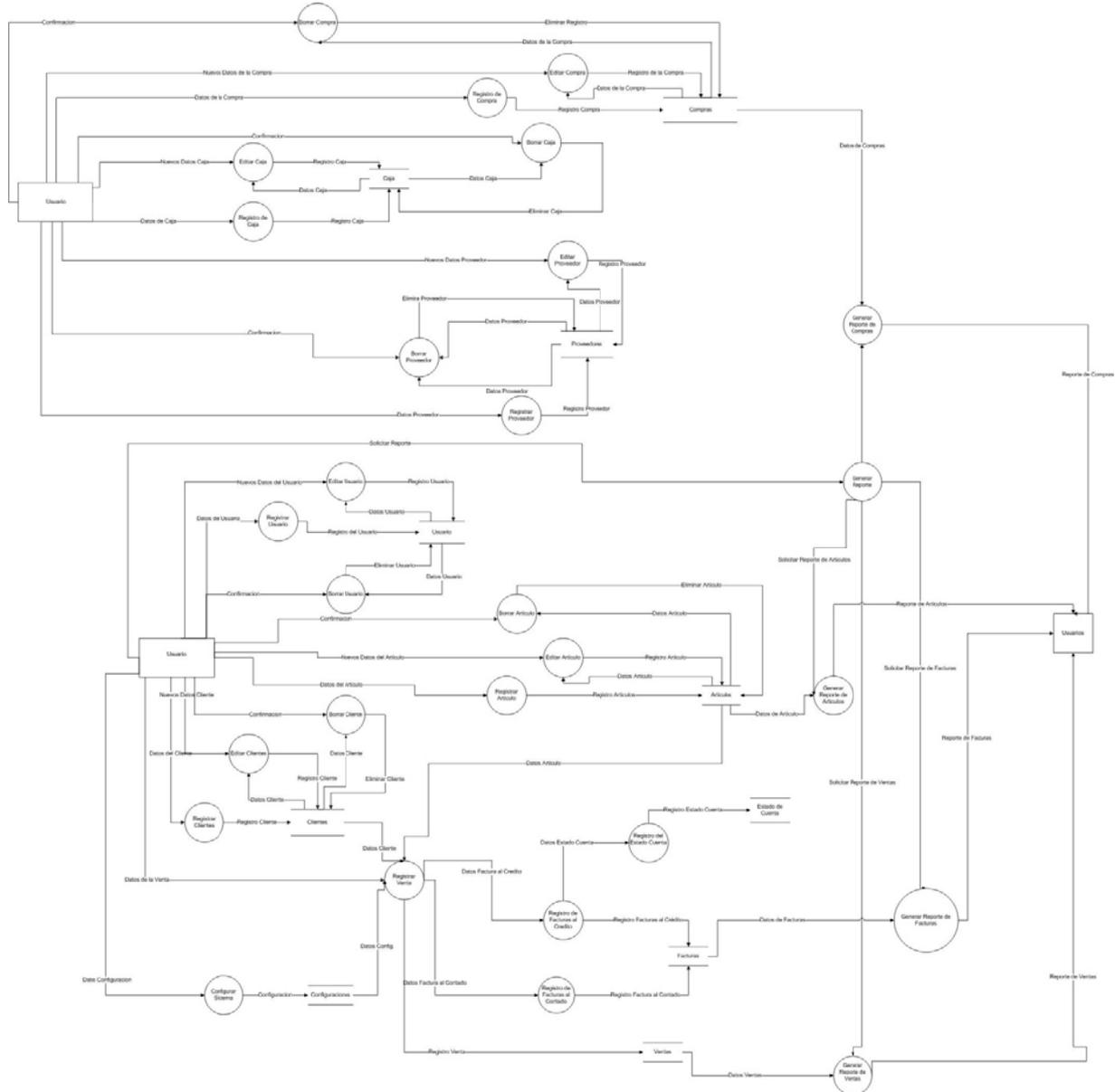
Wikipedia. (20 de Septiembre de 2010). *Wikipedia.org*. Recuperado el 12 de Octubre de 2010,  
de [http://es.wikipedia.org/wiki/Servidor\\_web](http://es.wikipedia.org/wiki/Servidor_web)



# Aplicación Web para Automatizar la administración e inventario del negocio el Chompie en el mercado el BISNE.

## XIII. ANEXOS

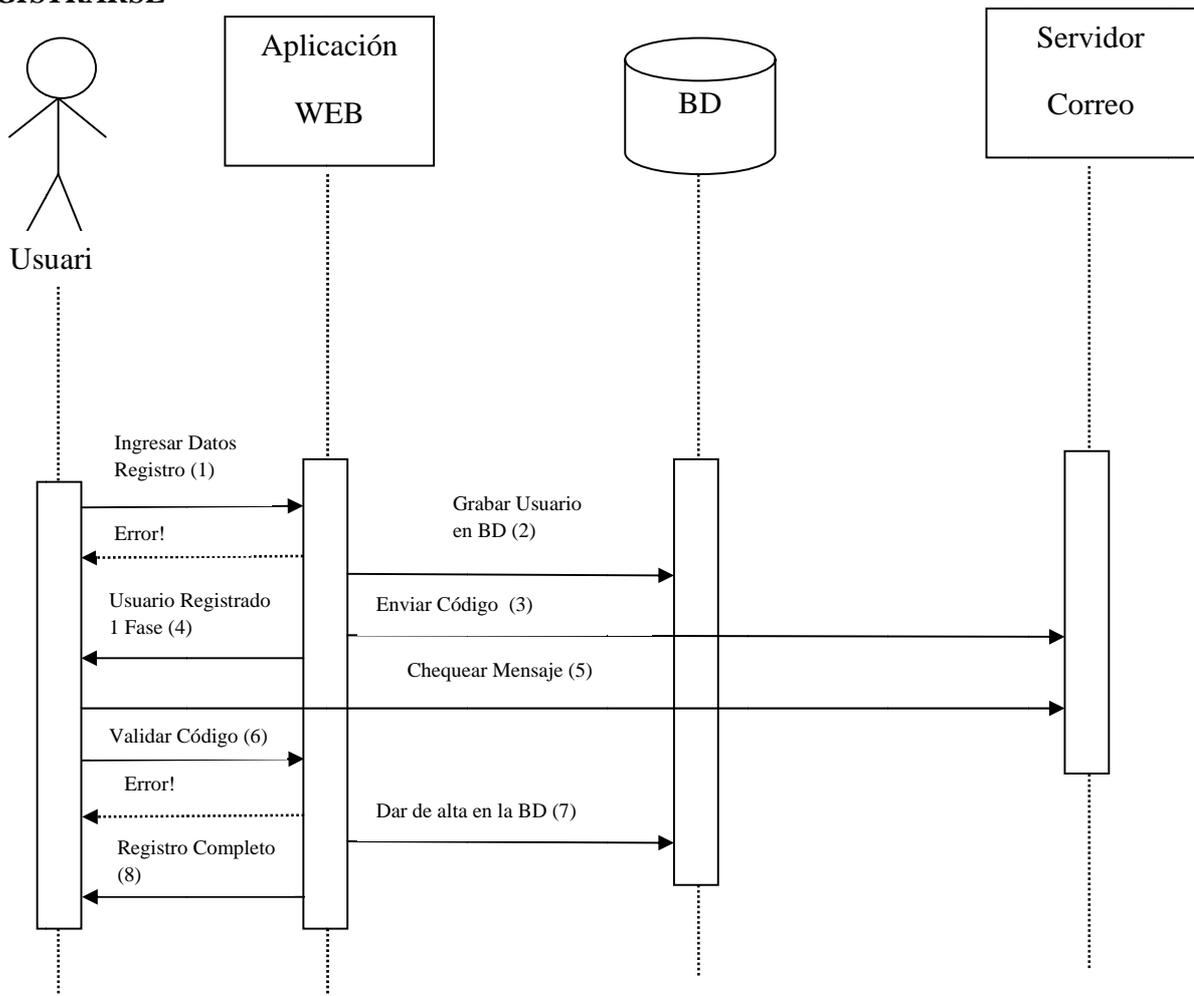
### 13.1. Diagrama de Flujo de datos





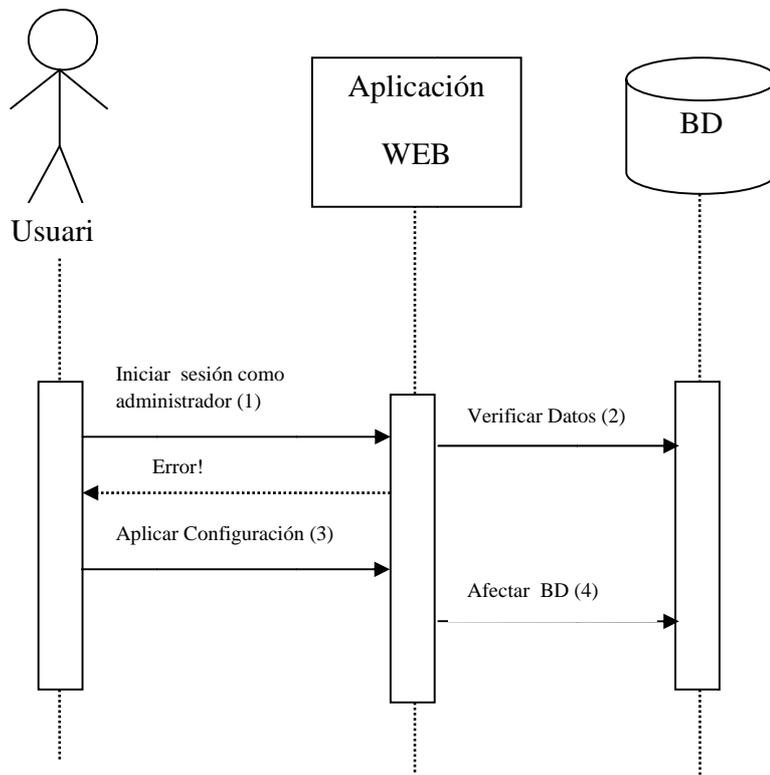
### 13.2. Diagramas de Secuencia

#### REGISTRARSE

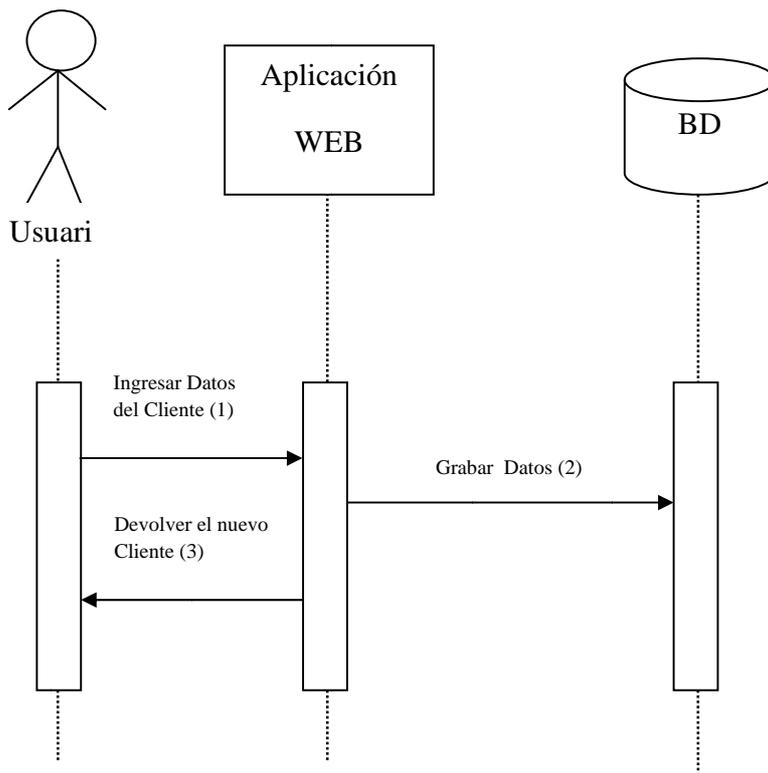




### CONFIGURAR SISTEMA

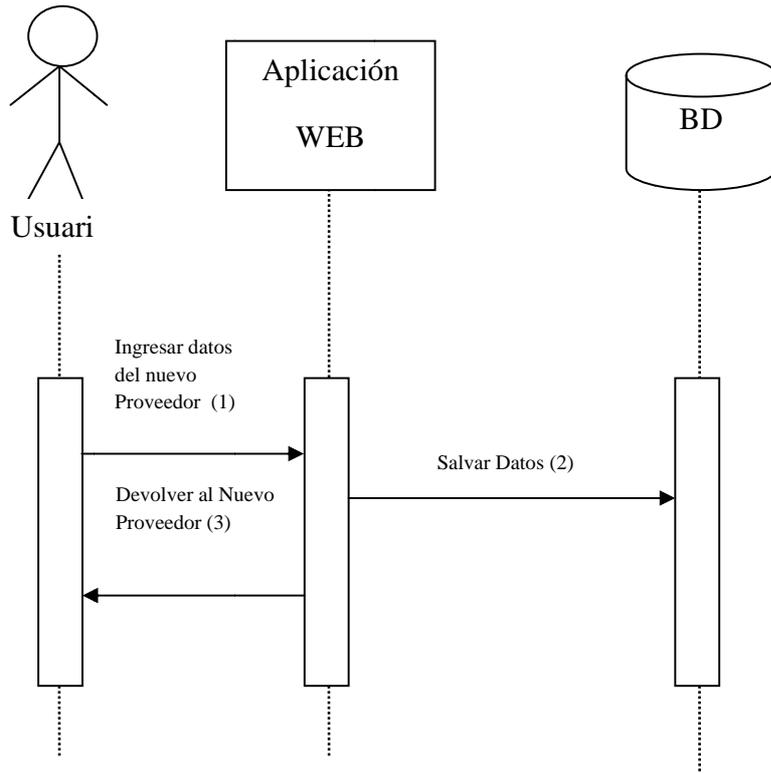


### REGISTRAR CLIENTES

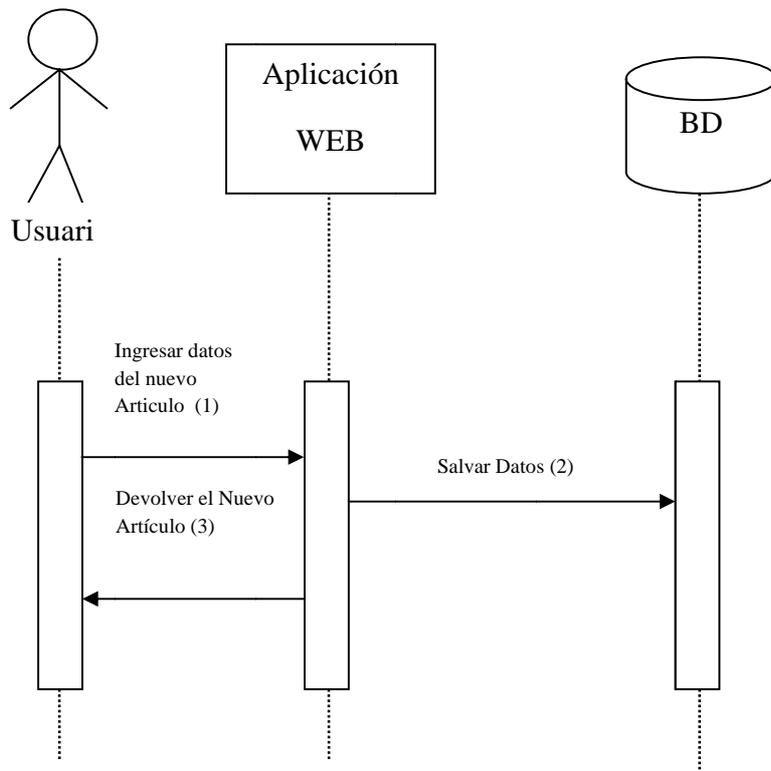




### REGISTRAR PROVEEDOR

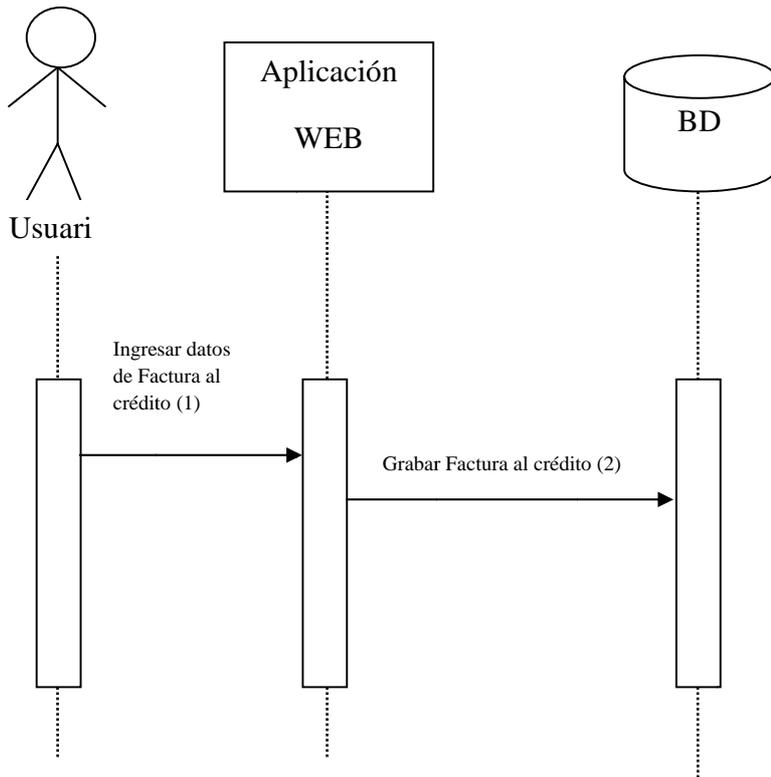


### REGISTRAR ARTÍCULO

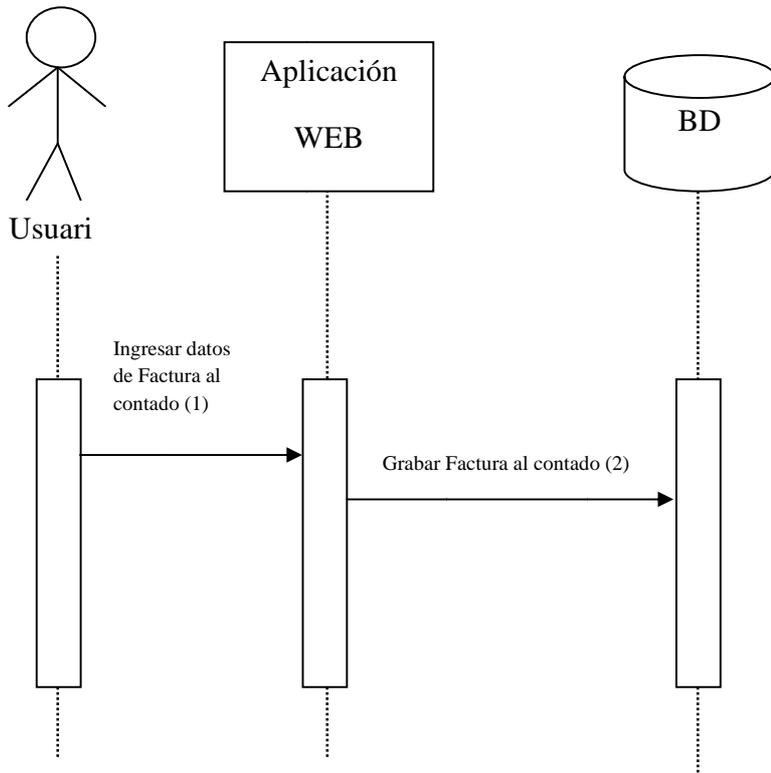




### REGISTRAR FACTURAS AL CREDITO

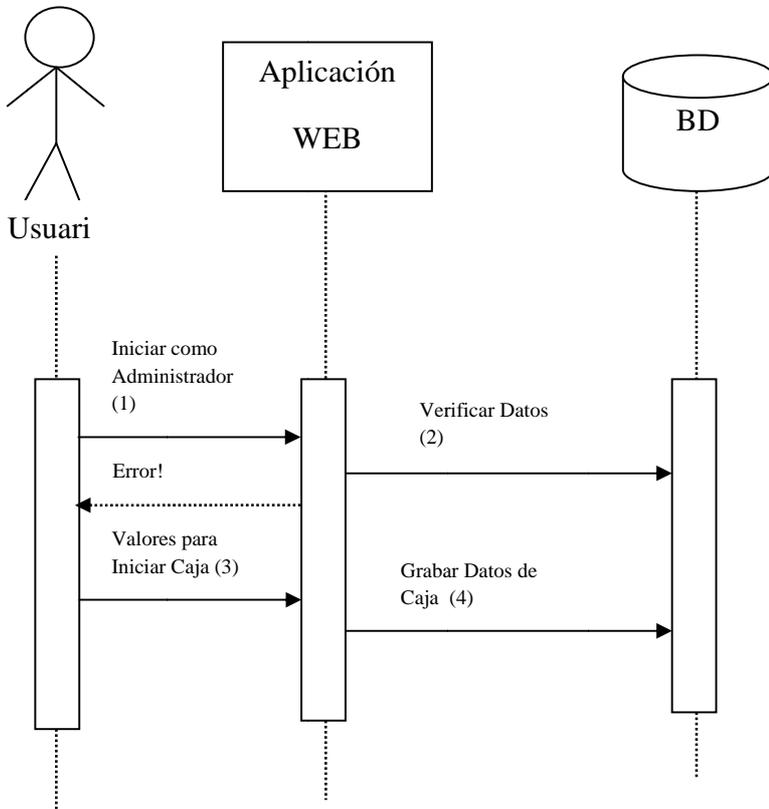


### REGISTRAR FACTURAS AL CONTADO

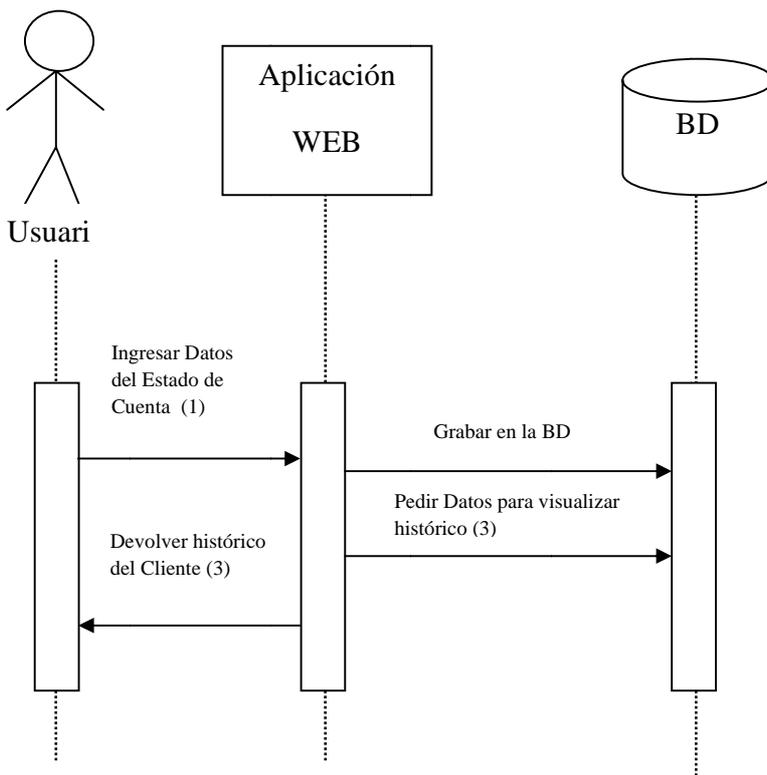




### REGISTRAR CAJA

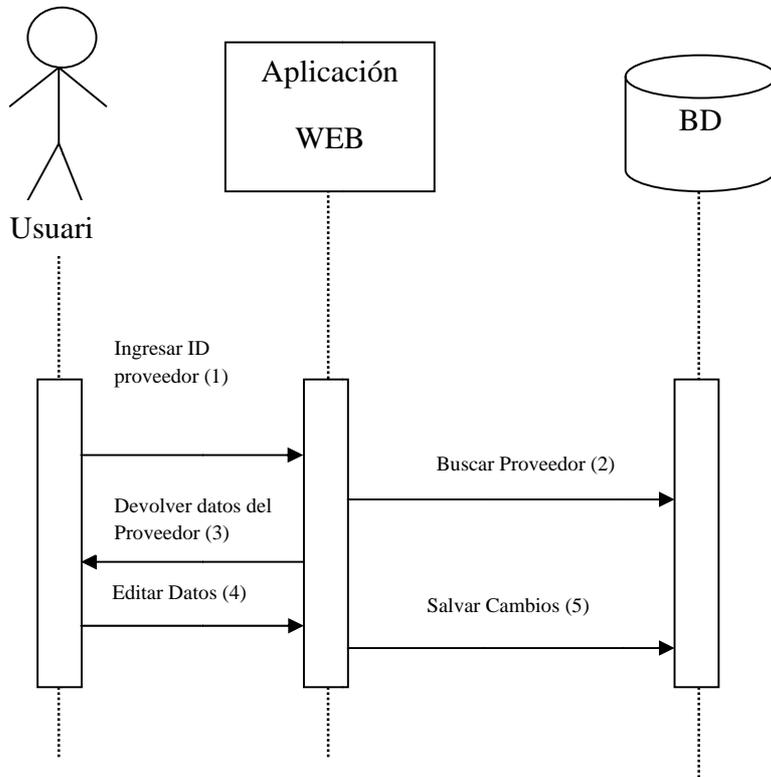


### REGISTRAR ESTADO DE CUENTA DE CADA CLIENTE

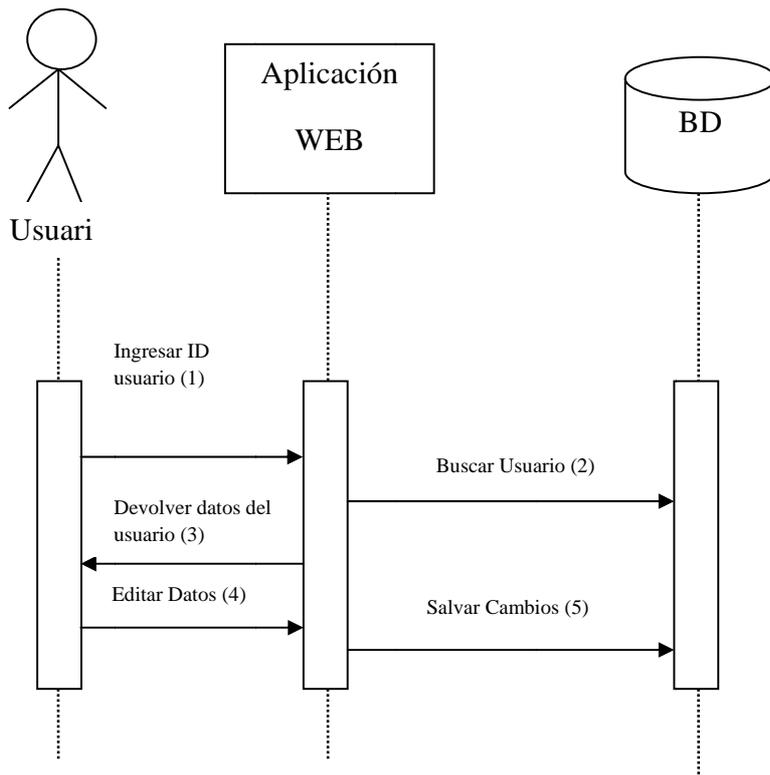




### **EDITAR PROVEEDOR**

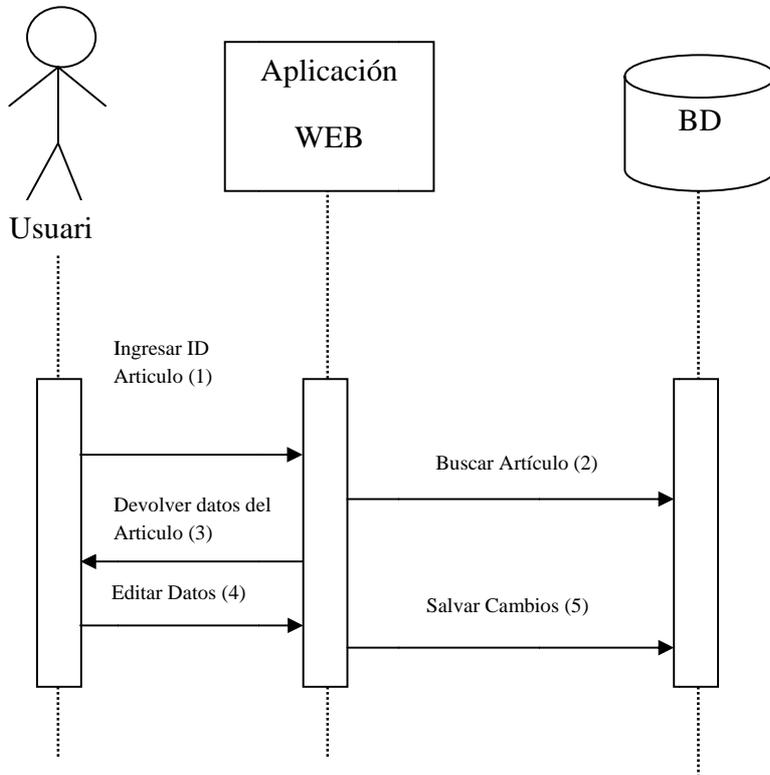


### **EDITAR USUARIOS**

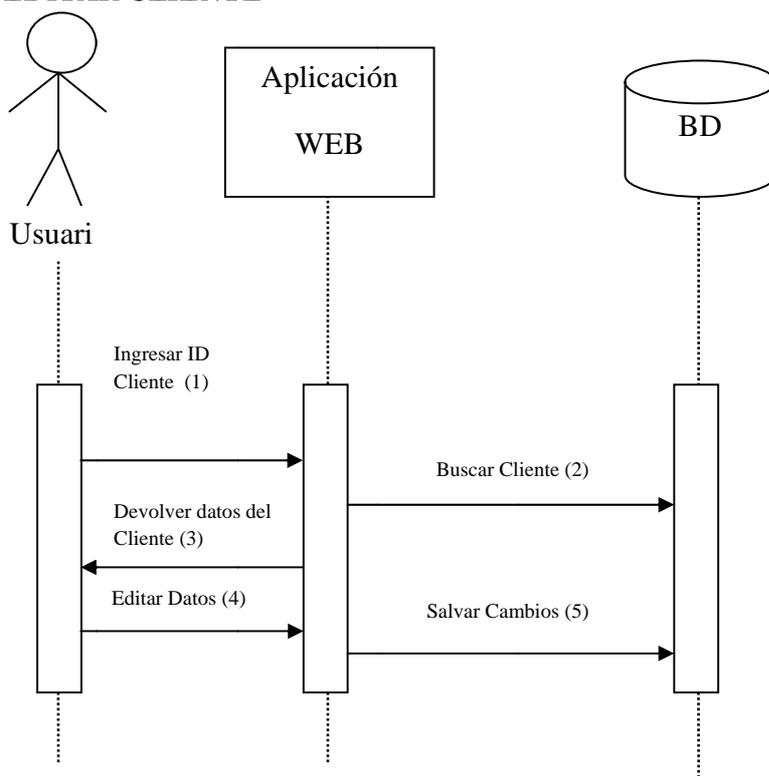




### EDITAR ARTICULOS

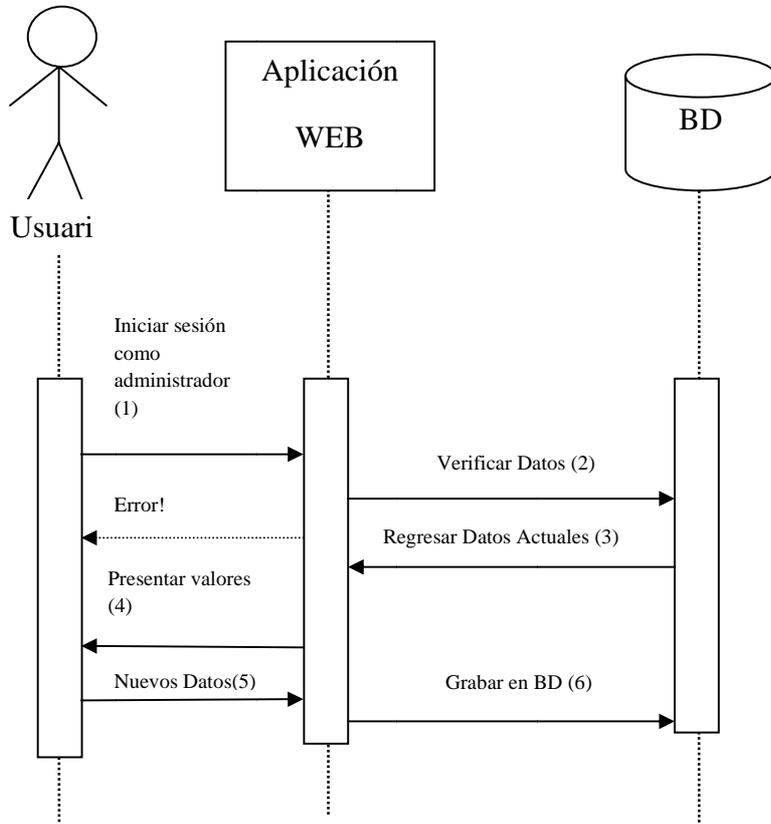


### EDITAR CLIENTE

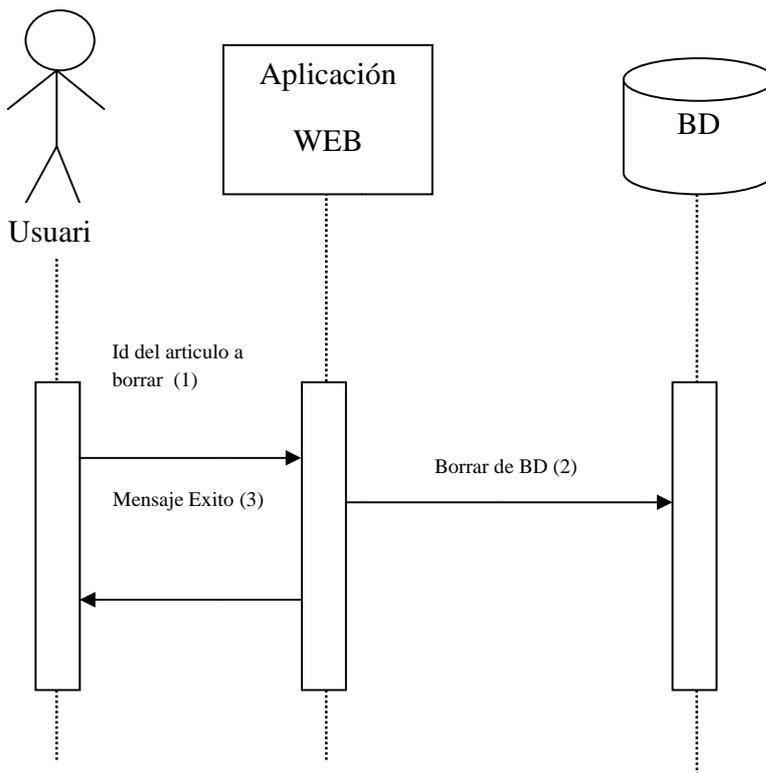




### EDITAR CAJA

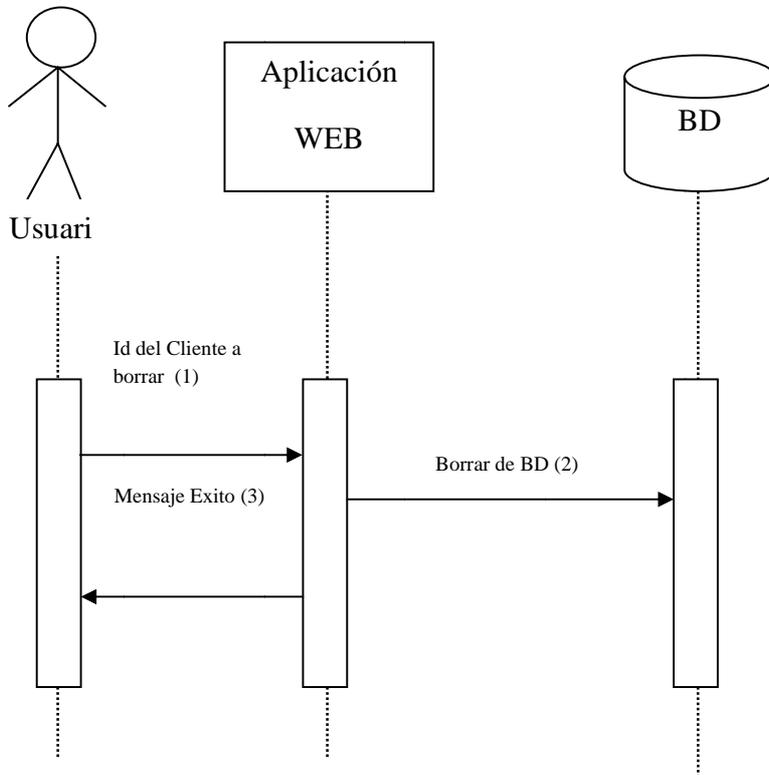


### BORRAR ARTÍCULO

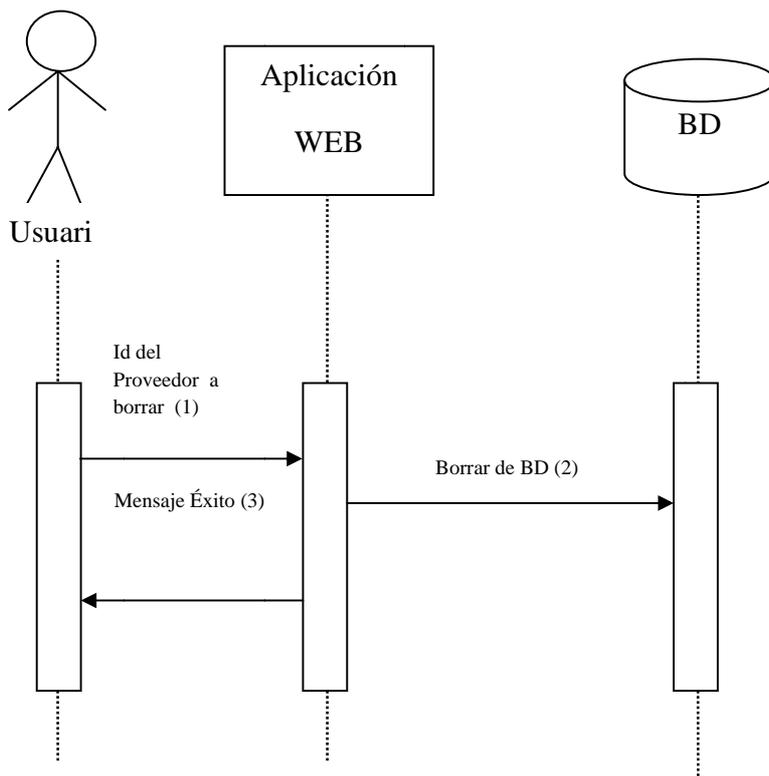




### ***BORRAR CLIENTES***

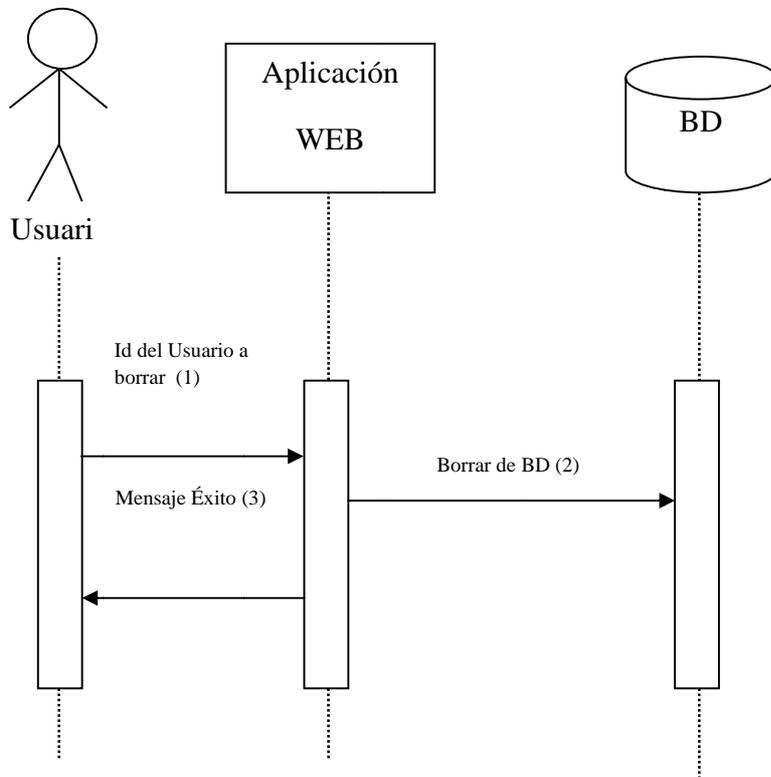


### ***BORRAR PROVEEDOR***

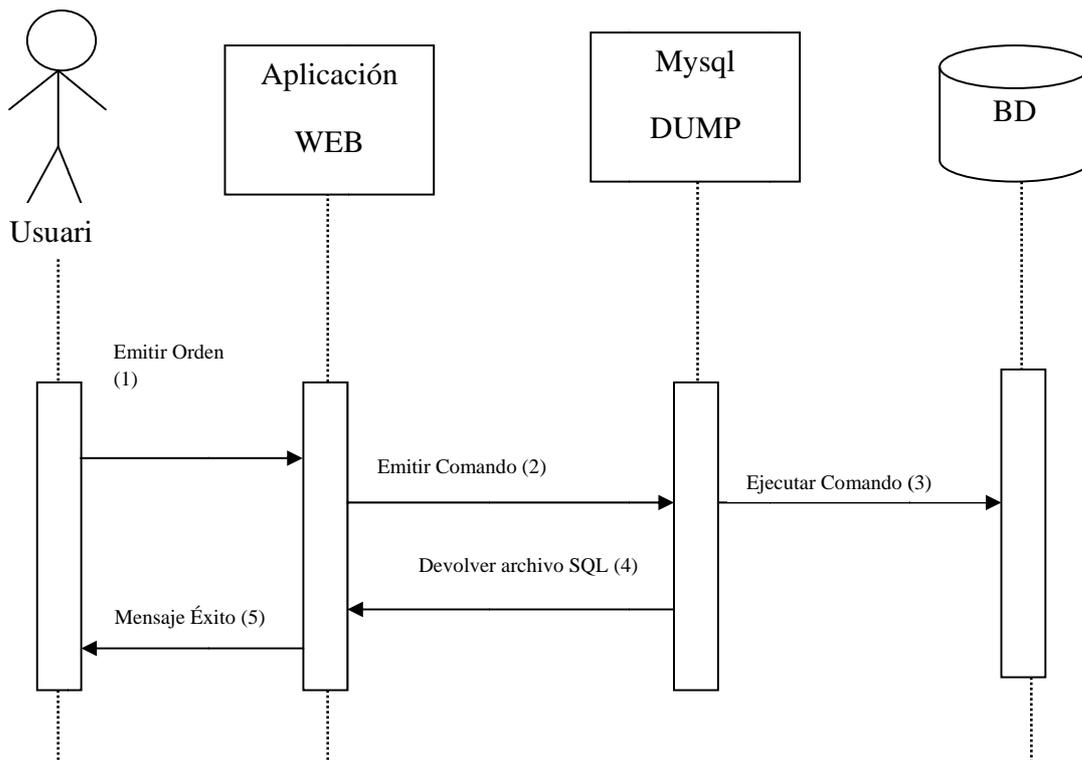




### **BORRAR USUARIO**

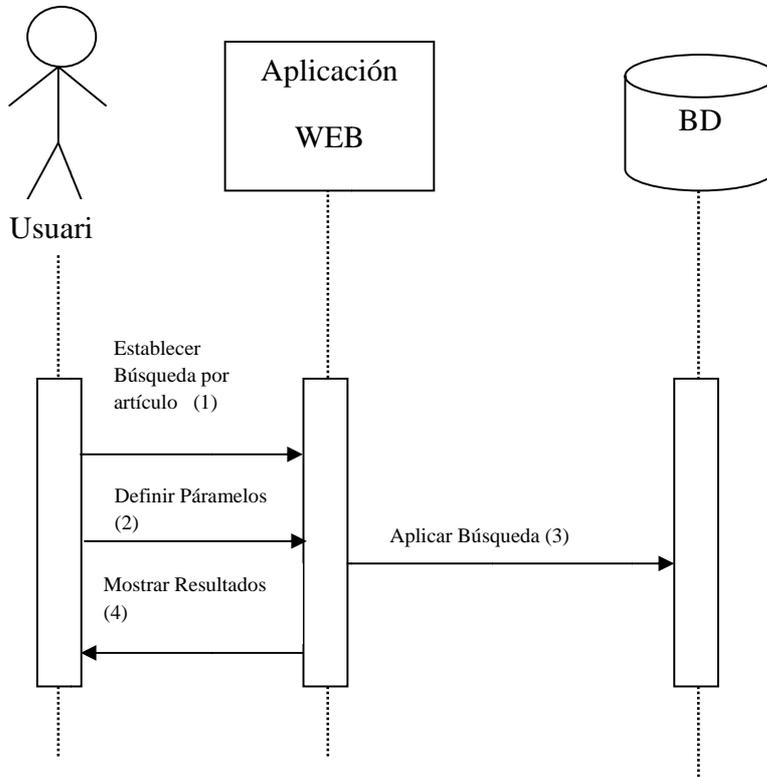


### **REALIZAR RESPALDO DE LA BASE DE DATOS**

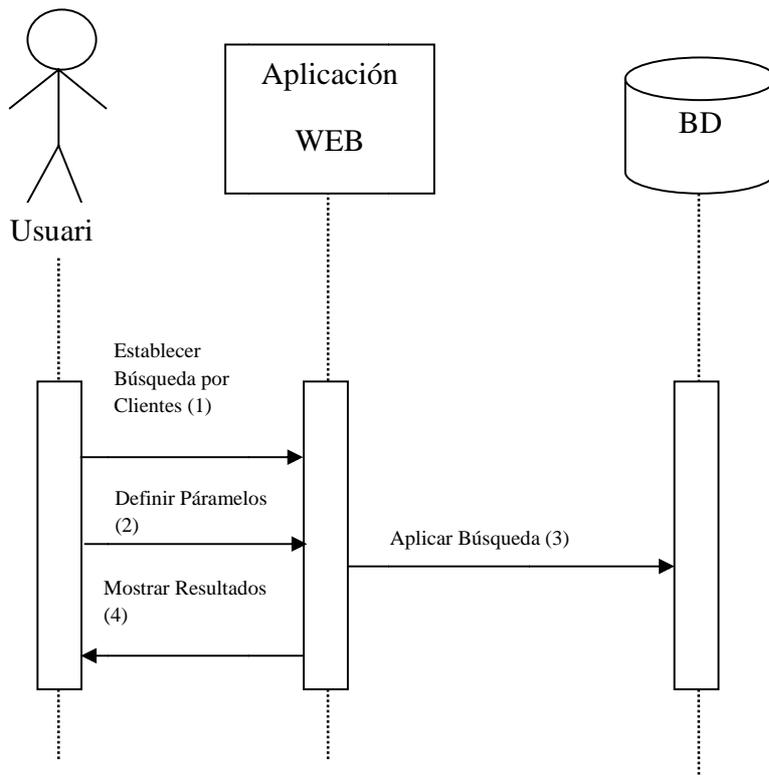




### REALIZAR BUSQUEDAS POR ARTICULOS

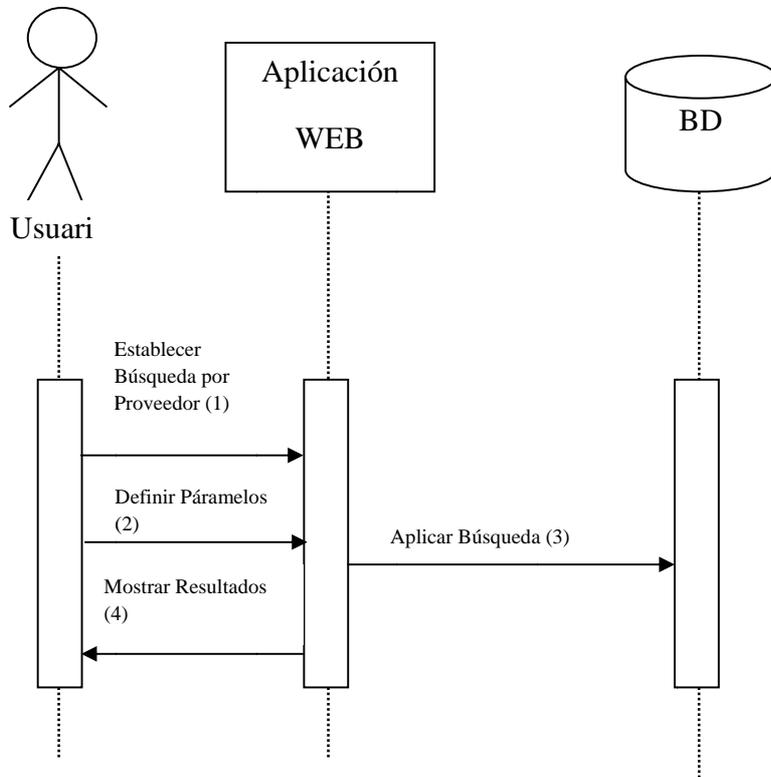


### REALIZAR BUSQUEDA POR CLIENTES

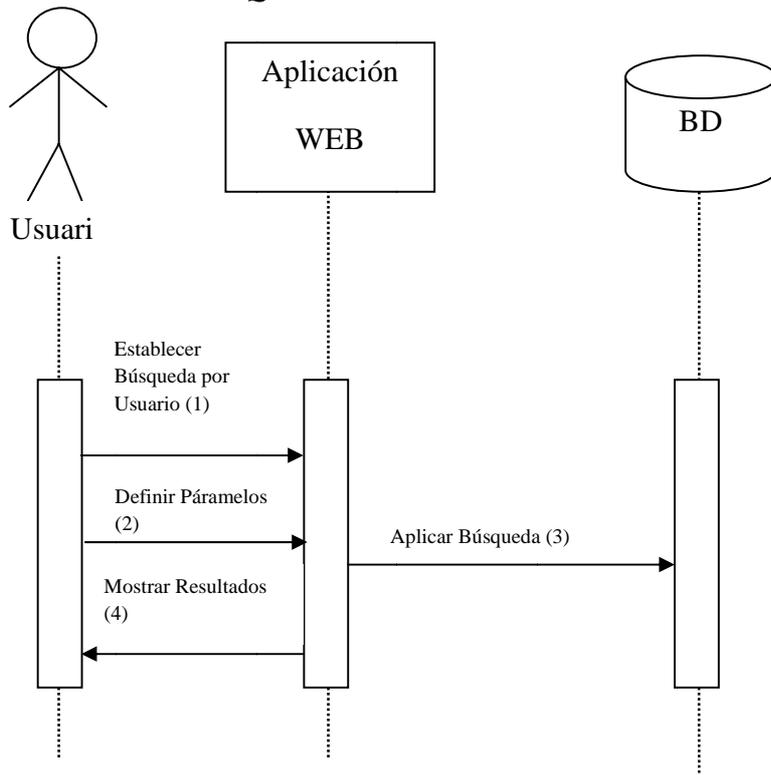




### REALIZAR BUSQUEDA POR PROVEEDORES

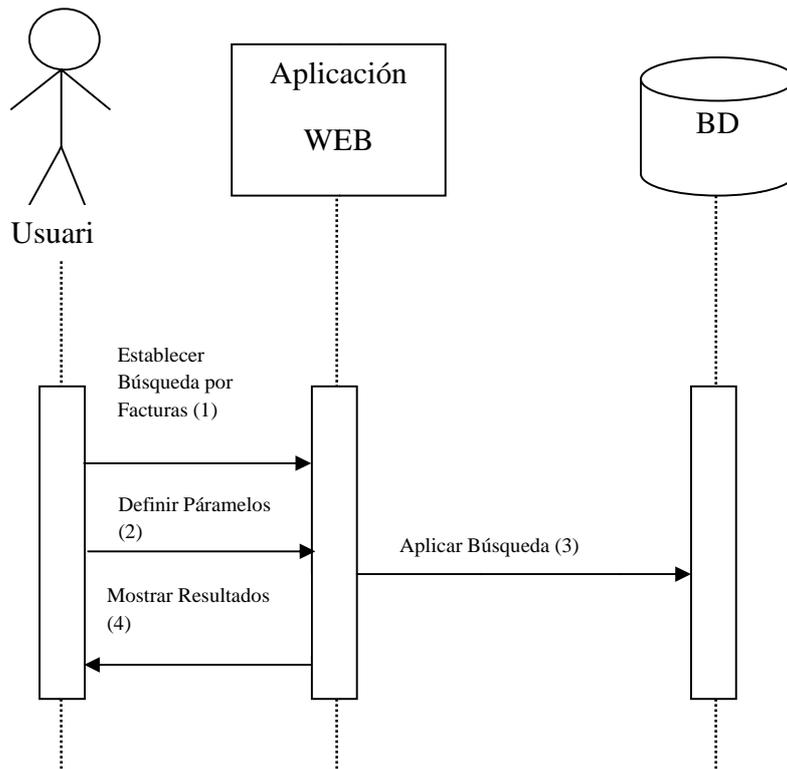


### REALIZAR BUSQUEDA POR USUARIOS

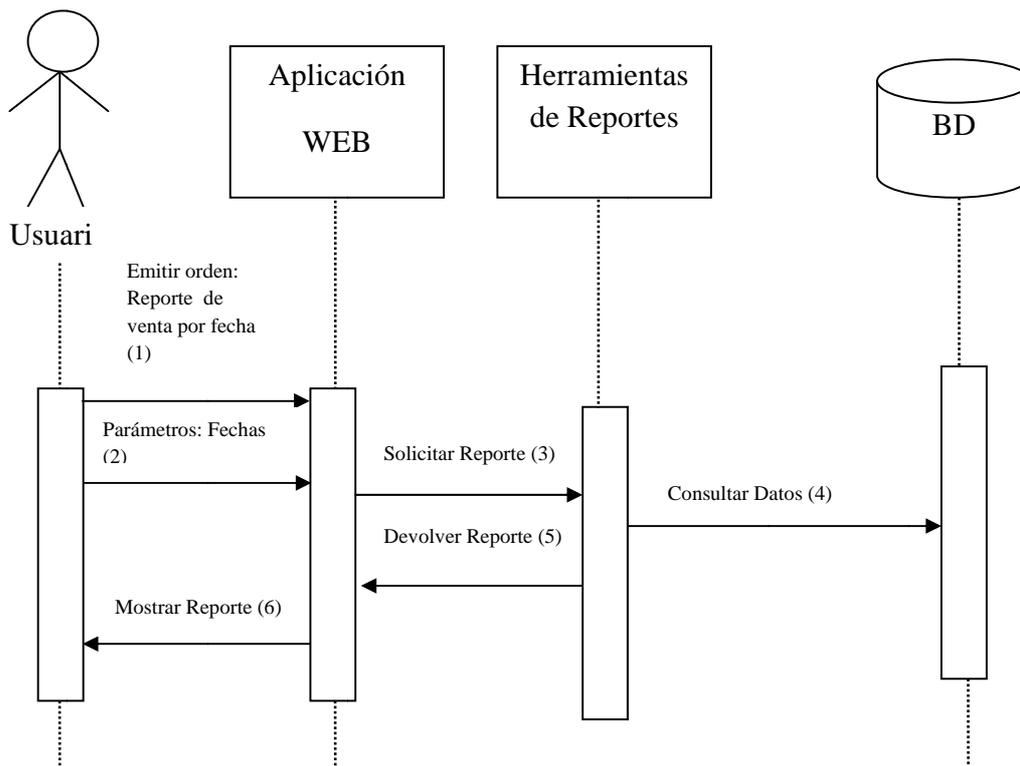




### REALIZAR BUSQUEDAS POR FACTURAS

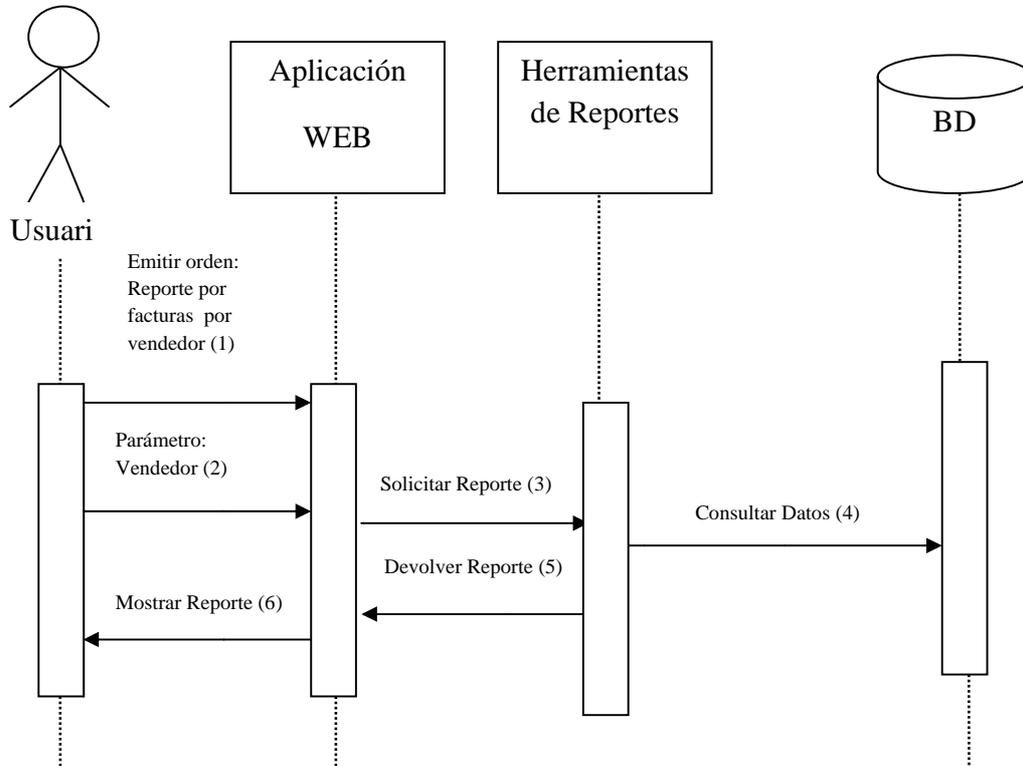


### GENERAR REPORTES DE VENTAS POR RANGO DE FECHAS

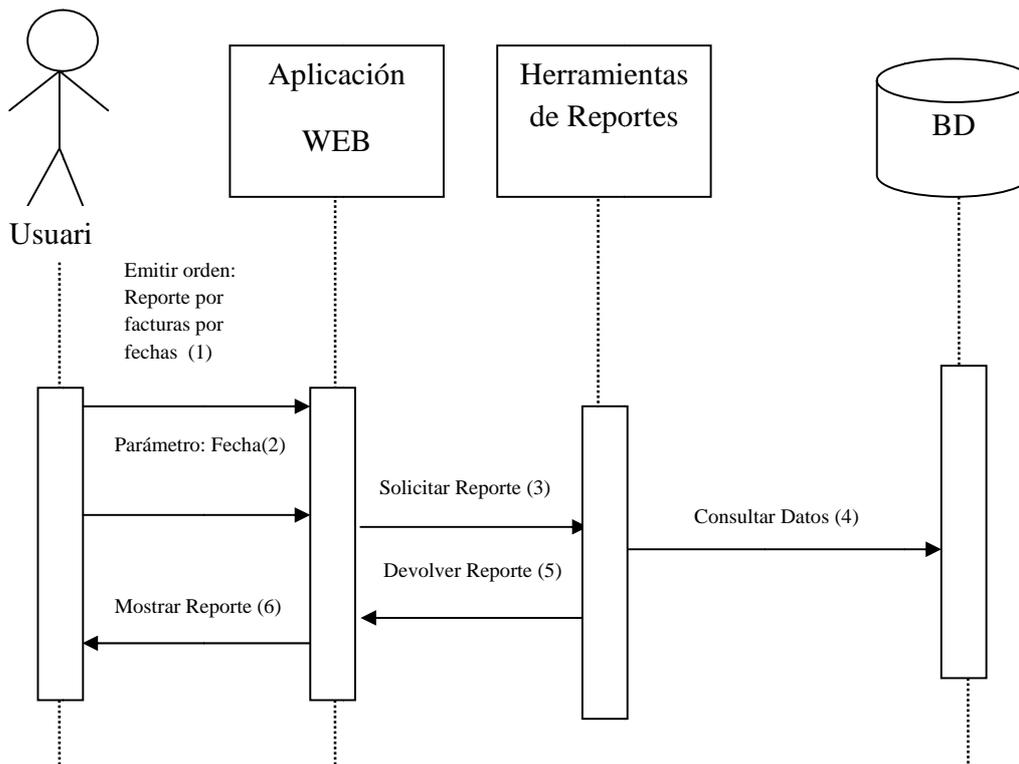




### GENERAR REPORTES DE FACTURAS POR VENDEDOR

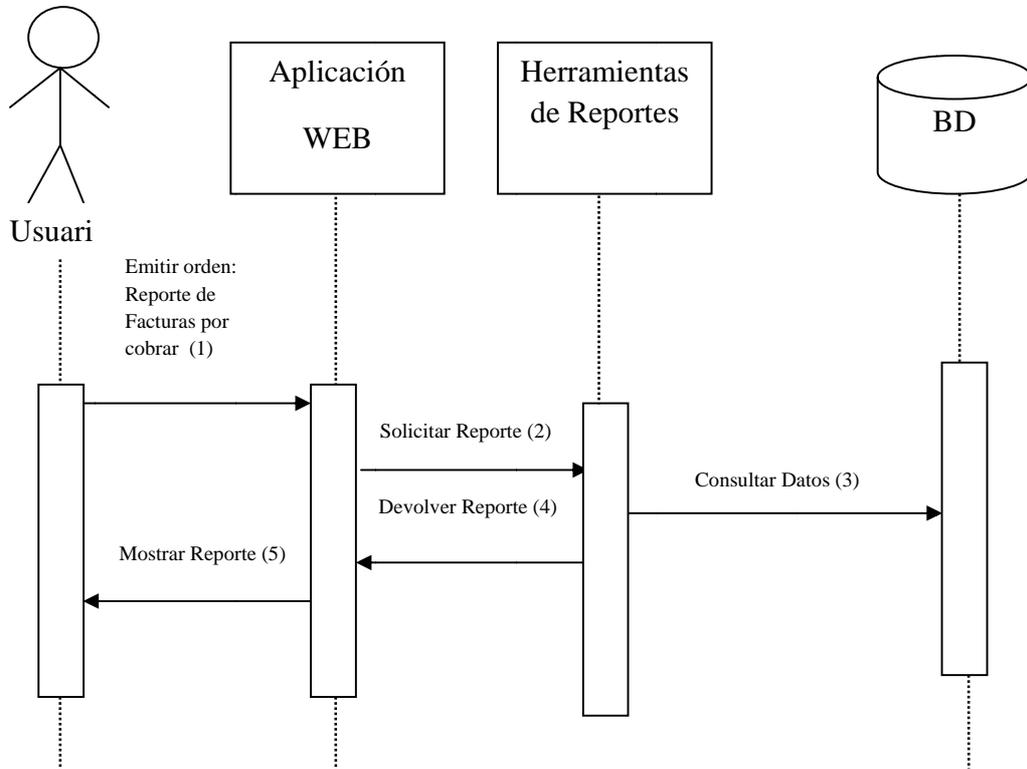


### GENERAR REPORTES DE FACTURAS POR RANGO DE FECHAS

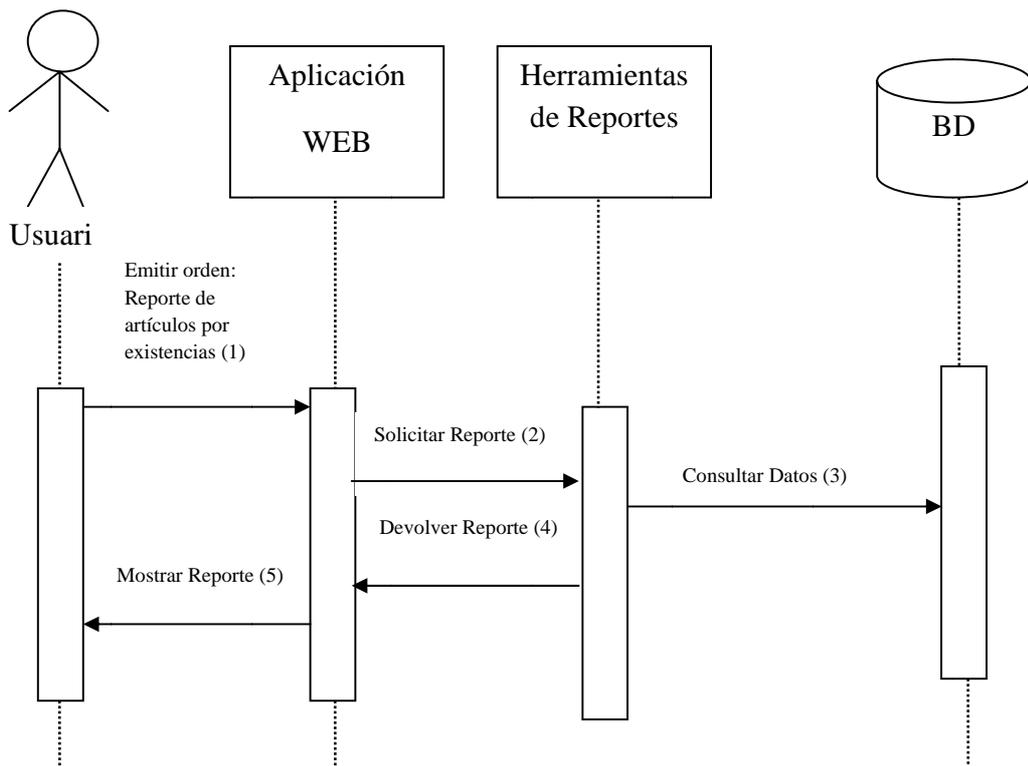




### GENERAR REPORTES DE FACTURAS POR COBRAR

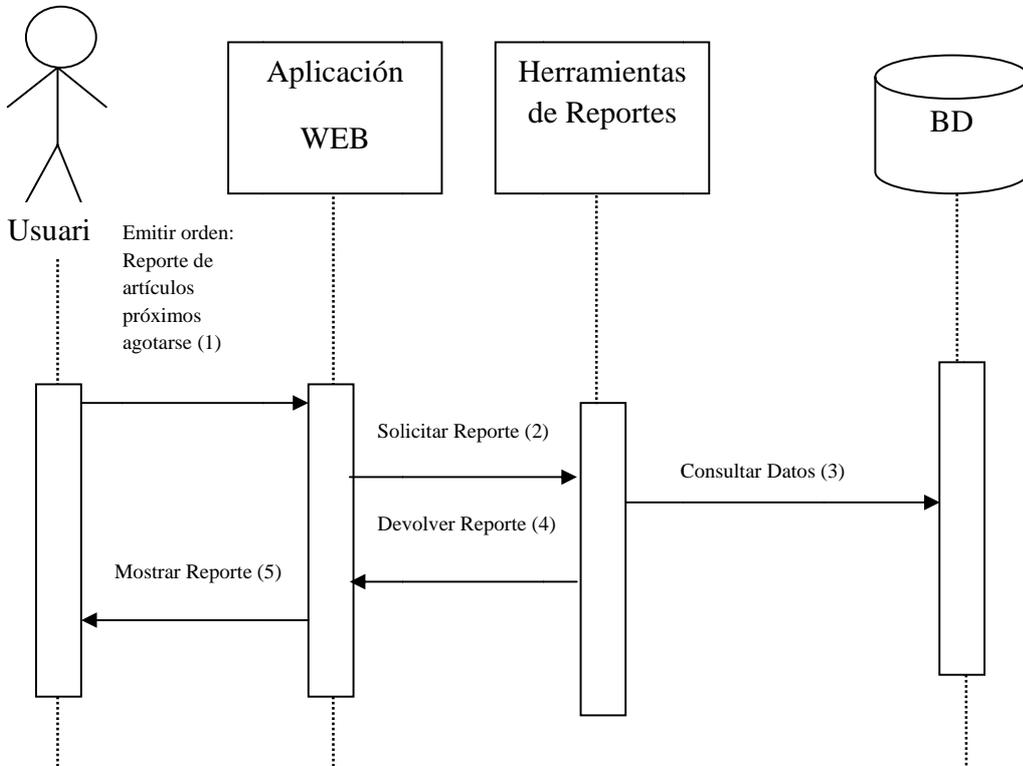


### GENERAR REPORTES DE ARTICULOS POR EXISTENCIAS

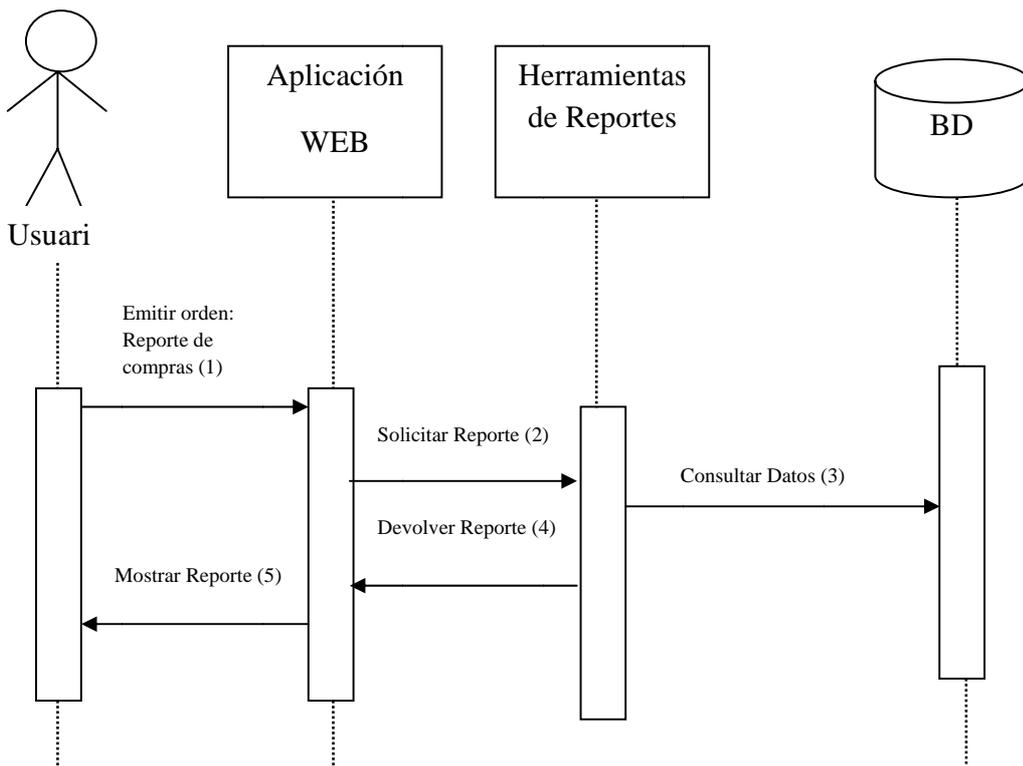




**GENERAR REPORTES DE ARTICULOS PROXIMOS AGOTARSE**



**GENERAR REPORTES DE COMPRAS**





### **13.3. Especificación de Requisitos Software Completo**

#### **1. Introducción**

##### **1.1. Propósito**

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales y del sistema para la implementación de una aplicación WEB que permitirá llevar a cabo las operaciones de inventario del negocio “El Chompipe” en el mercado el Bisne del Chinandega.

Este documento va dirigido a los propietarios del negocio y a los usuarios finales que deberán estudiarlo para su aprobación o desacuerdo antes de abordar la fase de análisis.

##### **1.2. Alcance**

Diseño, desarrollo e implementación del Sistema de Inventario del negocio “El Chompipe”  
El Sistema de Inventario será una aplicación que funcionara en entorno Web que permitirá realizar las actividades de administración, gestión e inventario del negocio. Esta aplicación realizara las siguientes funciones.

- ✓ Registrar Usuarios en el Sistema.
- ✓ Configurar el Sistema.
- ✓ Captura y registro de los datos de un Cliente.
- ✓ Captura y registro de los datos de un Proveedor.
- ✓ Registro de Artículos.
- ✓ Captura de Facturas al Crédito y al Contado.
- ✓ Registro de Caja.
- ✓ Captura del estado de cuenta de cada cliente.
- ✓ Edición de los datos de un Proveedor.
- ✓ Edición de los datos de un Usuario.
- ✓ Edición de los datos de un Artículo.
- ✓ Edición de los datos de Caja.
- ✓ Eliminar un Artículo.
- ✓ Eliminar un Cliente.
- ✓ Eliminar un Proveedor.
- ✓ Eliminar un Usuario.
- ✓ Generar los siguientes Reportes:
  - Reporte de Ventas por rango de fechas.
  - Reportes de Facturas por cobrar, por rango de fechas o por usuario.
  - Reporte de Existencia de Productos y Productos próximos a agostarse.
  - Reporte de Listado de Compras.
- ✓ Realizar búsqueda de Artículos, Proveedores, Clientes.
- ✓ Cierre de turno del negocio.
- ✓ Realizar una copia de respaldo de toda la información de la empresa.



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

### 1.3. Personal involucrado

Nombre	Gustavo Adolfo Chavarría
Rol	Programador y Coordinador
Categoría profesional	Ingeniero en Sistemas
Responsabilidades	Coordinador del proyecto y contacto con el Cliente, codificar la aplicación y diseñar la base de datos
Información de contacto	Telf. Celular: 89103865
Nombre	Gustavo Baca
Rol	Programador y Tested
Categoría profesional	Ingeniero en Sistema
Responsabilidades	Codificar la aplicación, realizar las pruebas del sistema.
Información de contacto	Telf. Celular: 83588018

Nombre	HarvinLLanes
Rol	Programador y Diseñador Grafico
Categoría profesional	Ingeniero en Sistemas
Responsabilidades	Codificar la aplicación y realizar las interfaces graficas de la aplicación.
Información de contacto	Telf. Celular: 84169135

### 1.4. Definiciones, acrónimos y abreviaturas

#### 1.4.1. Del Negocio.

- a) **Registro:** proceso en el cual se captura el dato referente a un artículo, cliente, usuario o proveedor.
- b) **Usuario:** persona que manejará la aplicación y nos brinda los requerimientos. Se conocerá como USER.
- c) **Proveedor:** empresa o persona que suministra artículos al negocio, se conocer como PROVEEDOR.
- d) **Factura al Contado:** Documento que se entrega al cliente al cancelar la compra de un artículo. Se conocerá como FAC\_CONT.
- e) **Factura al Crédito:** Documento que se entrega al cliente al realizar una compra para cancelarse posteriormente. Se conocerá como FAC\_CRED.
- f) **Estado de Cuenta:** Cantidad de dinero que un Cliente debe pagar al negocio. Se conocerá como EST\_CNT.
- g) **Cliente:** persona que realiza una compra en el negocio.
- h) **Caja:** cantidad de dinero en existencia en el negocio.
- i) **Artículo:** objeto suministrado por un Proveedor al que se le estipula un precio y una categoría. Se conocerá como ARTICULO.

#### 1.4.2. Del Sistema.

- a) **Administrar:** acción de agregar, modificar, eliminar o buscar un cliente, articulo, proveedor, usuarios o caja.



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

- b) **Usuario:** persona que interactúa de manera directa con el sistema, previamente registrado y autenticado el sistema. Se conocerá como USUARIO.
- c) **Administrador del Sistema:** Persona encargada de ofrecer el soporte técnico y operativo del Sistema de Inventario.
- d) **Pruebas:** Proceso mediante el cual se realizan actividades para verificar la óptima función del sistema.
- e) **Rol:** conjunto de permisos que pueden asignarse a un usuario.

### 1.4.3. De tecnología.

- a) **Dominio:** Nombre base que agrupa a un conjunto de equipos o dispositivos y que permite proporcionar nombres de equipo más fácilmente recordables en lugar de una dirección IP numérica, Ej.: inventario.com.
- b) **Servidor WEB:** Es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertextmarkuplanguage): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música
- f) **URL:** Significa UniformResourceLocator, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.
- g) **HTTP:** El protocolo de transferencia de hipertexto (HTTP, HyperTextTransferProtocol) es el protocolo usado en cada transacción de la Web (WWW).
- h) **Enlaces :** una referencia en un documento de hipertexto a otro documento o recurso
- i) **Sistema de gestión de Base de Datos:** Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.
- j) **Base de Datos:** Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.
- k) **Aplicación:** Es un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo.
- l) **ASP.NET:** tecnología destinada para el comportamiento activo de páginas web permitiendo ejecutar código en el Servidor que solicita un Cliente.
- m) **MySQL:** Sistema de gestión de base de datos relacional. Comúnmente utilizados en aplicaciones Web.
- n) **Apache:** Servidor de WEB de código abierto.
- o) **Dirección IP (Internet Protocol):** Número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol).



- p) **Protocolo IP** : El Protocolo de Internet (IP, de sus siglas en inglés InternetProtocol) es un protocolo no orientado a la conexión, usado tanto por el origen como por el destino para la comunicación de estos a través de una red (Internet).
- q) **HTML**: (HyperText Markup Language): Es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.
- r) **Ajax** :(Asynchronous JavaScript And XML): es una técnica de desarrollo web para crear aplicaciones interactivas.

### 1.5. Referencias

Referencia	Título	Ruta	Fecha	Autor
Standard IEEE 830 - 1998	IEEE			IEEE

### 1.6. Resumen

En el presente documento se encontrará información acerca de las características del producto de software, interfaces de usuario, interfaces del sistema, características del usuario y una descripción de cada uno de los requisitos funcionales y no funcionales del sistema.

## 2. Descripción general

### 2.1 Perspectiva del producto

El Sistema de Inventario será una aplicación diseñada para trabajar en entorno WEB, lo que permitirá su uso descentralizado, es además un producto independiente de otros sistemas.

### 2.2. Funcionalidad del producto

El sistema de Inventario realizará las siguientes funciones:

- ✓ **Registrar Usuarios en el Sistema**: el proceso que permitirá registrar un nuevo USUARIO en el sistema.
- ✓ **Configurar el Sistema**: Proceso en el que un usuario del sistema establecerá los valores de cualquiera de los siguientes datos: valor del IVA, nombre de la empresa, dirección de la empresa, teléfono de la empresa, tasa de cambio del dólar de la empresa, número RUC del negocio.
- ✓ **Captura y registro de los datos de un Cliente**: Proceso en el que se captura y registra los datos de un CLIENTE.
- ✓ **Captura y registro de los datos de un Proveedor**: Proceso en el que se captura y registran los datos de un PROVEEDOR.



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

- ✓ **Registro de Artículos:** Proceso en el que se captura y registra información acerca de un nuevo ARTICULO suministrado por un PROVEEDOR.
- ✓ **Captura de Facturas al Crédito y al Contado:** Proceso en el que se captura y registra información sobre una compra, posteriormente al registro se genera un FAC\_CRED o FAC\_CONT que se entrega al CLIENTE el cual contiene la información de la compra.
- ✓ **Registro de Caja:** proceso en el que el que un usuario del sistema establece el valor de Caja.
- ✓ **Captura del estado de cuenta de cada cliente:** Proceso en el que se captura y registra información referente al EST\_CNT de un CLIENTE.
- ✓ **Edición de los datos de un Proveedor:** Proceso en el que se modifica los datos referente a un PROVEEDOR y se registran los cambios.
- ✓ **Edición de los datos de un Usuario:** Proceso en el que se modifica los datos referente a un USUARIO y se registran los cambios.
- ✓ **Edición de los datos de un Artículo:** Proceso en el que se modifica los datos referente a un ARTICULO y se registran los cambios.
- ✓ **Edición de los datos de Caja:** Proceso en el que se modifica los datos referente a CAJA y se registran los cambios.
- ✓ **Eliminar un Artículo:** Proceso en el que se borra de la base de datos la información de un ARTICULO.
- ✓ **Eliminar un Cliente:** Proceso en el que se borra de la base de datos información acerca de un CLIENTE.
- ✓ **Eliminar un Proveedor:** Proceso en el que se borra de la base de datos información acerca de un PROVEEDOR.
- ✓ **Eliminar un Usuario:** Proceso en el que se borra de la base de datos los datos de un USUARIO.
- ✓ **Generar los siguientes Reportes:** Proceso en el que se genera un documento que contiene información sobre.
  - **Ventas por rango de fechas:** el usuario elige el rango de fechas al generarse el documento contiene los siguientes datos: número de ventas, fecha, número de artículo, número de clientes, cantidad, total y numero de factura.
  - **Facturas por cobrar:** el documento contiene los siguientes datos: número de factura, fecha, observaciones, vendedor, tipo, estado, total.
  - **Facturas por rango de fechas:** el usuario elige el rango de fechas al generarse el documento contiene los siguientes datos: número de factura, fecha, observaciones, vendedor, tipo, estado, total.
  - **Facturas por usuario:** el usuario indica el nombre del usuario y al generarse el documento contiene los siguientes datos: número de factura, fecha, observaciones, total, tipo, estado y el total.
  - **Existencia de Productos o Próximos agotarse:** el documento contiene la siguiente información: numero, nombre del artículo, marca, precio, costo, existencia, descripción, fecha de adquisición.
  - **Reporte de Listado de Compras:** el documento contiene la siguiente información: número de compra, número de artículo, marca, numero de proveedor, cantidad, precio, fecha, no registro y el total.
- ✓ **Realizar búsqueda de Artículos, Proveedores, Clientes:** Proceso en el que un usuario solicita información sobre un PROVEEDOR, CLIENTES O ARTICULOS, para ello el usuario ofrece información sobre el objeto a buscar,



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

- ✓ **Cierre de turno:** Proceso en el cual el cliente obtendrá las ganancias y conocerá el total de ventas, compras, otros gastos e ingresos.
- ✓ **Realizar una copia de respaldo de toda la información de la empresa:** Proceso en el que un USUARIO genera un archivo que contiene toda la información del negocio.

### 2.3. Características de los usuarios

El sistema de Inventario contiene dos tipos de usuario que interactuarán y lo administrarán: Administrador, Normal.

Tipo de usuario	Normal
Formación	Ninguna
Habilidades	Dominio de software de oficina y uso del Sistema de Inventario.
Actividades	Registro de Cliente, Artículos, Proveedores o Compras y generar reportes.

Tipo de usuario	Administrador
Formación	Ninguna
Habilidades	Dominio del Software de Inventario y conocimiento de la información vital de la empresa.
Actividades	Registro de Usuarios, Establece configuraciones del Sistema, Modificar Caja.

### 2.4. Restricciones

#### 2.4.1. Políticas de Software.

EL Sistema de Inventario se desarrollará usando software de licencia abierta por lo que no se deberá pagar por el uso de: servidor WEB (Apache), Sistema de Gestión de base de datos (MySQL) y lenguaje de programación C# en conjunto con la tecnología ASP.NET. El uso de este software estará regido por las políticas de licenciamiento de software.

#### 2.4.2. Limitaciones de Hardware.

Para el funcionamiento de la aplicación es necesario un computador servidor en el cual se instalará el servidor Web Apache, MySQL, el complemento de ASP.NET para Apache y la aplicación de Sistema de Inventario.

#### 2.4.3. Interfaces con otras aplicaciones.

La aplicación será un sistema autónomo y no interactuará con otras aplicaciones por lo que no se desarrollará interfaces con otras aplicaciones. Las conexiones necesarias para la conexión del servidor web, MySQL, Asp.Net, se hará por medio de la configuración de estos programas.



#### **2.4.4. Funcionamiento Paralelo.**

No es condición del proyecto.

#### **2.4.5. Funciones de Auditoria.**

No es condición del proyecto.

#### **2.4.6. Funciones de Control.**

La aplicación deberá controlar los permisos de acceso de cada usuario de para su correcta accesibilidad, de manera que únicamente pueda acceder a la información correspondiente con sus rol. Debe de tener controles de validación de datos y la correcta actualización y eliminación de información en la base de datos.

#### **2.4.7. Requisitos de Lenguaje.**

Todo el material que se realice para el usuario y la aplicación debe de estar en lenguaje español.

#### **2.4.8. Protocolos Señalados.**

Se usara el protocolo de comunicación TCP/IP, HTTP.

#### **2.4.9. Requisitos de Fiabilidad.**

Los datos correspondiente con los clientes, proveedores debe de verificarse. Los datos correspondientes a caja deben de estar ajustados a una moneda estándar y a una tasa de cambio uniforme.

#### **2.4.10. Credibilidad de la aplicación.**

Para garantizar la credibilidad de la aplicación al usuario final, el sistema será sometido a una serie de pruebas de funcionamiento para establecer que se encuentra acorde a los requerimientos que se plasman en este documento así como la consistencia de los datos, redimiendo de la aplicación y tiempo de respuesta.

#### **2.4.11. Consideraciones de Seguridad.**

Cada usuario debe de autenticarse y su acceso verificado por una sola terminal para su correcto desempeño según sus rol. Todas las claves de seguridad deben de estar en su defecto encriptados en la base de datos para ofrecer un correcto funcionamiento.

### **2.5. Suposiciones y dependencias**

- ✓ La red interna deberá de estar configurada para el manejo de protocolos TCP/IP, HTTP, DNS, principalmente todo lo relacionado en cuanto a desempeño y seguridad.



- ✓ Debe realizarse una capacitación adecuada y acorde a lo que cada usuario va a realizar. Su capacitación se hará en el momento que sea necesaria y a la persona indicada.

## 2.6. Evolución previsible del sistema

No se planea ninguna versión mejorada del sistema.

## 3. Requisitos específicos

### 3.1. Requisitos Funcionales

#### 3.1.1. Registrarse en el Sistema

##### 3.1.1.1. Especificación

###### 3.1.1.1.1. Introducción

Este proceso se encargará de hacer la captura de todos los datos referente a un USUARIO del sistema a través de un formulario por pantalla y realizará automáticamente el registro de este

###### 3.1.1.1.2 Entradas

Por pantalla: datos para registrar el Usuario

- ✓ Nombre
- ✓ Cedula
- ✓ Clave
- ✓ Dirección
- ✓ Teléfono
- ✓ Tipo

Datos proporcionados por el Sistema

- ✓ Ninguno

###### 3.1.1.1.3. Proceso.

Se mostrará por pantalla en el navegador un formulario para la introducción de datos del usuario. El número con que se registrará el nuevo usuario será asignado



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

automáticamente de manera correlativa y su estado establecido de acuerdo a si está conectado o no al sistema. La fecha de registro se registra automáticamente.

Los datos necesarios a introducir serán:

- ✓ Nombre del Usuario: es un dato obligatorio. No es necesario que exista en la tabla Usuarios..
- ✓ Cedula: es un dato obligatorio. No es necesario que exista en la tabla de Usuarios.
- ✓ Clave: dato obligatorio. No debe de existir en la tabla Usuarios y en caso de que exista debe de enviarse un mensaje de advertencia.
- ✓ Dirección: dato opcional. No es necesario que exista en la tabla Usuario de la base de datos.
- ✓ Tipo: dato obligatorio. A menos que el usuario no sea de tipo Administrador, debe de haber al menos un usuario de tipo Administrador.
- ✓ Teléfono: dato opcional.

### 3.1.1.1.4. Salida

Con todos los datos mencionados se almacenara el registro de un nuevo usuario en la base de datos del sistema posteriormente se enviara por pantalla un mensaje que indica si se realizó o no el registro en la base de datos.

## 3.1.2. Configurar Sistema

### 3.1.2.1. Especificación

#### 3.1.2.1.1. Introducción

Este proceso se encargara de realizar la captura de los datos referente a las configuraciones del sistema que los USUARIOS establecerán a través de un formulario y realizar automáticamente el registro y aplicación de estas.

#### 3.1.2.1.2 Entradas

Por pantalla: datos para establecer las configuraciones

- ✓ IVA
- ✓ Nombre de la Empresa
- ✓ Tasa de Cambio
- ✓ Estado de la Clave



- ✓ Numero Ruc
- ✓ Dirección
- ✓ Teléfono

Datos proporcionados por el Sistema

Referente al Usuario

- ✓ Nombre
- ✓ Estado
- ✓ Clave
- ✓ Dirección
- ✓ Teléfono
- ✓ Tipo

#### 3.1.2.1.3. Proceso

Se mostrara por pantalla un formulario para la introducción de datos acerca de la configuración del sistema con respecto a la empresa. El id de la nueva configuración será asignado automáticamente de manera correlativa, la fecha de la nueva configuración será establecida por el sistema.

Los datos necesarios a introducir será:

- ✓ IVA: dato opcional.
- ✓ Nombre de la Empresa: dato obligatorio.
- ✓ Tasa de Cambio: dato opcional. No es necesario que exista en la base de datos.
- ✓ Numero de RUC: dato obligatorio. No debe de existir en la tabla Configuraciones de la base de datos de lo contrario se indicara con un mensaje de error.
- ✓ Dirección: dato opcional. No es necesario que exista en la tabla configuraciones de la base de datos.
- ✓ Teléfono: dato opcional.

#### 3.1.2.1.3. Salida

Con los datos mencionados se almacenara la configuración establecida por un usuario, se enviara un mensaje por pantalla que indique que la configuración se ha guardado exitosamente.



### 3.1.3. Registrar Clientes

#### 3.1.3.1. Especificación

##### 3.1.3.1.1. Introducción

Este proceso se encarga de la captura de datos de un CLIENTE mediante un formulario web que se muestra en el Navegador, posteriormente se realiza el registro del cliente automáticamente.

##### 3.1.3.1.2. Entradas

Por pantallas: datos para registrar un cliente.

- ✓ Nombre
- ✓ Teléfono
- ✓ Dirección
- ✓ Ciudad
- ✓ Cedula
- ✓ Email
- ✓ Límite de Crédito
- ✓ Tipo

Datos proporcionados por el sistema:

- ✓ Ninguno

##### 3.1.3.1.3. Proceso

Se muestra en por pantalla un formulario web para la introducción de datos correspondientes a un nuevo cliente. El identificador de este nuevo cliente se generara automáticamente de manera de manera correlativa-

Los datos necesarios para generar el registro serán:

Nombre de Cliente: dato obligatorio. Se almacenara en el campo nombre de la tabla cliente de la base de datos.

Teléfono: dato opcional. Este se almacenara en el campo teléfono de la tabla clientes de la base de datos.

Dirección: dato obligatorio. Este dato se almacenará en el campo dirección de la tabla cliente de la base de datos.



Ciudad: dato opcional. Este dato se almacena en el campo ciudad de la tabla clientes.

Cedula: dato obligatorio. Se debe de verificar que este campo este compuesto por 11 dígitos o letras.

Email: dato opcional. Se debe de comprobar que la dirección de email es una dirección validar para validar la cuenta.

Límite de Crédito; dato opcional.

Tipo: dato opcional.

#### 3.1.3.1.4. Salidas

Con los datos mencionados anteriormente se hará el registro de un nuevo cliente en el sistema y se confirmara con un mensaje por pantalla cuando se haya realizado satisfactoriamente.

### 3.1.4. Registrar Proveedor

#### 3.1.4.1. Especificación

##### 3.1.4.1.1 Introducción

Este proceso se encarga de realizar a través de un formulario la captura de todos los datos de un PROVEEDOR y realzar automáticamente el registro de este.

##### 3.1.4.1.2. Entradas

Por pantalla: los datos para registrar un proveedor.

- ✓ Nombre del proveedor
- ✓ Cedula
- ✓ Dirección
- ✓ Ciudad
- ✓ Teléfono
- ✓ Nombre de la Empresa
- ✓ Email
- ✓ Dirección Web de la Empresa
- ✓ Observaciones

Datos proporcionados por el sistema:



Ninguno.

#### 3.1.4.1.3. Proceso

Al solicitar el registro de un nuevo proveedor se muestra por pantalla un formulario de registro con todos los campos referente a un proveedor. El identificador de este nuevo proveedor se generara automáticamente de manera correlativa.

Los datos necesarios a introducir son:

Nombre del proveedor: dato obligatorio. No es necesario que exista en la base de datos.

Cedula: dato obligatorio. Se debe de verificar que la longitud del este dato corresponda con los 11 dígitos o letras de una cedula de identidad.

Dirección: dato obligatorio.

Ciudad: dato obligatorio.

Teléfono: dato opcional. Se debe comprobar que la longitud del número de teléfono corresponda con un número de teléfono real.

Empresa: dato opcional.

Email: dato opcional. Se debe verificar que la dirección electrónica se verídica.

Web: dato opcional.

Observaciones: dato opcional.

#### 3.1.4.1.3. Salida

Con los datos anteriormente mencionados se hará el registro de un nuevo proveedor en la base de datos posteriormente se enviara un mensaje de confirmación por pantalla.

### 3.1.5. Registrar Artículo

#### 3.1.5.1. Especificación

##### 3.1.5.1.1. Introducción



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

Este proceso se encarga de la captura de los datos de un ARTÍCULO mediante un formulario web que se muestra por pantalla y se hará automáticamente el registro de este.

### 3.1.5.1.2. Entradas

Por pantalla datos para registrar el artículo:

- ✓ Nombre del Artículo
- ✓ Marca
- ✓ Categoría
- ✓ Precio
- ✓ Unidad Medida.
- ✓ Costo
- ✓ Existencias
- ✓ Cantidad
- ✓ Descuento
- ✓ Estado
- ✓ Descripción
- ✓ Fecha de Adquisición
- ✓ Fecha de Caducidad.

Datos proporcionados por el sistema

Referente al proveedor.

- ✓ Nombre.
- ✓ Cedula
- ✓ Dirección
- ✓ Ciudad
- ✓ Teléfono
- ✓ Empresa
- ✓ Email
- ✓ Web
- ✓ Observaciones

### 3.1.5.1.3 Proceso

Se muestra por pantalla en el navegador web un formulario de registro de artículos. El identificador con que se hará el registro de este nuevo artículo se generara de manera automática.



Los datos necesarios a introducir serán:

- ✓ Nombre: dato obligatorio.
- ✓ Unidad de Medida: dato obligatorio. Debe ser únicamente una de las cinco unidades de medida (Unidad, Docena, Metro, Yarda, Libra).
- ✓ Estado: dato obligatorio. Debe de indicarse como un de los cuatros estados de un artículo (bueno, regular, retirar, devuelto).
- ✓ Cantidad: dato obligatorio. Debe de ser mayor que cero.
- ✓ Marca: dato opcional.
- ✓ Categoría: dato opcional.
- ✓ Precio: dato obligatorio. Debe de indicarse el precio por unidad de manera flotante.
- ✓ Proveedor: dato opcional. Debe de existir en la tabla proveedores de la base de datos.
- ✓ Costo: dato obligatorio.
- ✓ .Descuento: dato opcional.
- ✓ Descripción: dato opcional.
- ✓ Fecha Adquisición: dato opcional.
- ✓ Fecha Caducidad: dato opcional.

A partir de los datos anteriores se calculara el precio por docena y se actualizara las existencias de dicho artículo.

#### 3.1.5.1.4 Salida.

Con los datos anteriores se registrara el nuevo artículo, posteriormente se enviara un mensaje por pantalla que indique si se completó el registro.

### **3.1.6. Registrar Facturas al Contado**

#### 3.1.6.1. Especificación

##### 3.1.6.1.1. Introducción

Este proceso se encarga de la captura de datos de una venta a través de un formulario web y realiza automáticamente el registro de una FAC\_CONT en la base de datos.

##### 3.1.6.1.2. Entradas.



Por pantalla datos para registrar la factura:

- ✓ Vendedor
- ✓ Cliente
- ✓ Artículo
- ✓ Cantidad
- ✓ Tipo Factura

Datos proporcionados por el sistema

- ✓ Nombre del Artículo
- ✓ Numero de Artículo
- ✓ Descripción
- ✓ Precio
- ✓ Unidad Medida
- ✓ Total

#### 3.1.6.1.3. Proceso.

Se muestra por pantalla un formulario para la introducción de datos sobre una nueva venta en la que se registrarán los datos sobre el o los artículos que se compren. La fecha será proporcionada automáticamente por el sistema.

Los datos necesarios a introducir serán:

Vendedor: dato obligatorio.

Cliente: dato obligatorio. Se debe de verificar que el cliente exista en la base de datos de lo contrario se enviara un mensaje de error por pantalla, si es un nuevo cliente se procederá a registrarlo.

Artículo: dato obligatorio. Se mostrara una lista de Artículos en existencias para su venta.

Cantidad: dato obligatorio. Se debe de verificar que la cantidad no sobrepase las existencias.

Tipo de Factura. Dato obligatorio. En caso de no identificar la factura se establecer por defecto como una factura al contado.

A partir de esto datos se calculara el precio total a pagar por artículo, el Subtotal de la compra y al registrar el pago el estado de la factura.



#### 3.1.6.1.4. Salidas.

Con todos los datos mencionados se almacenar el registro de una nueva factura al contado en la base de datos y se mostrar la opción de imprimir la factura, posteriormente al registro de la venta se enviara un mensaje por pantalla que indique si el registro fue exitoso o no.

### 3.1.7. Registrar Factura al Crédito.

#### 3.1.7.1. Especificación.

##### 3.1.7.1.1. Introducción.

Este proceso se encarga de la captura de datos de una venta a través de un formulario web de manera que cuando un cliente quiere realizar una compra al crédito se registra automáticamente el registro de una al FAC\_CRED en la base de datos.

##### 3.1.7.1.2. Entradas.

Por pantalla los datos para registrar la factura:

- ✓ Vendedor.
- ✓ Cliente.
- ✓ Artículo.
- ✓ Cantidad.
- ✓ Tipo Factura.
- ✓ Monto Inicial.
- ✓ Tiempo de Crédito.
- ✓ Porcentaje por Mora.

Datos proporcionados por el sistema

- ✓ Nombre del Artículo.
- ✓ Numero de Artículo.
- ✓ Descripción.
- ✓ Precio.
- ✓ Unidad Medida.
- ✓ Total.
- ✓ Cantidad a Pagar por Cuota.

##### 3.1.7.1.3. Proceso.



Se muestra por pantalla un formulario para el registro de una nueva venta en la que se registra los datos sobre el o los artículos que se compren. Los datos referentes al cliente y el artículo son proporcionados por el sistema.

Los datos necesarios a introducir son:

- ✓ Cliente: dato obligatorio. Se debe de verificar que el cliente exista en la base de datos de lo contrario se enviara un mensaje de error por pantalla, si es un nuevo cliente se procederá a registrarlo.
- ✓ Artículo: dato obligatorio. Se mostrara una lista de Artículos en existencias para su venta.
- ✓ Cantidad: dato obligatorio. Se debe de verificar que la cantidad no sobrepase las existencias.
- ✓ Monto Inicial: dato obligatorio. Se debe de verificar que el monto no sea mayor que el límite de crédito.
- ✓ Tiempo de crédito: dato obligatorio. Se debe de indicar el tipo de crédito (mensual o quincenal).
- ✓ Porcentaje Mora: dato opcional.

A partir de estos datos se calculara el Total por Artículo, Subtotal de la compra y los datos referentes al Estado de Cuenta de un Cliente.

#### 3.1.7.1.4. Salida.

Con todos los datos mencionados se almacenara una Factura al Crédito, posterior se habilitara la opción de imprimir la Factura, al registrar la Venta se enviara un mensaje por pantalla que indique si el registro de esta fue éxitos o no.

### 3.1.8. Registrar Caja

#### 3.1.7.1. Especificación.

##### 3.1.7.1.1. Introducción.

Este proceso se encargara de capturar de los datos referentes a CAJA a través de un formulario web y realizar automáticamente el registro de esta en la base de datos.

##### 3.1.8.1.2. Entradas.



Por pantalla los datos para registrar la CAJA:

- ✓ Caja Inicial.
- ✓ Nombre del Usuario.

Datos proporcionados por el sistema.

Referente al Usuario:

- ✓ Usuario
- ✓ Monto inicial en caja
- ✓ fecha

#### 3.1.8.1.3. Proceso.

Se mostrara por pantalla un formulario para la introducción de los datos referente a caja. El identificador de caja ser generar automáticamente. La fecha de inscripción de caja se generar automáticamente.

Los datos necesarios a introducir:

- ✓ Caja Inicial: dato opcional. Debe de indicarse la cantidad inicial de la caja y registrarse en el campo caja inicial de la base de datos.
- ✓ Nombre de Usuario: dato obligatorio. Se debe de verificar que el registro exista en la tabla usuarios de la base de datos de otro forma se deberá de indicar un error

#### 3.1.8.1.4. Salida.

Con los datos mencionados se hará el registro de una nueva caja en la base de datos para un determinado usuario. Posteriormente se enviara un mensaje que indicara si se logró registrar exitosamente de lo contrario enviara un error.

### **3.1.9- Registrar Estado de Cuenta de cada Cliente**

#### **3.1.9.1- Especificación.**

##### **3.1.9.1.1- Introducción.**

Su objetivo es llevar un control del estado de un cliente frente a las ventas que este ha realizado al crédito y saber de esta forma cuanto es su saldo en mora, las fechas en que a realizados los pagos para cancelar la deuda, la cantidad abonada, tipo de crédito entre otros.

##### **3.1.9.1.2- Entradas.**



No de cuota, No de Cliente, No factura, fecha inicial, estado, monto inicial, monto por cuota, tiempo de crédito, tipo de crédito, monto pagado.

**3.1.9.1.3- Proceso:**

Se consulta al servidor de Base de datos y se muestra por pantalla los datos asociados estado\_cuenta. Los datos pueden ser editados si el usuario así lo desea y posteriormente actualizados o guardados en el servidor de bases de datos.

**3.1.9.1.4- Salida:**

Se muestran por pantalla todos los campos de la tabla estado cuenta:  
No de estado de cuenta, No cuota, No cliente, No factura, Fecha inicial, Estado, Monto en mora, Porcentaje en mora, monto inicial, monto por cuota, tiempo de crédito, tipo de crédito, monto pagado, fecha realizado el pago.

### **3.1.10- Editar Proveedor**

#### **3.1.10.1- Especificación.**

##### **3.1.10.1.1- Introducción.**

Este proceso se encarga editar los datos asociados a los proveedores en la base de datos.

##### **3.1.10.1.2- Entradas.**

A través de la interfaz gráfica de usuario se accede a todos los datos asociados a los proveedores, esto es acceder a la tabla correspondiente (proveedores) en la base de datos.

##### **3.1.10.1.3- Proceso:**

Se muestran los datos encontrados en la base de datos, el usuario puede editar dichos datos si eso es lo que desea, posteriormente estos datos son guardados y actualizados.

##### **3.1.10.1.4- Salida**

Se muestran los datos de la tabla proveedores:  
Número, Nombre, Cédula, Dirección, Ciudad, Teléfono, Empresa, E-mail, Web, Observaciones.

### **3.1.11- Editar Usuarios**

#### **3.1.11.1- Especificación.**

##### **3.1.11.1.1- Introducción.**



Este proceso se encarga editar los datos asociados a los usuarios en la base de datos.

#### **3.1.11.1.2- Entradas.**

A través de la interfaz gráfica de usuario se accede a todos los datos asociados a los usuarios, esto es acceder a la tabla correspondiente (usuarios) en la base de datos.

#### **3.1.11.1.3- Proceso:**

Se muestran los datos encontrados en la base de datos, el usuario puede editar dichos datos si eso es lo que desea, posteriormente estos datos son guardados y actualizados.

#### **3.1.11.1.4- Salida:**

Se muestran los datos de la tabla usuarios:  
Número, Nombre, Cédula, Dirección, Clave, Ciudad, Teléfono, Fecha, Tipo.

### **3.1.12- Editar Artículo**

#### **3.1.12.1- Especificación.**

##### **3.1.12.1.1- Introducción.**

Este proceso se encarga editar los datos asociados a los artículos en la base de datos.

##### **3.1.12.1.2- Entradas.**

A través de la interfaz gráfica de usuario se accede a todos los datos asociados a los usuarios, esto es acceder a la tabla correspondiente (artículo) en la base de datos.

##### **3.1.12.1.3- Proceso:**

Se muestran los datos encontrados en la base de datos, el usuario puede editar dichos datos si eso es lo que desea, posteriormente estos datos son guardados y actualizados.

##### **3.1.12.1.4- Salida:**

Se muestran los datos de la tabla artículo:  
Número, Nombre, Marca, Categoría, Precio unitario, Unidad Medida, Costo, Existencias, Descuento, Estado.

### **3.1.13- Editar Cliente**



### **3.1.13.1- Especificación.**

#### **3.1.13.1.1- Introducción.**

Este proceso se encarga editar los datos asociados a los clientes en la base de datos.

#### **3.1.13.1.2- Entradas.**

A través de la interfaz gráfica de usuario se accede a todos los datos asociados a los usuarios, esto es acceder a la tabla correspondiente (cliente) en la base de datos.

#### **3.1.13.1.3- Proceso:**

Se muestran los datos encontrados en la base de datos, el usuario puede editar dichos datos si eso es lo que desea, posteriormente estos datos son guardados y actualizados.

#### **3.1.13.1.4- Salida:**

Se muestran los datos de la tabla cliente:

Id Cliente, nombre, teléfono, dirección, ciudad, cedula, email, limite crédito, tipo.

### **3.1.14- Editar Caja**

#### **3.1.14.1- Especificación.**

##### **3.1.14.1.1- Introducción.**

Este proceso se encarga de editar el monto inicial de caja y los otros gastos e ingresos realizados.

##### **3.1.14.1.2- Entradas.**

El usuario puede definir la cantidad de dinero inicial (caja inicial) o modificar este valor, también puede ingresar otros gastos u otros ingresos en Caja.

##### **3.1.14.1.3- Proceso:**

Se graban los datos que el usuario a definido para la Caja en la tabla caja dentro de la base de datos, el programa efectúa el cálculo correspondiente al total de la ganancia, así como el usuario puede definir otros gastos u otros ingresos es necesario definir concepto y monto.

##### **3.1.14.1.4- Salida:**

Se muestra en la interfaz gráfica la siguiente información:

Caja Inicial, Caja Actual, Total ganancia.



### **3.1.15- Borrar Artículo**

#### **3.1.15.1- Especificación.**

##### **3.1.15.1.1- Introducción.**

Este proceso se encarga de eliminar un artículo si el usuario así lo desea de la tabla artículo en la base de datos.

##### **3.1.15.1.2- Entradas.**

El usuario a través de la interfaz gráfica de usuario selecciona el artículo, ósea el registro de la tabla artículos a eliminar.

##### **3.1.15.1.3- Proceso:**

Se pide confirmación acerca del artículo que el usuario desea eliminar, de ser afirmativo el registro se da de alta definitivamente de la base de datos.

##### **3.1.15.1.4- Salida:**

Se envía un mensaje indicando la ejecución satisfactoria de la operación.

### **3.1.16- Borrar Cliente**

#### **3.1.16.1- Especificación.**

##### **3.1.16.1.1- Introducción.**

Este proceso se encarga de eliminar un cliente si el usuario así lo desea de la tabla cliente en la base de datos.

##### **3.1.16.1.2- Entradas.**

El usuario a través de la interfaz gráfica de usuario selecciona el cliente, ósea el registro de la tabla clientes a eliminar.

##### **3.1.16.1.3- Proceso:**

Se pide confirmación acerca del cliente que el usuario desea eliminar, de ser afirmativo el registro se da de alta definitivamente de la base de datos.

##### **3.1.16.1.4- Salida:**

Se envía un mensaje indicando la ejecución satisfactoria de la operación.

### **3.1.17- Borrar Proveedor**

#### **3.1.17.1- Especificación.**

##### **3.1.17.1.1- Introducción.**



Este proceso se encarga de eliminar un proveedor si el usuario así lo desea de la tabla proveedor en la base de datos.

#### **3.1.17.1.2- Entradas.**

El usuario a través de la interfaz gráfica de usuario selecciona el proveedor, ósea el registro de la tabla proveedores a eliminar.

#### **3.1.17.1.3- Proceso:**

Se pide confirmación acerca del proveedor que el usuario desea eliminar, de ser afirmativo el registro se da de alta definitivamente de la base de datos.

#### **3.1.17.1.4- Salida:**

Se envía un mensaje indicando la ejecución satisfactoria de la operación.

### **3.1.18- Borrar Usuario**

#### **3.1.18.1- Especificación.**

##### **3.1.18.1.1- Introducción.**

Este proceso se encarga de eliminar un usuario si el propio usuario así lo desea de la tabla proveedor en la base de datos.

##### **3.1.18.1.2- Entradas.**

El usuario a través de la interfaz gráfica de usuario selecciona el proveedor, ósea el registro de la tabla proveedores a eliminar.

##### **3.1.18.1.3- Proceso:**

Se confirme que el usuario a eliminar no sea el mismo usuario en uso, de ser diferente, se pide confirmación acerca del usuario que el usuario desea eliminar, de ser afirmativo el registro se da de alta definitivamente de la base de datos.

##### **3.1.18.1.4- Salida:**

Se envía un mensaje indicando la ejecución satisfactoria de la operación.

### **3.1.19- Realizar Respaldo**

#### **3.1.19.1- Especificación**

##### **3.1.19.1.1- Introducción**

Permitirá al USER realizar una copia de seguridad de los datos almacenados en la BD.

##### **3.1.19.1.2- Entradas**

El usuario no introducirá ningún dato.

##### **3.1.19.1.3- Procesos**

Se verificará que exista la BD.



#### **3.1.19.1.4- Salida**

Se generará un archivo o script con extensión SQL que contendrá los datos y estructura de la BD.

### **3.1.20. Realizar búsquedas por Artículos**

#### **3.1.20.1. Especificación**

##### **3.1.20.1.1- Introducción**

Permitirá al USER realizar búsquedas con diversos tipos de filtrado.

##### **3.1.20.1.2- Entradas**

Opciones de búsquedas brindadas por el USER:

- Número de Art.
- Nombre
- Marca
- Categoría
- Precio
- Proveedor

##### **3.1.20.1.3- Procesos**

El sistema autocompletar la palabra a buscar según las letras que esta contenga y luego se realizará la búsqueda palabra completa.

##### **3.1.20.1.4- Salida**

Se mostrará en pantalla un listado con todos los datos relacionados al artículo buscado

### **3.1.21. Realizar búsquedas por Cliente**

#### **3.1.21.1. Especificación**

##### **3.1.21.1.1- Introducción**

Permitirá al USER realizar búsquedas con diversos tipos de filtrado.

##### **3.1.21.1.2- Entradas**

Opciones de búsquedas brindadas por el USER:

- Número de Cliente.
- Nombre
- Cédula
- Límite de Crédito
- Ciudad

##### **3.1.21.1.3- Procesos**

El sistema buscará según los datos introducidos por el usuario el cliente correspondiente.

##### **3.1.21.1.4- Salida**



Se mostrará en pantalla un listado con todos los datos relacionados al cliente buscado

### **3.1.22. Realizar búsquedas por Proveedor**

#### **3.1.22.1. Especificación**

##### **3.1.22.1.1- Introducción**

Permitirá al USER realizar búsquedas con diversos tipos de filtrado.

##### **3.1.22.1.2- Entradas**

Opciones de búsquedas brindadas por el USER:

- Número de Proveedor.
- Nombre
- Cédula
- Ciudad

##### **3.1.22.1.3- Procesos**

El sistema buscará según los datos introducidos por el usuario el proveedor correspondiente.

##### **3.1.22.1.4- Salida**

Se mostrará en pantalla un listado con todos los datos relacionados al proveedor buscado

### **3.1.23. Realizar búsquedas por Usuarios**

#### **3.1.23.1. Especificación**

##### **3.1.23.1.1- Introducción**

Permitirá al USER realizar búsquedas con diversos tipos de filtrado.

##### **3.1.23.1.2- Entradas**

Opciones de búsquedas brindadas por el USER:

- Número de USER.
- Nombre
- Cédula
- Tipo

##### **3.1.23.1.3- Procesos**

El sistema buscará según los datos introducidos por el usuario el usuario correspondiente.

##### **3.1.23.1.4- Salida**

Se mostrará en pantalla un listado con todos los datos relacionados al usuario buscado



### **3.1.24. Realizar búsquedas por Facturas**

#### **3.1.24.1. Especificación**

##### **3.1.24.1.1- Introducción**

Permitirá al USER realizar búsquedas con diversos tipos de filtrado.

##### **3.1.24.1.2- Entradas**

Opciones de búsquedas brindadas por el USER:

- Número de Factura.
- Fecha
- Estado
- Tipo
- Vendedor

##### **3.1.24.1.3- Procesos**

El sistema buscará según los datos introducidos por el usuario la factura correspondiente.

##### **3.1.24.1.4- Salida**

Se mostrará en pantalla un listado con todos los datos relacionados a la factura buscada.

### **3.1.25. Generar Reporte de ventas por rango de fecha**

#### **3.1.25.1. Especificación**

##### **3.1.25.1.1- Introducción**

Permitirá al USER realizar un informe de ventas entre fechas

##### **3.1.25.1.2- Entradas**

Datos proporcionados por el USER:

- Fecha inicial
- Fecha final

##### **3.1.25.1.3- Procesos**

El sistema buscará las ventas realizadas (“<”, “>”) entre las fechas introducidas por el usuario.

##### **3.1.25.1.4- Salida**

Se mostrará en pantalla un listado de todas las ventas entre el rango seleccionado, mostrando al final el total de ventas.

### **3.1.26. Generar Reporte de facturas por vendedor**

#### **3.1.26.1. Especificación**

##### **3.1.26.1.1- Introducción**

Permitirá al USER realizar un informe de facturas según los usuarios que hayan realizado una venta.

##### **3.1.26.1.2- Entradas**



Datos proporcionados por el USER:

- Nombre del usuario

#### **3.1.26.1.3- Procesos**

El sistema buscará solo en los usuarios que hayan realizados una ventas.

#### **3.1.26.1.4- Salida**

Se mostrará en pantalla un listado de todas las facturas cuyo vendedor sea el usuario ingresado, mostrando al final el total de facturas realizadas por este.

### **3.1.27. Generar Reporte de facturas por rango de fecha**

#### **3.1.27.1. Especificación**

##### **3.1.27.1.1- Introducción**

Permitirá al USER realizar un informe de facturas entre fechas

##### **3.1.27.1.2- Entradas**

Datos proporcionados por el USER:

- Fecha inicial
- Fecha final

##### **3.1.27.1.3- Procesos**

El sistema buscará las facturas realizadas (“<”,”>”) entre las fechas introducidas por el usuario.

##### **3.1.27.1.4- Salida**

Se mostrará en pantalla un listado de todas las facturas entre el rango seleccionado, mostrando al final el total de facturas.

### **3.1.28. Generar Reporte de facturas por cobrar**

#### **3.1.28.1. Especificación**

##### **3.1.28.1.1- Introducción**

Permitirá al USER realizar un informe de facturas al crédito en espera de cobro.

##### **3.1.28.1.2- Entradas**

El USER no introducirá ningún dato.

##### **3.1.28.1.3- Procesos**

El sistema buscará aquellas facturas que no se hayan pagado o en mora.

##### **3.1.28.1.4- Salida**

Se mostrará en pantalla un listado de todas las facturas, mostrando al final el total de facturas.



### **3.1.29. Generar Reporte de artículos por existencias**

#### **3.1.29.1. Especificación**

##### **3.1.29.1.1- Introducción**

Permitirá al USER realizar un informe de todos los artículos que tiene en existencia.

##### **3.1.29.1.2- Entradas**

El USER no introducirá ningún dato.

##### **3.1.29.1.3- Procesos**

El sistema buscará solo aquellos productos que su existencia sean mayor a cero.

##### **3.1.29.1.4- Salida**

Se mostrará en pantalla un listado de todos los artículos con todos sus datos ingresado a lo hora de ser registrados.

### **3.1.30. Generar Reporte de artículos próximos agotarse**

#### **3.1.30.1. Especificación**

##### **3.1.30.1.1- Introducción**

Permitirá al USER realizar un informe de todos los artículos que están por acabarse

##### **3.1.30.1.2- Entradas**

El USER no introducirá ningún dato.

##### **3.1.30.1.3- Procesos**

El sistema buscará solo aquellos productos que sean menor que 5 en su existencia.

##### **3.1.30.1.4- Salida**

Se mostrará en pantalla un listado de todos los artículos con todos sus datos ingresado a lo hora de ser registrados.

### **3.1.31. Generar Reporte de compras**

#### **3.1.31.1. Especificación**

##### **3.1.31.1.1- Introducción**

Permitirá al USER realizar un informe de las compras realizadas.

##### **3.1.31.1.2- Entradas**

El USER no introducirá ningún dato.

##### **3.1.31.1.3- Procesos**

El sistema buscará todas las compras realizadas

##### **3.1.31.1.4- Salida**

Se mostrará en pantalla un listado de todas las compras realizadas con todos sus datos ingresado.



### **3.1.32. Registrar Cierre de Turno.**

#### **3.1.32.1. Especificación.**

##### **3.1.32.1.1. Introducción.**

Este proceso se encarga de registrar las ganancias obtenidas al final del día y de actualizar el valor de CAJA en base al valor anterior y las ganancias actuales.

##### **3.1.32.1.2. Entradas**

- ✓ Caja Inicial.
- ✓ Caja Actual.
- ✓ Otros Ingresos.
- ✓ Otros Gastos.

##### **3.1.32.1.3. Proceso.**

Se mostrara por pantalla un formulario que capturara los datos referentes a las ganancias obtenidas en el día, las ganancias en existencias, así como los gastos y otras ganancias obtenidos.

Los datos necesarios a introducir serán:

- ✓ Caja Inicial: dato opcional. Debe de existir en la base de datos, ya que siempre debe de registrarse una caja inicial; sin embargo se podrá establecer un nuevo valor para la caja inicial.
- ✓ Otros Gastos: dato opcional. No es obligatorio que exista en la base de datos; pero si se registraron otros gastos se mostraran.
- ✓ Otros Ingresos: dato opcional. No es obligatorio que exista en la base de datos, únicamente cuando se registren otros gastos se mostraran.

Con los datos mencionados anteriormente se calculara el nuevo valor de caja y se registrara el cierre del turno.

##### **3.1.32.1.4. Salida.**

Se enviara por pantalla un mensaje que confirma el registro o indica un error en dicho registro.

## **3.2. Requisitos no funcionales**

### **3.2.1. Requisitos de rendimiento**

El sistema trabajará en una red compuesta de seis computadoras, donde no todas se conectarán simultáneamente. El tiempo de cada transacción no será mayor a 5 segundos.

### **3.2.2. Seguridad**

El sistema constará de contraseñas para el acceso a este, encriptados en alguno de los algoritmos de encriptación conocidos, al igual la base de datos también constará de clave de acceso encriptado.



## Aplicación Web para Automatizar la administración e inventario del negocio el Chompipe en el mercado el BISNE.

---

También constará con diferentes niveles de acceso según el tipo de usuario protegiendo información de gran importancia.  
El sistema llevará un registro de las operaciones realizadas por cada usuario.

### 3.2.3. Portabilidad

La aplicación podrá ser usada en cualquier sistema operativo que tenga un navegador web. El servidor puede ser exportado solo si se cambia al servidor apache por otro lado se maneja en la plataforma Windows



### 13.4. Diccionario de datos

#### Entradas al sistema:

**Datos-del-usuario** = Nombre + Cedula + Direccion + Telefono + Email

Nombre = { Caracter }

Cedula = {Numero}10 + Caracter

Direccion = { Caracter }

Telefono = { Numero } 8

Email= {caracter} + @ {caracter} . {caracter}

**Nuevos-Datos-Usuario** = ( Nombre) + ( Cedula) + ( Direccion) + (Telefono) + (Email )

**Datos-Articulo** = Nombre-art + Unidad-Medida + Estado + ( Categoria ) + ( Proveedor ) + ( Marca ) + Precio-unidad + Costo + Descuento + ( Cantidad ) + (Limite-existencias) + Fecha-adquisicion + (Fecha-caducidad) + (Descripcion)

Nombre-art = { caracter }

Unidad-Medida = [ Unidad | Docena | Metro | Yarda | Libra ]

Estado = [ Bueno | Defectuoso | Regular ]

Categoria = { caracter }

Proveedor = { caracter }

Marca = { caracter }

Precio-unidad= Decimal

Costo = Decimal

Descuento = Decimal

Cantidad = {Numero}

Limite-existencias = {Numero }

Fecha-adquisicion = Fecha



Fecha-caducidad = Fecha

Descripcion = { caracter }

**Nuevos-datos-articulos** = (Nombre-art) +( Unidad-Medida) +( Estado) + ( Categoria ) + ( Proveedor ) + ( Marca ) + (Precio-unidad) +( Costo) + (Descuento) + ( Cantidad ) + (Limite-existencias) + (Fecha-adquisicion) + (Fecha-caducidad) + (Descripcion)

**Datos-Cliente** = Nombre-cli + Cedula +(Telefono) + (Email) + (limite-cred) + Direccion + (Ciudad) + (Tipo)

Nombre-cli = { caracter }

Cedula = {Numero}10 + Caracter

Direccion = { Caracter }

Telefono = { Numero } 8

Email= { caracter } + @ { caracter } . { caracter }

Limite-cred = Decimal

Ciudad = { caracter }

Tipo = { caracter }

**Nuevos-Datos-Cliente** = (Nombre-cli) + (Cedula) +(Telefono) + (Email) + (limite-cred) +( Direccion) + (Ciudad) + (Tipo)

**Datos-Venta** = { No-Factura + Vendedor + Cliente + Fecha-venta + Articulos + Cantidad + (Unidad-medida) + (Precio) + (Costo) + (Descuento) + Tipo-factura } + (Observaciones) +( Desactivar-iva) + ( Aplicar-descuento) + Subtotal + IVA + Total

No-Factura = {Numero}

Vendedor = {Caracter}

Cliente = {Caracter}

Fecha-venta = Fecha

Articulos = {Caracter}

Cantidad = {Numero}



Unidad-medida = [Unidad | Docena | Metro | Yarda | Libra ]

Precio = Decimal

Costo = Decimal

Descuento = Decimal

Tipo-factura = [Contado | Crédito]

Observaciones = {Caracter}

Desactivar-iva = Booleano

Aplicar-descuento = Booleano

Subtotal = Decimal

IVA = Decimal

Total = Decimal

**Datos-Configuracion** = (Configurar-iva) + (Taza-cambio) + (Datos-empresa)

Configurar-iva = Decimal

Taza-cambio = Decimal

Datos-empresa = (nombre-empresa) + (direccion-empresa) + (Telefono-empresa) +  
(ruc)

Nombre-empresa = {Caracter}

Direccion-empresa = {Caracter}

Teléfono-empresa = { Numero } 8

Ruc = { Numero } 6

**Datos-compra** = { Articulos + Proveedor + Cantidad + Precio } + (Observaciones) + Total +  
Comprador + Fecha-compra



Articulos = {Caracter}

Proveedor = {Caracter}

Cantidad = {Numero}

Precio = Decimal

Observaciones = {Caracter}

Total = Decimal

Comprador = {Caracter}

Fecha-compra = Fecha

**Datos-caja** = (Caja-inicial) + ({Otros-gastos}) + ({Otros-ingresos})

Caja-inicial = Decimal

Otros-gastos = {Caracter}

Otros-ingresos = {Caracter}

**Datos-proveedor** = Nombre-proveedor + Cedula + Ciudad + Direccion + (teléfono) +  
(Empresa) + (Email) + (SitioWeb) + (Observaciones)

Nombre-proveedor = {Caracter}

Cedula = {Numero}10 + Caracter

Ciudad = {Caracter}

Direccion = {Caracter}

Teléfono = { Numero } 8

Empresa = {Caracter}

Email = {caracter} + @ {caracter} . {caracter}

SitioWeb = {Caracter}

Observaciones = {Caracter}



**Nuevos-Datos-Proveedor** = (Nombre-proveedor) + (Cedula) +( Ciudad) + (Direccion) +  
(teléfono) + (Empresa) + (Email) + (SitioWeb) + (Observaciones)

Caracter = [ A-Z | a-z ]

Numero = [ 0-9 ]

Decimal = {0-9} + . + {0-9} 2

Fecha = {0-9} 2 + / + {0-9} 2 + / + {0-9} 4

Booleano = [ true | false]

### Salidas del sistema

**Reporte-compras** = Comprador + Fecha-actual + Total-compras + {Fecha-compra + Artículo +  
Cantidad + Precio + Empresa + Total}

Comprador = {Caracter}

Fecha-actual = Fecha

Total-compras = Decimal

Fecha-compra = Fecha

Artículo = {Caracter}

Cantidad = {Numero}

Precio = Decimal

Empresa = {Caracter}

Total = Decimal

**Reporte-artículos** = [Existencias | Proximos-agotarse]

Existencias = Fecha-actual + {No + Nombre + Marca + Precio + Costo + Existencias-art  
+ Descripción + Fecha-adquisición}

Proximos-agotarse = Fecha-actual + {No + Nombre + Marca + Precio + Costo +  
Existencias-art + Descripción + Fecha-adquisición}



Fecha-actual = Fecha

No = {Numero}

Nombre = {Caracter}

Marca = {Caracter}

Precio = Decimal

Costo = Decimal

Existencias = {Numero}

Descripción = {Caracter}

Fecha-adquisición = Fecha

**Reporte-facturas** = [Por-cobrar | Por-rango-fecha | Por-Usuario]

Por-cobrar = Fecha-actual + {No + Fecha-fac + Observaciones + Vendedor + Tipo +  
Estado + Total }

Por-rango-fecha = Fecha-actual + {No + Fecha-fac + Observaciones + +Tipo + Estado +  
Total }

Por-Usuario = {No + Fecha-fact + Observaciones + Total + Tipo + Estado}

Fecha-actual = Fecha

No = {Numero}

Fecha-fac = Fecha

Observaciones = {Caracter}

Vendedor = {Caracter}

Tipo = [Contado | Crédito]

Estado = [Cancelada | Vigente | Pagada]

Total = Decimal

**Reporte-venta** = [Por-rango-fecha | Por-usuario]



Por-rango-fecha = Fecha-actual + Total-ventas { No + Fecha-fact + No-art + No-  
cliente + Cantidad + Precio + Total-precio + No-factura }

Fecha-actual = Fecha

Total-ventas = Decimal

No = {Numero}

Fecha-fact = Fecha

No-art = {Numero}

No-cliente = {Numero}

Cantidad = {Numero}

Precio = Decimal

Total-precio = Decimal

No-factura = {Numero}

Por-usuario = Nombre-usuario + {No-fact + Fecha-fact + Observaciones + Total +  
Tipo + Estado}

Nombre-usuario = {Caracter}

Fecha-fact = Fecha

Observaciones = {Caracter}

Total = Decimal

Tipo = [Contado | Crédito]

Estado = [Cancelada | Vigente | Pagada]