

SISTEMA PARA LA ADMINISTRACIÓN DE INVENTARIOS

LUISA FERNANDA ALDANA GALLO  
HERNAN CAMILO QUINTERO ORREGO

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES  
FACULTAD DE INGENIERÍA  
INGENIERÍA DE SISTEMAS  
BOGOTÁ  
2015

SISTEMA PARA LA ADMINISTRACIÓN DE INVENTARIOS

LUISA FERNANDA ALDANA GALLO  
HERNAN CAMILO QUINTERO ORREGO

PROYECTO DE GRADO

ING. AUGUSTO JOSÉ ÁNGEL MORENO  
DIRECTOR

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES  
FACULTAD DE INGENIERÍA  
INGENIERÍA DE SISTEMAS  
BOGOTÁ  
2015

Nota de Aceptación

---

---

---

---

---

---

---

Presidente del Jurado

---

Jurado

---

Jurado

Bogotá, Noviembre 30 de 2015

## TABLA DE CONTENIDO

	pág.
INTRODUCCIÓN .....	10
1. JUSTIFICACIÓN .....	11
2. OBJETIVOS .....	12
2.1. Generales .....	12
2.2. Específicos .....	12
3. MARCO TEÓRICO.....	13
3.1. METODOLOGÍA DE DESARROLLO.....	14
3.1.1. Extreme Programing Explainend XP .....	14
3.2. Base de Datos .....	17
3.2.1. Sql – Structure Query Language.....	17
3.2.1.1. Ddl - Data Definition Language.....	17
3.2.1.2. Dml - Data Manipulation Language.....	18
3.2.1.3. Dcl - Data Control Language.....	18
3.2.2. Normalización.....	18
3.2.2.1. Primera forma normal (1fn).....	19
3.2.2.2. Segunda Forma Normal (2fn).....	19
3.2.2.2. Tercera Forma Normal (3fn). .....	20
3.2.2.3. Forma Normal Boyce-Codd (Fnbc).....	21
3.2.2.4. Cuarta Forma Normal (4fn).....	21
3.2.2.5. Quinta Forma Normal (5fn).....	22
3.2.3. Reglas De Codd. ....	23
3.2.3.1. Regla 1 - De la información.....	23
3.2.3.2. Regla 2 - Del acceso garantizado. ....	23
3.2.3.3. Regla 3 - Tratamiento sistemático de valores nulos.....	23
3.2.3.4. Regla 4 - Catálogo en línea basado en el modelo relacional. ....	24

3.2.3.5.	Regla 5 - Regla del sub-lenguaje de datos completo.....	24
3.2.3.6.	Regla 6 - de actualización de vista.....	24
3.2.3.7.	Regla 7 - inserción, actualización y borrado de alto nivel.....	24
3.2.3.8.	Regla 8 - Independencia física de datos. ....	25
3.2.3.9.	Regla 9 - Independencia lógica de datos. ....	25
3.2.3.10.	Regla 10 - Independencia de integridad.....	25
3.2.3.11.	Regla 11 - Independencia de distribución. ....	25
3.2.3.12.	Regla 12 - De la no sub-versión.....	25
3.3.	LENGUAJE DE PROGRAMACIÓN .....	26
3.3.1	VISUAL STUDIO .NET .....	26
4.	INGENIERÍA DEL PROYECTO.....	27
4.1	Descripción de la situación actual.....	27
4.2.	REQUERIMIENTOS DE LA INFORMACIÓN .....	27
4.2.1.	Funcionales. ....	27
4.2.2.	No funcionales.....	28
4.3.	MODELAMIENTO DEL SISTEMA.....	29
4.4	DESCRIPCIÓN DEL SISTEMA.....	29
4.4.1	Aplicación web.....	30
4.4.2	Interface de ingreso .....	30
5.	EVALUACIÓN ECONÓMICA DEL PROYECTO .....	32
5.1	Presupuesto Pre operativo .....	32
5.2	Presupuesto de Infraestructura .....	32
6.	BENEFICIOS DE LA IMPLEMENTACIÓN .....	33
6.1	Beneficios Operacionales .....	33
6.2	Beneficios de Gestión.....	33
6.3	Beneficios Estratégicos .....	33
6.4	Beneficios de infraestructura .....	34

6.5 Beneficios de TI.....	34
7. ALCANCE DEL PROYECTO .....	35
7.1 Lo que se va a lograr .....	35
7.2 Lo que no se va a lograr .....	35
8. LIMITACIONES DEL PROYECTO .....	36
9. CRONOGRAMA.....	37
10. RECOMENDACIONES .....	38
BIBLIOGRAFÍA.....	40

## LISTA DE FIGURAS

Figura 1. Ejemplo tabla sin normalización .....	19
Figura 2. Ejemplo tabla en primera forma normal .....	19
Figura 3. Ejemplo tabla en segunda forma normal .....	20
Figura 4. Ejemplo tabla en tercera forma normal .....	20
Figura 5. Ejemplo tabla en cuarta forma normal .....	22
Figura 6. Imagen de autenticación de la aplicación .....	30
Figura 7. Imagen de la Interface principal.....	31

## LISTA DE TABLAS

Tabla 8. Presupuesto pre operativo .....	32
Tabla 9. Presupuesto de Infraestructura.....	32

## LISTA DE ANEXOS

ANEXO A Ficha de caso de uso 1 .....	42
ANEXO B Ficha de caso de uso 2 .....	43
ANEXO C Ficha de caso de uso 3 .....	44
ANEXO D Ficha de caso de uso 4 .....	45
ANEXO E Ficha de caso de uso 5 .....	46
ANEXO F Ficha de caso de uso 6 .....	47
ANEXO G Ficha de caso de uso 7 .....	48
ANEXO H Diccionario de datos .....	49
ANEXO I Modelo Entidad Relación.....	52

## INTRODUCCIÓN

El camino de las nuevas tecnologías, que a través de los años se ha convertido más bien en una carrera de obstáculos, ha traído al mercado global múltiples posibilidades con respecto al manejo de la información, con enfoques tan pertinentes como la mejora del uso del tiempo, los recursos y las propias vidas de los involucrados.

Esta carrera se ha convertido en un evento que permea a toda la población, logrando así que desde chicos hasta grandes estén interesados en todos sus avances y sean partícipes de ellos.

Uno de los obstáculos ha sido el manejo ideal de la información de los recursos, materiales o intelectuales de una persona o empresa, tanto así que los avances tecnológicos han logrado cambiar las formas y los elementos para dicho fin, disminuyendo el tamaño y aumentando su capacidad a la vez, permitiendo tener grandes cantidades de información almacenada en dispositivos tan pequeños como una moneda.

De tener grandes listas en grandes libros llevados de forma manual, agendas completas con información personal y todo tipo de repositorios que requieren grandes espacios físicos de almacenamiento, y de tener que realizar búsquedas engorrosas y demoradas, se pasa a tener cientos de veces la misma información, almacenada en un dispositivo de mucho menor tamaño y con la gran posibilidad de encontrar un dato en cuestión de segundos.

De esta manera y con estas posibilidades se decide mejorar y aportar en el crecimiento comercial de una pequeña empresa, enfocando de forma responsable los conocimientos y la educación profesional adquirida, aportándolos a nuestra sociedad

Así que en el presente trabajo se desarrollan las diferentes etapas del diseño de una aplicación de inventario, teniendo en cuenta la importancia a nivel económico que tiene el uso adecuado de un software para su fin, y a su vez se pueden ver reflejadas las limitaciones funcionales de un software sin etapas definidas, ni metodologías para su correcta implementación.

## 1. JUSTIFICACIÓN

El diseño de software se realiza como objetivo principal en la puesta en práctica de los conocimientos adquiridos en las aulas universitarias y como opción de grado.

Se busca dar solución a problemas evidenciados para el manejo y la administración del inventario a través del software estudiado, generando nuevos procesos y soluciones a través del análisis y diseño de todos los módulos involucrados en el software.

Al realizar el levantamiento y análisis de información se detectan grandes fallas en el diseño de la base de datos y de las interfaces funcionales, además de encontrar que la aplicación no cuenta con funcionalidades ideales para una administración adecuada del inventario. Es así que se hace evidente la importancia de realizar el diseño de software en base a una metodología definida y con los procesos adecuados, ya que su correcto uso e implementación permitirán contar con software de grandes prestaciones y de ideal rendimiento, haciendo a la empresa un participante más de esta gran carrera.

Se espera lograr mejoras significativas en la usabilidad y funcionalidad del software de inventarios, permitiendo a los usuarios finales un uso intuitivo, integrando nuevas herramientas, que optimicen el análisis de la información y la administración de la misma.

## **2. OBJETIVOS**

### **2.1. Generales**

Diseñar e implementar el software de inventario de la empresa Pakis Medical, buscando mejorar sus funcionalidades y la calidad de la información a través de nuevas estructuras de bases de datos y su administración, con nuevas herramientas para el manejo de los datos.

### **2.2. Específicos**

- Revisar la metodología de administración de inventarios en uso para así poder determinar sus fallas, y de esta manera obtener los requerimientos a atacar.
  
- Realizar el levantamiento de la información para poder analizar y construir de manera adecuada un nuevo modelo de bases de datos.
  
- Diseñar en base al levantamiento y análisis de información, los diferentes módulos de presentación, mejorando su usabilidad y funcionalidad.
  
- Implementar la solución final, permitiendo al área encargada trabajar sobre ella para sacar provecho de sus características.

### 3. MARCO TEÓRICO

La gestión de inventarios a través del tiempo se ha convertido en una herramienta vital para la administración de una empresa, de esta forma ha estado sujeta a cambios importantes y llenos de tecnología, cambios que han logrado mejorar los índices contables al interior de muchas compañías, generando también nuevos modelos de administración de inventarios tanto para empresas públicas como privadas.

Las nuevas tecnologías y su globalización son grandes contribuyentes a este tema permitiendo no solo mejorar la administración de inventario a través de nuevas herramientas, más confiables, ágiles y poderosas en comparación con los libros, o las hojas de cálculo, sino también la posibilidad de obtener mediante estas información adicional y vital que con las anteriores no era posible. La movilidad es una característica que se ha convertido en la punta del iceberg, el futuro es ahora, con la posibilidad de tener acceso a la información casi desde cualquier parte del mundo, y la gestión de inventarios a través de software no escapa a ello.

Grandes empresas a nivel mundial implementan soluciones de software robustas para administrar de la mejor manera, sus productos en un inventario general, y esta buena práctica las hace aunque no parezca, mucho más competitivas; Para las micro empresas la mecánica es diferente, porque debido a su tamaño muchas no ven la necesidad de administrar óptimamente su inventario, o creen que la herramienta que están usando les da todo lo que necesitan, por esta razón y otras más es necesario mejorar las herramientas utilizadas al interior de estas empresas, dando así la posibilidad de tener las mismas y más funciones, la facilidad de acceder a la información desde cualquier punto y la innovación requerida por el mercado, logrando como resultado mejoras en los procesos de inventarios contables, reducción en tiempos de búsqueda, optimización de recursos y minimización de riesgos inherentes al área, traduciéndose todo esto en aumento de ganancias para dichas empresas.

Siendo las pymes, compañías en crecimiento, se hace necesario que este crecimiento vaya acorde no solo con su modelo de negocio, sino también con todo lo que permea a ese modelo de negocio en el mercado, una de esas cosas es la tecnología, e incluso las técnicas para evolucionar a la par de esa tecnología, por esta razón la implementación de una solución de software para administrar

inventario será un cambio sustancial en la manera en cómo se desarrollará la dinámica al interior de la empresa.

El desarrollo de la aplicación se realizó con la guía de la metodología de desarrollo de software XP extreme Programming, además contará con una base de datos construida en SQL SERVER 2014 la cual contará con la 3ra forma de normalización, el lenguaje de desarrollo de la aplicación es Visual Studio .NET.

### **3.1. METODOLOGÍA DE DESARROLLO.**

#### **3.1.1. Extreme Programming Explained XP**

Es una metodología ágil de la ingeniería de software, que es flexible en la medida en que permite los cambios sobre la marcha, su adaptabilidad a ellos es su principal fortaleza. El minimizar la búsqueda total de los requisitos al inicio del proyecto, le permite invertir el tiempo de manera diferente y así lograr optimizar los procesos de desarrollo y estar preparado para los cambios, siendo esta una forma de ver la realidad de las necesidades más fielmente, que si se obtuvieran todos los requisitos iniciando el proyecto.

Esta metodología ágil se fundamenta en mejorar y potenciar las relaciones interpersonales como herramienta principal en la obtención del éxito del desarrollo de software, incentivando el trabajo en equipo y prestando atención a las necesidades de aprendizaje de los desarrolladores.

Las principales características de la programación extrema son: simplicidad, comunicación, retroalimentación, coraje y respeto, este último añadido en la segunda edición de Extreme Programming.

La simplicidad es el pilar fundamental de la programación extrema. Se realizan diseños simplificados para que el desarrollo sea más ágil y facilitar así su mantenimiento. De la misma manera se aplica esta característica en la documentación, todo el código debe documentarse de manera justa.

La comunicación por su parte se realiza de múltiples formas, en general la comunicación entre programadores mejora en la medida en que el código sea más simple. El desarrollo o programación en parejas permite que la comunicación para con todo el equipo de trabajo sea la ideal, sin importar el tamaño del proyecto la programación por pares permite que todo el equipo este enterado de los

pormenores del mismo. La constante comunicación con el cliente es permanente, ya que desde el inicio se vuelve a este parte del equipo de desarrollo. El cliente puede decidir sobre las características del desarrollo que se deben priorizar y también debe estar disponible para resolver las dudas generadas en el proceso.

La retroalimentación es un derivado de la buena comunicación, ya que en todo momento el cliente está integrado y se puede conocer su opinión sobre el estado del proyecto en tiempo real. La programación en XP realiza ciclos cortos de programación para mostrar resultados, permitiendo esto recibir retroalimentación por parte de los involucrados y así no tener que corregir errores cuando el desarrollo se encuentre avanzado, sino al final de cada ciclo poder resolver y así poder concentrarse en las partes importantes.

El coraje o la valentía les permiten a los programadores la reconstrucción de su código cuando lo consideren necesario. Pueden revisar los desarrollos existentes y realizar los cambios, si con esto se asegura que los cambios futuros se implementaran fácilmente. La valentía también implica saber cuándo se debe desechar un código o quitar códigos fuentes obsoletos, acá no debe importar cuanto tiempo y esfuerzo se invirtió en este código. La persistencia es tanta valentía como tiene el desarrollador para resolver los problemas que se le presenten, tomándole muchas veces días enteros, pero terminando siempre encontrando una solución.

El respeto acoge a todo el grupo de personas involucradas en el proyecto, los clientes, directores, desarrolladores y todos los que hagan parte de él. Todos y cada uno de los participantes hacen de su trabajo una fuente de respeto, los desarrolladores no cambiarán código que arruinen las pruebas y que a su vez esto demore la entrega de los ciclos. El cliente no debe cambiar de parecer con sus observaciones. Todos respetan el trabajo del otro, y todos se enfocan en alcanzar resultados de alta calidad.

El desarrollo bajo XP cuenta con unas características fundamentales que le permiten ser reconocida entre las metodologías ágiles como una de las mejores sino la mejor. También cuenta con prácticas básicas, las cuales deben seguirse de manera estricta para así asegurar el funcionamiento y el resultado final del proyecto

El equipo completo indica que deben formar parte del equipo todas aquellas personas que tengan algo que ver con el proyecto, incluidos el cliente y el responsable del proyecto.

La planificación indica que se deben hacer las historias de usuario, para con esta información construir mini versiones del producto. La planificación se debe revisar continuamente.

El test del cliente es una propuesta de este último sobre las pruebas a realizar para validar cada una de estas mini versiones.

Las versiones pequeñas son estas mini versiones que son entregadas al usuario en pocas semanas, permitiéndole a este poder revisar avances entendibles y prácticos funcionales y no líneas de código sin mucho sentido para él.

El diseño simple es fundamental en el funcionamiento ideal de la metodología, ya que se mantiene un diseño con lo mínimo necesario pero con lo más imprescindible y se mantiene el código lo más sencillo posible.

La programación por parejas indica que el desarrollo lo deben realizar dos personas sentadas frente a un mismo ordenador, realizando cambios de parejas con frecuencia. Estos cambios se sugieren sean diarios, así todo el equipo de desarrolladores conocerá los pormenores del código.

El desarrollo guiado por las pruebas automáticas sugiere la construcción de programas de pruebas automáticas que se estén ejecutando con frecuencia para así validar los desarrollos. Entre más pruebas automáticas se ejecuten al código mucho mejor.

La integración continua es la práctica que más resultados positivos nos muestra, ya que de tenerse un ejecutable del proyecto que funcione, se debe agregar a este cada parte del desarrollo que vaya siendo funcional por pequeña que sea, se debe recompilar y probarse. Tener una versión en stand-by por demasiado tiempo mientras se agrega un trozo grande de funcionalidades es una falla ya que en el momento de un error no se podrá saber con facilidad que parte del código nuevo agregado es la que está fallando, generando demoras en la planificación.

El código es de todos es el resultado de la programación en parejas, ya que todos pueden y deben conocer y poder tocar cualquier parte del código.

Las normas de codificación sugieren que debe haber un solo tipo de codificación, de tal manera que parezca que el código ha sido hecho por una única persona y sea entendible para cualquiera del grupo que lo vea.

Las metáforas indican que se deben buscar frases o nombres para definir cómo funcionan las distintas partes de un programas y que de esta manera al referirse a una de estas metáforas, todos puedan entender de qué se trata y no se preste para malos entendidos, ejemplo, "Recolector de Basura".

Por último el ritmo sostenible permite que el trabajo se pueda mantener indefinidamente. No deben existir días donde no se realice nada y tampoco días

donde se excedan los esfuerzos. Se debe trabajar según lo planificado, haciendo los esfuerzos necesarios por terminar cada historia de usuario y poder entregar así su respectiva mini versión.

### **3.2. Base de Datos**

Las bases de datos son archivos de almacenamiento sistematizado que permiten guardar datos de forma relacional y categorizada, vinculadas por algún tipo de criterio o dato en común que permite su clasificación y ordenamiento; Actualmente con los avances tecnológicos, han colocado a las bases de datos como un componente primordial para la unificación de la información cuando esta es consultada o manipulada por varias fuentes.

Para la gestión de las bases de datos existen aplicaciones o software diseñados para elaborar esta tarea, denominados Sistemas Gestores de Bases de Datos (SGBD) o en inglés Data Base Management System (DBMS), que suministran una serie de herramientas que permiten la administración de la estructura de los objetos y los datos de la base de datos.

#### **3.2.1. Sql – Structure Query Language.**

Lenguaje de alto nivel compuesto por comandos, cláusulas, operadores y funciones de agregado, que permite el ingreso, actualización y recuperación de la estructura de los objetos y de los datos de las bases de datos de tipo relacional por medio del manejo del álgebra y el cálculo relacional, definido como estándar en la mayoría de los SGBD. SQL maneja 3 tipos de comandos:

##### **3.2.1.1. Ddl - Data Definition Language.**

Conocido en español como lenguaje de definición de datos, es un lenguaje que contiene los comandos necesarios para la gestión sobre las bases de datos, sus características y la estructura de sus objetos (tablas, índices, relaciones, etc.).

### **3.2.1.2. Dml - Data Manipulation Language.**

Conocido en español como lenguaje de manipulación de datos, es un lenguaje suministrado por el SGDB que contiene los comandos requeridos y permiten al usuario la manipulación de los datos contenidos en la base de datos.

### **3.2.1.3. Dcl - Data Control Language.**

Conocido en español como lenguaje de control de datos, es un lenguaje que proporciona los comandos necesarios por medio del SGDB para la gestión de los privilegios de los datos y los objetos de la base de datos.

## **3.2.2. Normalización.**

La normalización es el proceso mediante el cual se realiza la aplicación de una serie de reglas que permiten mejorar la integridad y estructura de los datos almacenados dentro de la base de datos, minimizando la redundancia de la información al momento del diseño del modelo relacional, ofreciendo estabilidad, agilidad y simpleza al momento de la manipulación de los mismos; En la actualidad la normalización está dividida o compuesta por 6 formas normales:

- Primera Forma Normal (1FN)
- Segunda Forma Normal (2FN)
- Tercera Forma Normal (3FN)
- Forma Normal de Boyce-Codd (FNBC)
- Cuarta Forma Normal (4FN)
- Quinta Forma Normal (5FN)

**Figura 1. Ejemplo tabla sin normalización**

USUARIOS				
nombre	empresa	direccion_empresa	url1	url2
Joe	ABC	1 Work Lane	abc.com	xyz.com
Jill	XYZ	1 Job Street	abc.com	xyz.com

Fuente: <http://www.cs.upc.edu/~bcasas/docencia/pfc/NormalitzacioBD.pdf>

### 3.2.2.1. Primera forma normal (1fn).

Su objetivo es la eliminación de las columnas que tengan el mismo tipo de dato y colocarse en tablas separadas y estas a su vez deben tener una llave primaria que identifique al dato, con ello se a la atomicidad de los atributos y la supresión de los encabezados de columna múltiple.

**Figura 2. Ejemplo tabla en primera forma normal**

USUARIOS			
nombre	empresa	direccion_empresa	url
Joe	ABC	1 Work Lane	abc.com
Joe	ABC	1 Work Lane	xyz.com
Jill	XYZ	1 Job Street	abc.com
Jill	XYZ	1 Job Street	xyz.com

Fuente: <http://www.cs.upc.edu/~bcasas/docencia/pfc/NormalitzacioBD.pdf>

### 3.2.2.2. Segunda Forma Normal (2fn).

Para aplicar esta regla la base de datos debe estar en 1FN y determina que los atributos que no tengan ningún tipo de dependencia directa con la llave primaria de la tabla para ser identificados deben ser separados en una nueva tabla y enlazados por sus índices únicos, con esto se elimina el exceso de datos en los registros.

**Figura 3. Ejemplo tabla en segunda forma normal**

USUARIOS			
userID	nombre	empresa	direccion_empresa
1	Joe	ABC	1 Work Lane
2	Jill	XYZ	1 Job Street

+

URLS		
urlID	relUserID	url
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

Fuente: <http://www.cs.upc.edu/~bcasas/docencia/pfc/NormalitzacioBD.pdf>

### 3.2.2.2. Tercera Forma Normal (3fn).

Para aplicar esta regla la base de datos debe estar en 2FN y determina que los atributos que no sean llave y su dependencia no sea directamente funcional sino con una columna no llave de la tabla, deben ser separados en una nueva tabla y enlazados por sus índices únicos, con esto se elimina errores en la manipulación de los datos o registros.

**Figura 4. Ejemplo tabla en tercera forma normal**

USUARIOS		
userID	nombre	relEmpresaID
1	Joe	1
2	Jill	2

EMPRESAS		
emprID	empresa	direccion_empresa
1	ABC	1 Work Lane
2	XYZ	1 Job Street

URLS		
urlID	relUserID	url
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

Fuente: <http://www.cs.upc.edu/~bcasas/docencia/pfc/NormalitzacioBD.pdf>

### 3.2.2.3. Forma Normal Boyce-Codd (FNBC).

Esta regla se cumple si está en 3FN y cuando los atributos de las tablas son totalmente dependientes de la llave primaria, en pocas palabras se han eliminado los atributos que no tengan relación directa con el atributo que tiene establecida la llave.

### 3.2.2.4. Cuarta Forma Normal (4FN).

La aplicación de esta regla en la base de datos no es obligatoria para los diseñadores, en caso de emplear esta norma, las tablas deben estar en 3FN como mínimo y no se obliga a cumplir con la forma normal de Boyce-Codd; La función de esta norma es determinar las relaciones de varios con varios, con el fin de tener una óptima relación entre las tablas y sus datos.

**Figura 5. Ejemplo tabla en cuarta forma normal**

USUARIOS		
userID	nombre	relEmpresalD
1	Joe	1
2	Jill	2

EMPRESAS		
emprID	empresa	direccion_empresa
1	ABC	1 Work Lane
2	XYZ	1 Job Street

URLS		
urlID	relUserID	url
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

RELACION_URLS		
relacionUrlID	relUserID	relUrlID
1	1	1
2	1	2
3	2	1
4	2	2

Fuente: <http://www.cs.upc.edu/~bcasas/docencia/pfc/NormalitzacioBD.pdf>

### 3.2.2.5. Quinta Forma Normal (5fn).

La aplicación de esta regla no es relevante, su función consiste en la reconstrucción de la estructura original de la tabla de datos a la cual se le aplicó la normalización con el fin de verificar que no tenga atributos extraños, emplear esta norma es considerada una buena práctica para bases de datos estructuralmente pequeñas por la complejidad del proceso y según la necesidad.

### **3.2.3. Reglas De Codd.**

Conjunto de reglas que permiten dar mayor fiabilidad al proceso de diseño de bases de datos relacionales con el fin de evitar que se conviertan en solamente tablas de almacenamiento de datos, estas reglas son difíciles de seguir, por lo cual en la actualidad se estipula el grado de una base de datos relacional según la cantidad de reglas cumplidas.

Las reglas de Codd están conformadas por 12 ítems, que permiten dar un mayor perfilamiento a la práctica del diseño y administración de bases de datos.

#### **3.2.3.1. Regla 1 - De la información.**

Los datos contenidos dentro de las tablas en una base de datos relacional están en valores de posiciones de columnas y filas, definiendo que la información se represente a nivel lógico; También se define que el valor "nulo" es considerado un valor con 2 posibles interpretaciones "desconocido" o "no aplicable".

#### **3.2.3.2. Regla 2 - Del acceso garantizado.**

Los valores atómicos contenidos en las tablas deben ser accesibles unívocamente por medio de un direccionamiento lógico donde se especifica el nombre de tabla, nombre de la columna y su llave primaria, haciendo cumplir el requisito de llaves primarias dentro de las tablas y que los datos que las conforman sean únicos y no nulos.

#### **3.2.3.3. Regla 3 - Tratamiento sistemático de valores nulos.**

Los valores nulos deben estar definidos dentro de los SGDB (Sistema Gestor de Bases de Datos) como valores que represente la ausencia de datos o campos no aplicables dentro de las tablas de las bases de datos, independientemente de los tipos de datos que represente vacío o nulidad de información.

#### **3.2.3.4. Regla 4 - Catálogo en línea basado en el modelo relacional.**

El SGDB (Sistema Gestor de Bases de Datos) debe contar con un catálogo en línea que describa y permita la consulta, como la modificación de la estructura de la base de datos y sus metadatos por parte de los usuarios autorizados.

#### **3.2.3.5. Regla 5 - Regla del sub-lenguaje de datos completo.**

El SGDB (Sistema Gestor de Bases de Datos) debe soportar por lo menos un lenguaje que contenga una cadena completa de caracteres que permita la creación de sentencias expresables con una sintaxis coherente, soportando:

- Definición de datos
- Definición de vistas
- Manipulación de datos (interactiva y por programa)
- Restricciones de integridad
- Autorización
- Fronteras de transacción (comienzo, cumplimiento y vuelta atrás).

#### **3.2.3.6. Regla 6 - de actualización de vista.**

Las vistas implementadas en la base de datos y que teóricamente son actualizables, también deben ser actualizables por el SGDB (Sistema Gestor de Bases de Datos).

#### **3.2.3.7. Regla 7 - inserción, actualización y borrado de alto nivel.**

Define que el sistema debe permitir no solo la consulta de varios registros, también la inserción, actualización y/o borrado de múltiples duplas, como parte del proceso de manipulación de datos en alto nivel.

### **3.2.3.8. Regla 8 - Independencia física de datos.**

Los datos se deben poder manipular de forma independiente, sin afectar o alterar a nivel lógico la estructura física de las aplicaciones que se usan para su manipulación.

### **3.2.3.9. Regla 9 - Independencia lógica de datos.**

La estructura y metadatos de los objetos y de la misma base de datos se deben poder manipular de forma independiente, sin afectar o alterar a nivel lógico la estructura física de las aplicaciones que se usan para su manipulación.

### **3.2.3.10. Regla 10 - Independencia de integridad.**

Las configuraciones para conservar la integridad de la información de la base de datos deben ser creados en sub-lenguaje, también deben ser almacenados en el catálogo de la misma y ser independientes a los programas o aplicaciones que se usen para su manipulación.

### **3.2.3.11. Regla 11 - Independencia de distribución.**

El tipo de distribución realizado a la base de datos (por localización, por fragmentación o por replicación), debe ser independiente e invisible a los usuarios de la misma, ya que una base de datos centralizada funciona con las mismas sentencias que una distribuida.

### **3.2.3.12. Regla 12 - De la no sub-versión.**

Para los SGDB (Sistema Gestor de Bases de Datos) que tengan interfaz de bajo nivel, no podrán utilizar dicha interfaz para alterar la base de datos, todo tipo de

modificación sobre la misma o su estructura debe ser realizado por medio de la interfaz de alto nivel dispuesta por el sistema.

### **3.3. LENGUAJE DE PROGRAMACIÓN**

#### **3.3.1 VISUAL STUDIO .NET**

Es un completo grupo de herramientas para desarrollo y construcción de aplicaciones web ASP, servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Una de sus herramientas es los formularios Web Forms, que son una tecnología ASP.NET y se utiliza para crear páginas web programables.

## **4. INGENIERÍA DEL PROYECTO**

### **4.1 Descripción de la situación actual.**

La empresa no cuenta con alguna aplicación especializada en el manejo de inventarios, por tanto la administración de sus productos se lleva a cabo con una hoja de cálculo donde se ingresan sin algún orden o parámetros específicos los productos existentes. La actualización de esta hoja de cálculo se realiza sin alguna periodicidad y tampoco cuenta con ningún tipo de seguridad de la información. La búsqueda de un producto se realiza mediante la búsqueda propia de la hoja de cálculo, generando varios resultados, convirtiéndose en un proceso poco práctico además de demorado debido a la gran cantidad de registros que tiene la hoja de cálculo.

Realizar y/o obtener algún tipo de informe es prácticamente imposible, ya que la información debe ser extraída manualmente del archivo y organizada de tal manera que se pueda presentar este informe, sin contar con que las cantidades de cada producto no son claras y suelen contener errores.

La relación de cada producto con su proveedor no existe y tampoco se cuenta con una relación del producto vendido y su cliente, esto hace que sea más difícil administrar de manera óptima el inventario y así mismo no se puedan obtener los beneficios de una ideal administración.

### **4.2. REQUERIMIENTOS DE LA INFORMACIÓN**

#### **4.2.1. Funcionales.**

- El sistema cuenta con una versión web y con su respectiva base de datos para el almacenamiento de la información.
- La aplicación debe permitir el acceso al usuario, según el rol que le corresponda por medio de un formulario de acceso que valide sus privilegios.
- La aplicación debe permitir que el usuario realice búsquedas según el criterio que desee por medio del módulo de búsquedas.

- La aplicación debe permitir que el usuario según su rol adicione y actualice información de los productos en la base de datos diseñada para tal fin
- La aplicación debe permitir que el usuario según su rol adicione y actualice información de los proveedores y clientes en la base de datos diseñada para tal fin.
- La aplicación debe permitir que el usuario según su rol cree y actualice información de los usuarios del sistema en la base de datos diseñada para tal fin.
- La aplicación debe tener una interfaz que permita a los usuarios, la visualización y validación de la información capturada de forma ágil y coherente.
- La aplicación debe permitir al usuario según su rol generar informes específicos por el módulo de informes.
- La aplicación debe generar alertas cada vez que se acceda a la misma sobre stock de productos bajos.

#### **4.2.2. No funcionales.**

- El sistema debe permitir la conexión de múltiples usuarios sin afectar la operatividad y eficacia del mismo.
- La aplicación debe contar con la validación de acceso al sistema por medio de usuarios específicos y contraseñas encriptadas.
- El tiempo de implementación del sistema será gradual, de acuerdo a los cambios o actualizaciones que desee efectuar el usuario.
- La aplicación debe permitir acceder a través de cualquier navegador de internet.
- La aplicación debe contar con la posibilidad de exportar digitalmente o a medios impresos la información generada en los informes.

### **4.3. MODELAMIENTO DEL SISTEMA**

El modelamiento del sistema se construye de acuerdo a los perfiles de usuarios que accederán al sistema. Con casos de uso se describe el paso a paso del proceso de ingreso y el uso de los diferentes módulos para la administración del inventario a través de la solución.

- CU-01 Técnico – (Ver anexo A)
- CU-02 Analista – (Ver anexo B)
- CU-03 Técnico – (Ver anexo C)
- CU-04 Analista – (Ver anexo D)
- CU-05 Analista – (Ver anexo E)
- CU-06 Analista – (Ver anexo F)
- CU-07 Analista – (Ver anexo G)

### **4.4 DESCRIPCIÓN DEL SISTEMA**

El desarrollo y la implementación del sistema permiten realizar una ideal administración del inventario de la empresa, ya que cuenta con la posibilidad de ingresar toda la información a una base de datos a través de los diferentes módulos y de la misma manera visualizarla y modificarla según corresponda. Además de tener la posibilidad de generar informes con los datos almacenada en la base de datos que a su vez será diseñada según la medida de las necesidades del usuario. Todo esto se logra a través del entorno gráfico de la aplicación web.

El sistema cuenta don dos partes

- Aplicación web
- Base de datos

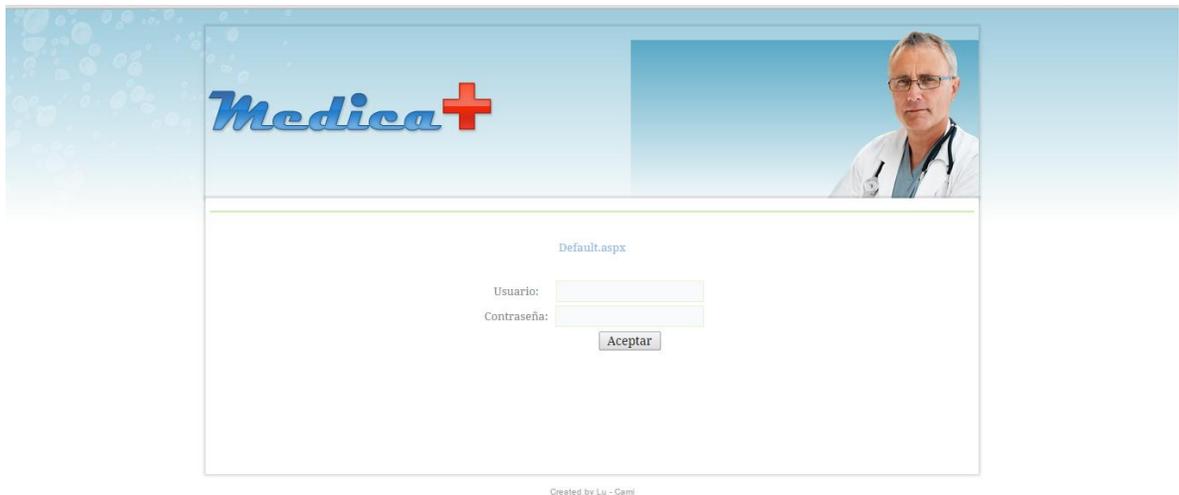
#### 4.4.1 Aplicación web.

La aplicación está diseñada en lenguaje .NET, implementado Web Forms logrando así ser accesible desde la web, se compone por un menú que cuenta con diversas opciones para la administración de inventario. Desde las diferentes opciones se puede buscar, ingresar, actualizar y generar informes de cualquier tipo de producto susceptible de estar en el inventario, además de poder buscar, ingresar y actualizar la información de los clientes y proveedores adyacentes al actividad económica de la empresa y por ultimo cuenta con la opción de crear los usuarios con su respectivo rol para ingresar al sistema de administración. Toda la información manipulada a través de los diferentes módulos y opciones será ingresada en la base de datos diseñada para el manejo de toda la información.

#### 4.4.2 Interface de ingreso

Esta interface cuenta en la ventana principal con un formulario básico de acceso con dos campos (usuario y contraseña) y un botón "Aceptar" el cual será el encargado de enviar la información para validar las credenciales en la base de datos. En la parte superior se ubicaran los logos correspondientes a la empresa.

**Figura 6. Imagen de autenticación de la aplicación**



Fuente: Imagen tomada de la aplicación

Figura 7. Imagen de la Interface principal



Fuente: Imagen tomada de la aplicación

En esta ventana se visualizan todas las opciones con las que cuenta el usuario según el rol que corresponda. En la parte superior se mantendrán los logos de la empresa, e inmediatamente debajo de estos está el menú con las opciones pertinentes para la administración del inventario, en la parte central está el espacio donde se visualizará la información generada por las búsquedas y donde se ubicarán los contenidos correspondientes a cada opción.

## 5. EVALUACIÓN ECONÓMICA DEL PROYECTO

Los valores presentados como parte del proyecto, son valores estimados y subjetivos que se presentan en la medida en que la empresa tuviera que incurrir en esta inversión, ya que la misma cuenta con la infraestructura tecnológica definida para la realización del proyecto.

### 5.1 Presupuesto Pre operativo

**Tabla 1. Presupuesto pre operativo**

PRESUPUESTO PRE OPERATIVO			
CONCEPTO	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Escritorio de Trabajo	1	\$ 400.000	\$ 400.000
Silla para Escritorio	1	\$ 180.000	\$ 180.000
Archivador	1	\$ 150.000	\$ 150.000
TOTAL PRESUPUESTO			\$ 730.000

### 5.2 Presupuesto de Infraestructura

En la siguiente tabla se puede observar el factible presupuesto de costos de la infraestructura necesaria para la implementación de la solución de software.

**Tabla 2. Presupuesto de Infraestructura**

PRESUPUESTO DE INFRAESTRUCTURA			
CONCEPTO	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Equipo de Computo	1	\$ 1.200.000	\$ 1.200.000
Servidor de Aplicación y Bases de datos	1	\$ 5.000.000	\$ 5.000.000
TOTAL PRESUPUESTO			\$ 6.200.000

## **6. BENEFICIOS DE LA IMPLEMENTACIÓN**

### **6.1 Beneficios Operacionales**

Se reducen los tiempos de respuesta frente a la búsqueda de un producto de inventario, mejorando también a la vez la utilización de los recursos tecnológicos disponibles.

Permite el acceso a la información a través de la web, generando movilidad en la información haciendo que la toma de decisiones sea más fácil.

Genera reportes históricos de productos, proveedores, clientes y usuarios del sistema, permitiendo la disponibilidad de la información.

### **6.2 Beneficios de Gestión**

Se mejora sustancialmente la administración del inventario, generando este a su vez, reducción en las compras de productos. Se mejora la toma de decisiones en la medida en que la información generada es más exacta y se cuentan con más parámetros de análisis.

### **6.3 Beneficios Estratégicos**

Mantener una buena administración del inventario a través de una herramienta dispuesta para tal fin permite a la empresa basar sus estrategias de marca y posicionamiento en las cifras almacenadas en el sistema, las cuales indican cantidades de productos vendidos y la rotación de los mismos.

Administrar un inventario a través de la herramienta permite no incurrir en costos innecesarios en la compra de productos e incluso permite conocer sobre los proveedores y clientes potenciales de negocio.

#### **6.4 Beneficios de infraestructura**

La implementación de la solución de software permitirá tener más seguridad que la prestada por una hoja de cálculo para el almacenamiento de la información, ya que la definición de roles de usuario minimizará los riesgos de pérdida de información y/o alteración de la misma.

El almacenamiento en una base de datos permite mantener la información resguardada de riesgos tanto humanos con tecnológicos, brindando seguridad a la empresa.

#### **6.5 Beneficios de TI**

Logra para el área de ventas y administración mejoras sustanciales en la medida en que la modernización del método y la tecnología utilizada para llevar el inventario, resultara a futuro en ser la primera piedra para una actualización mayor, convirtiéndose así en un área vital para la empresa.

## **7. ALCANCE DEL PROYECTO**

La realización de este proyecto pretenderá desarrollar e implementar una herramienta de gestión de inventarios que pueda ser accedida a través de una plataforma web, que permitirá mejorar la administración del inventario, generando un directo impacto en el área de ventas y compras, mejorando los procesos de búsqueda de artículos e información sobre los mismos. Minimizará los riesgos de pérdida de información ya que esta estará almacenada de forma segura y a largo plazo generará reducción de tiempos en los procesos que interactúen con el inventario.

### **7.1 Lo que se va a lograr**

Mejora en la administración del inventario, generando nuevos y mejores procesos.

Complementar la información de los productos al poder contar con datos adicionales como proveedor y cliente.

Almacenar la información referente al inventario en un único repositorio.

Asegurar la información referente a los productos, proveedores y clientes, además de la información de usuarios manteniendo su integridad y disponibilidad.

Permitir el acceso a la herramienta a través de un entorno web

Permitir generar históricos de productos, clientes y proveedores

Generar alertas al inicio de la aplicación sobre los productos con stock bajo

### **7.2 Lo que no se va a lograr**

Generar históricos de acceso por usuarios.

Permitir el acceso y administración de la base de datos

## **8. LIMITACIONES DEL PROYECTO**

- El funcionamiento de la aplicación está sujeto a la correcta comunicación con el servidor donde se almacena tanto la aplicación como la base de datos.
- El acceso a los servidores y código fuente está disponible solamente al funcionario líder del proyecto por parte de la empresa.
- La modificación de cualquier parámetro o modulo, está sujeta a un nuevo desarrollo previo a un requerimiento.

## 9. CRONOGRAMA

El cronograma está definido en 6 fases, para un total de 98 días y se verá en el anexo J.

Id		Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
1			<b>Cronograma Proyecto Inventario</b>	<b>98 días</b>	<b>lun 06/07/15</b>	<b>mié 18/11/15</b>
2			<b>1. Fase de Inicio</b>	<b>5 días</b>	<b>lun 06/07/15</b>	<b>vie 10/07/15</b>
3			Definición del proyecto	1 día	lun 06/07/15	lun 06/07/15
4			Levantamiento de Información	3 días	lun 06/07/15	mié 08/07/15
5			Análisis de la información	2 días	jue 09/07/15	vie 10/07/15
6			<b>2. Fase de planeación</b>	<b>3 días</b>	<b>lun 13/07/15</b>	<b>mié 15/07/15</b>
7			Definición de roles	1 día	lun 13/07/15	lun 13/07/15
8			Definición de metodología	1 día	mar 14/07/15	mar 14/07/15
9			Asignación de tareas	1 día	mié 15/07/15	mié 15/07/15
10			<b>3. Fase de Diseño</b>	<b>10 días</b>	<b>jue 16/07/15</b>	<b>mié 29/07/15</b>
11			Diseño de Aplicación	5 días	jue 16/07/15	mié 22/07/15
12			Diseño de Bases de Datos	5 días	jue 23/07/15	mié 29/07/15
13			<b>4. Fase de Desarrollo</b>	<b>50 días</b>	<b>jue 30/07/15</b>	<b>mié 07/10/15</b>
14			Desarrollo y construcción de m	50 días	jue 30/07/15	mié 07/10/15
15			Desarrollo y construcción de B	50 días	jue 30/07/15	mié 07/10/15
16			<b>5. Fase de Pruebas con Cliente</b>	<b>5 días</b>	<b>jue 08/10/15</b>	<b>mié 14/10/15</b>
17			Pruebas de Aplicación	5 días	jue 08/10/15	mié 14/10/15
18			<b>6. Fase de Implementación, Esta</b>	<b>25 días</b>	<b>jue 15/10/15</b>	<b>mié 18/11/15</b>
19			Implementación y pruebas cor	10 días	jue 15/10/15	mié 28/10/15
20			Estabilización de la aplicación	5 días	jue 29/10/15	mié 04/11/15
21			Documentación	10 días	jue 05/11/15	mié 18/11/15

## **10. RECOMENDACIONES**

Programar capacitaciones al personal encargado del uso de la herramienta, para así mejorar los resultados de su uso y minimizar el riesgo de pérdida o mal manejo de la información

Definir el rol de cada usuario para así mismo realizar su creación en la herramienta.

Disponer de las herramientas necesarias como requerimientos mínimos de hardware para el uso ideal de la aplicación

Generar procesos de retroalimentación del uso y funcionamiento de la herramienta, para así obtener información valiosa para cambios y/o mejoras futuras

## CONCLUSIONES

Las experiencias obtenidas a través del desarrollo del proyecto nos permite concluir que:

- El uso de una herramienta inadecuada puede generar pérdidas de información, además de no permitir la administración adecuada de los datos, siendo esto último una falencia en la administración de un inventario.
- El desarrollo e implementación de una herramienta de software diseñada específicamente para atender una necesidad genera en las partes involucradas crecimiento y ampliación del conocimiento referente al área usuaria.
- Poner en práctica los procesos y metodologías de desarrollo de software permite tanto a los desarrolladores como los dueños del proyecto encontrar soluciones a problemas presentados en la ejecución del proceso a mejorar.
- El desarrollo y la implementación del proyecto nos permitió aplicar los conocimientos previamente adquiridos en las aulas además de generar cambios sustanciales, siendo estos, aportes a la sociedad, en la medida que genera crecimiento tanto al interior como exterior de la empresa.
- El Implementar nuevas tecnologías contribuye con la modernización de las empresas, haciéndolas más competitivas, mejorando sus economías y la de la industria en general.

## BIBLIOGRAFÍA

ESCUELA DE EDUCACIÓN SECUNDARIA TÉCNICA NRO. 2. “Normalización de Bases de Datos”. {En línea}. Fecha. {18 Junio de 2015}. Disponible en: (<http://www.eet2mdp.edu.ar/alumnos/MATERIAL/MATERIAL/info/infonorma.pdf>)

FUNDACIÓN WIKIPEDIA INC. “Base de datos”. {En línea}. Fecha. {25 de Agosto de 2015}. Disponible en: ([https://es.wikipedia.org/wiki/Base\\_de\\_datos](https://es.wikipedia.org/wiki/Base_de_datos))

FUNDACIÓN WIKIPEDIA INC. “SQL”. {En línea}. Fecha. {25 de Agosto de 2015}. Disponible en: (<https://es.wikipedia.org/wiki/SQL>)

FUNDACIÓN WIKIPEDIA INC. “12 reglas de Codd”. {En línea}. Fecha. {25 de Agosto de 2015}. Disponible en: ([https://es.wikipedia.org/wiki/12\\_reglas\\_de\\_Codd](https://es.wikipedia.org/wiki/12_reglas_de_Codd))

HISPALINUX. “El Lenguaje SQL”. {En línea}. Fecha. {25 de Agosto de 2015}. Disponible en: (<http://es.tldp.org/Postgresql-es/web/navegable/tutorial/sql-language.html>)

MEDIEVALSTIMES.WORDPRESS.COM. “12 reglas de Codd para bases de datos Relacionadas”. {En línea}. Fecha. {30 de Agosto de 2015}. Disponible en: (<https://medievalstrucos.wordpress.com/2013/07/18/12-reglas-de-codd-para-bases-de-datos-relacionadas/>)

APT SOFTWARE. “Las 12 Reglas De Codd Que Determinan La Fidelidad De Un Sistema Relacional Al Modelo Relacional”. {En línea}. Fecha. {22 Junio de 2015}. Disponible en: (<http://www.atpssoftware.net/Docs/12ReglasCodd.htm>)

CASAS FERNÁNDEZ, Bernardino. "Normalización de Bases de Datos y Técnicas de diseño". {En línea}. Fecha. {18 Junio de 2015}. Disponible en: (<http://www.cs.upc.edu/~bcasas/docencia/pfc/NormalitzacioBD.pdf>)

CENDEJAS VALDÉZ, José Luis "Modelos y metodologías para el desarrollo de software". {En línea}. Fecha. {12 Junio de 2015}. Disponible en: (<http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>)

## ANEXO A Ficha de caso de uso 1

<b>FICHA DE CASO DE USO INICIAR SESIÓN</b>	
<b>ID</b>	1
<b>NOMBRE</b>	Iniciar sesión
<b>DESCRIPCIÓN</b>	El usuario requiere iniciar sesión.
<b>FLUJO NORMAL</b>	
<b>ACTORES</b>	Usuario final de la aplicación
<b>PRECONDICIONES</b>	Poseer un usuario valido
<b>ACTIVACIÓN</b>	El usuario ingresa los datos
<b>DESCRIPCIÓN</b>	1- El usuario ingresa su nombre 2- El usuario ingresa su clave 3- Se confirman los datos ingresados
<b>POSTCONDICIONES</b>	Se procede a la pantalla principal del sistema dependiendo del rol que tenga el usuario.
<b>FLUJO ALTERNATIVO 1</b>	
<b>DESCRIPCIÓN</b>	3- Los datos ingresados son incorrectos
<b>POSTCONDICIONES</b>	Se informa el error con un mensaje. El formulario queda "limpio" para que se ingresen nuevos datos
<b>FLUJO ALTERNATIVO 2</b>	
<b>DESCRIPCIÓN</b>	3- El usuario no ingresa alguno de los campos
<b>POSTCONDICIONES</b>	Se informa con un mensaje de error el campo requerido que falta. Los datos previamente ingresados quedan en el formulario.

## ANEXO B Ficha de caso de uso 2

<b>FICHA DE CASO DE USO CERRAR SESIÓN</b>	
<b>ID</b>	2
<b>NOMBRE</b>	Cerrar sesión
<b>DESCRIPCIÓN</b>	El usuario desea abandonar el sistema
<b>FLUJO NORMAL</b>	
<b>ACTORES</b>	Usuario final de la aplicación
<b>PRECONDICIONES</b>	El usuario debe estar logueado al sistema
<b>ACTIVACIÓN</b>	El usuario abandona la aplicación
<b>DESCRIPCIÓN</b>	1- El usuario pulsa el botón de cerrar sesión de estado de login
<b>POSTCONDICIONES</b>	El usuario no pueden seguir haciendo uso del sistema sin antes volver a iniciar sesión.
<b>FLUJO ALTERNATIVO 1</b>	
<b>DESCRIPCIÓN</b>	
<b>FLUJO ALTERNATIVO 2</b>	
<b>DESCRIPCIÓN</b>	-

### ANEXO C Ficha de caso de uso 3

<b>FICHA DE CASO DE USO</b>	
<b>ID</b>	3
<b>NOMBRE</b>	Acerca de
<b>DESCRIPCIÓN</b>	El usuario ve la información de contacto del fabricante del sistema
<b>FLUJO NORMAL</b>	
<b>ACTORES</b>	Usuario final de la aplicación
<b>PRECONDICIONES</b>	El usuario debe estar logueado al sistema
<b>ACTIVACIÓN</b>	El usuario solicita a información acerca del fabricante
<b>DESCRIPCIÓN</b>	1- El usuario pulsa el botón "Acerca de"
<b>POSTCONDICIONES</b>	Se muestra una pantalla con la información solicitada
<b>FLUJO ALTERNATIVO 1</b>	
<b>DESCRIPCIÓN</b>	-
<b>FLUJO ALTERNATIVO 2</b>	
<b>DESCRIPCIÓN</b>	-

## ANEXO D Ficha de caso de uso 4

<b>FICHA DE CASO DE USO INGRESAR NUEVO ARTICULO</b>	
<b>ID</b>	4
<b>NOMBRE</b>	Nuevo articulo
<b>DESCRIPCIÓN</b>	Se carga un nuevo artículo a la BD
<b>FLUJO NORMAL</b>	
<b>ACTORES</b>	Usuario Administrador de la aplicación
<b>PRECONDICIONES</b>	El usuario debe poseer los privilegios necesarios El cliente no debe existir previamente
<b>ACTIVACIÓN</b>	El usuario ingresa un nuevo articulo
<b>DESCRIPCIÓN</b>	1- El usuario ingresa a la pantalla que tiene dicho fin 2- Se ingresan los datos 3- Se confirman los datos
<b>POSTCONDICIONES</b>	Hay un nuevo cliente en la BD
<b>FLUJO ALTERNATIVO 1</b>	
<b>DESCRIPCIÓN</b>	3- Se cancela el ingreso
<b>POSTCONDICIONES</b>	En la BD hay la misma cantidad de clientes
<b>FLUJO ALTERNATIVO 2</b>	
<b>DESCRIPCIÓN</b>	-

## ANEXO E Ficha de caso de uso 5

<b>FICHA DE CASO DE USO 5 ACTUALIZAR ARTICULO</b>	
<b>ID</b>	5
<b>NOMBRE</b>	Actualizar articulo
<b>DESCRIPCIÓN</b>	Se modifica información registrada en la BD
<b>FLUJO NORMAL</b>	
<b>ACTORES</b>	Usuario final de la aplicación
<b>PRECONDICIONES</b>	El usuario debe poseer los privilegios necesarios El articulo debe existir
<b>ACTIVACIÓN</b>	El usuario actualiza la información registrada en la BD
<b>DESCRIPCIÓN</b>	1- El usuario ingresa a la pantalla con dicho fin 2- Realiza la búsqueda del articulo 3- Se actualiza el estado del articulo
<b>POSTCONDICIONES</b>	
<b>FLUJO ALTERNATIVO 1</b>	
<b>DESCRIPCIÓN</b>	3- Se cancela la selección
<b>POSTCONDICIONES</b>	No se realizan modificaciones en la BD
<b>FLUJO ALTERNATIVO 2</b>	
<b>DESCRIPCIÓN</b>	-

## ANEXO F Ficha de caso de uso 6

<b>FICHA DE CASO DE USO MODIFICAR CLIENTE</b>	
<b>ID</b>	6
<b>NOMBRE</b>	Modificar cliente
<b>DESCRIPCIÓN</b>	Se modifican los datos de un cliente de la BD
<b>FLUJO NORMAL</b>	
<b>ACTORES</b>	Usuario final de la aplicación
<b>PRECONDICIONES</b>	El usuario debe poseer los privilegios necesarios El cliente debe existir
<b>ACTIVACIÓN</b>	El usuario modifica los datos de un cliente
<b>DESCRIPCIÓN</b>	1- El usuario ingresa a la pantalla con dicho fin 2- El usuario modifica los datos 3- Se confirman los cambios
<b>POSTCONDICIONES</b>	Los datos del cliente modificado son los que recién ingreso el usuario
<b>FLUJO ALTERNATIVO 1</b>	
<b>DESCRIPCIÓN</b>	3- Se cancela la selección
<b>POSTCONDICIONES</b>	Los datos del cliente siguen siendo los mismos que antes
<b>FLUJO ALTERNATIVO 2</b>	
<b>DESCRIPCIÓN</b>	-

## ANEXO G Ficha de caso de uso 7

<b>FICHA DE CASO DE USO BUSCAR ARTICULO</b>	
<b>ID</b>	7
<b>NOMBRE</b>	Buscar Articulo
<b>DESCRIPCIÓN</b>	Carga información referente a un parámetro de búsqueda.
<b>FLUJO NORMAL</b>	
<b>ACTORES</b>	Usuario final de la aplicación
<b>PRECONDICIONES</b>	El usuario debe poseer los privilegios necesarios El pedido debe existir
<b>ACTIVACIÓN</b>	El usuario buscar un artículo existente en la base de datos.
<b>DESCRIPCIÓN</b>	1- El usuario ingresa a la pantalla con dicho fin 2- Realiza la búsqueda dependiendo del parámetro. 3- Se confirma selección
<b>POSTCONDICIONES</b>	No se modifican registros en la base de datos.
<b>FLUJO ALTERNATIVO 1</b>	
<b>DESCRIPCIÓN</b>	3- Se cancela la selección
<b>POSTCONDICIONES</b>	No se modifican registros en la base de datos.
<b>FLUJO ALTERNATIVO 2</b>	
<b>DESCRIPCIÓN</b>	-

## ANEXO H Diccionario de datos

Tabla	Descripción
ARTICULO	Registra toda la información referente a los artículos, se relaciona con las tablas, Categoría, Ubicación, Proveedor, Compra y Venta.
CATEGORIA	Registra toda la información referente a las categorías de los Artículos, se relaciona con la tabla Artículo.
CLIENTE	Registra toda la información referente a los clientes, se relaciona con las tablas compra y venta.
COMPRA	Registra toda la información referente a las compras de los artículos, contribuye con el control de stock. Se relaciona con las tablas Artículo y Proveedor.
FUNCIONARIO	Registra toda la información referente a los funcionarios que trabajan en la compañía, se relaciona con la tabla Venta.
PROVEEDOR	Registra toda la información referente a los proveedores, se relaciona con las tablas Artículo y Compra.
UBICACION	Registra toda la información referente a la ubicación de los diferentes artículos. Se relaciona con la tabla Artículo.
VENTA	Registra toda la información referente a las ventas de los artículos, contribuye con el control de stock. Se relaciona con las tablas Artículo y Cliente.

Tabla	Columna	Tipo de Dato	Tamaño	Valor Nullo	Llave Primaria	Llave Foranea	Descripción
ARTICULO	Codigoarticulo	VarChar (20)	20	N	Y	N	Es el identificador de cada artículo, no se puede duplicar.
ARTICULO	Nombreaticulo	VarChar (50)	50	N	N	N	Es el nombre que recibe cada artículo.
ARTICULO	Marca	VarChar (50)	50	Y	N	N	La marca que recibe cada artículo, por su fabricante.
ARTICULO	Modelo	VarChar (50)	50	Y	N	N	Es el modelo del artículo.
ARTICULO	Color	VarChar (50)	50	Y	N	N	Indica si los artículos tiene diferentes tonalidades.
ARTICULO	Unidadmedida	VarChar (10)	10	Y	N	N	Se ingresa siempre y cuando los artículos se miden por magnitud física.
ARTICULO	Stock	Int	4	N	N	N	Indica la cantidad de artículos a disponibles.
ARTICULO	Codigocategoria	Int	4	N	Y	Y	Se relaciona con la tabla categoría, hace referencia a la categoría a la cual pertenece el artículo.
ARTICULO	Codigoubicacion	Int	4	N	Y	Y	Se relaciona con la tabla ubicación, hace referencia a la ubicación que tendrá el artículo.
ARTICULO	Codigoproveedor	Numeric	9	N	Y	Y	Se relaciona con la tabla proveedor, hace referencia al proveedor que distribuye el artículo.
ARTICULO	Estado	NVarChar (15)	15	Y	N	N	Indica si el stock esta o no esta disponible para venta.

Tabla	Columna	Tipo de Dato	Tamaño	Valor Nulo	Llave Primaria	Llave Foranea
CATEGORIA	Codigocategoria	Int	4	N	Y	N
CATEGORIA	Nombrecategoria	VarChar (50)	50	N	N	N
CATEGORIA	Estado	NVarChar (15)	15	Y	N	N

Tabla	Columna	Tipo de Dato	Tamaño	Valor Nulo	Llave Primaria	Llave Foranea	Descripción
CLIENTE	Codigocliente	Numeric	9	N	Y	N	Es el identificador de cada cliente, no se puede duplicar.
CLIENTE	Nombrecliente	VarChar (50)	50	N	N	N	Es el nombre que recibe cada cliente.
CLIENTE	Personacontactocliente	VarChar (50)	50	Y	N	N	Se refiere al nombre de la persona de contacto.
CLIENTE	Telefonocliente	VarChar (10)	10	Y	N	N	Es el telefono del cliente.
CLIENTE	Direccioncliente	VarChar (50)	50	Y	N	N	Es la direccion del cliente.
CLIENTE	Emailcliente	VarChar (50)	50	Y	N	N	Es el email del cliente.
CLIENTE	Cuidad	VarChar (50)	50	Y	N	N	Es la ciudad del cliente.
CLIENTE	Estado	NVarChar (15)	15	Y	N	N	Indica si el cliente esta o no esta disponible para ventas.

Tabla	Columna	Tipo de Dato	Tamaño	Valor Nulo	Llave Primaria	Llave Foranea	Descripción
COMPRA	Codigocompra	Int	4	N	Y	N	Es el identificador de cada Compra, no se puede duplicar.
COMPRA	Cantidad	Int	4	Y	N	N	Indica la cantidad comprada de articulos.
COMPRA	Valorunitario	Int	4	Y	N	N	Indica el valor unitario de cada articulo.
COMPRA	Valortotal	Int	4	Y	N	N	Indica el valor total de la compra.
COMPRA	fechacompra	DateTime	8	Y	N	N	Indica la fecha de la compra.
COMPRA	Codigoarticulo	VarChar (20)	20	N	Y	Y	Se relaciona con la tabla articulo, indica el codigo del articulo comprado.
COMPRA	Codigocategoria	Int	4	N	Y	Y	Se relaciona con la tabla categoria, indica la categoria del articulo comprado.
COMPRA	Codigubicacion	Int	4	N	Y	Y	Se relaciona con la tabla ubicacion, indica la ubicacion a la cual se cargara el articulo.
COMPRA	Codigoproveedor	Numeric	9	N	Y	Y	Se relaciona con la tabla proveedor, indica el proveedor que vende el articulo.

Tabla	Columna	Tipo de Dato	Tamaño	Valor Nulo	Llave Primaria	Llave Foranea	Descripción
FUNCIONARIO	Codigofuncionario	Int	4	N	Y	N	Es el identificador de cada funcionario, no se puede duplicar.
FUNCIONARIO	Nombre	VarChar (50)	50	N	N	N	Es el nombre del funcionario.
FUNCIONARIO	Apellido	VarChar (50)	50	Y	N	N	Es el Apellido del funcionario.
FUNCIONARIO	Cargo	VarChar (50)	50	Y	N	N	Es el cargo que desempeña el funcionario.
FUNCIONARIO	Usuario	VarChar (10)	10	N	N	N	Es el usuario con el que se loguea en la aplicación.
FUNCIONARIO	Password	VarChar (10)	10	N	N	N	Es la contraseña asignada para logueo en la aplicación.
FUNCIONARIO	Rol	Int	4	N	N	N	Es el rol que desempeñara en la aplicación.
FUNCIONARIO	Estado	NVarChar (15)	15	Y	N	N	Indica si el funcionario esta o no esta disponible para utilizar la aplicacion.

Tabla	Columna	Tipo de Dato	Tamaño	Valor Nulo	Llave Primaria	Llave Foranea	Descripción
PROVEEDOR	Codigoproveedor	Numeric	9	N	Y	N	Es el identificador de cada proveedor, no se puede duplicar.
PROVEEDOR	Nombreprveedor	VarChar (50)	50	N	N	N	Es el nombre que recibe cada proveedor.
PROVEEDOR	Personacontactoprov	VarChar (50)	50	Y	N	N	Se refiere al nombre de la persona de contacto.
PROVEEDOR	Telefonoproveedor	VarChar (15)	15	Y	N	N	Es el telefono del proveedor.
PROVEEDOR	Direccionproveedor	VarChar (50)	50	Y	N	N	Es la direccion del proveedor.
PROVEEDOR	Emailproveedor	VarChar (50)	50	Y	N	N	Es el email del proveedor.
PROVEEDOR	Ciudadproveedor	VarChar (50)	50	Y	N	N	Es la ciudad del proveedor.
PROVEEDOR	Estado	NVarChar (15)	15	Y	N	N	Indica si el proveedor esta o no esta disponible para ventas.

Tabla	Columna	Tipo de Dato	Tamaño	Valor Nulo	Llave Primaria	Llave Foranea	Descripción
UBICACION	Codigubicacion	Int	4	N	Y	N	Es el identificador de cada ubicacion, no se puede duplicar.
UBICACION	Nombrebubicacion	VarChar (50)	50	N	N	N	Es el nombre que recibe cada ubicacion.
UBICACION	Telefonoubicacion	VarChar (10)	10	Y	N	N	Es el telefono del sitio.
UBICACION	Direccionubicacion	VarChar (50)	50	Y	N	N	Es la direccion del sitio.
UBICACION	Detallesubicacion	VarChar (50)	50	Y	N	N	Si se requieren especificar detalles adicionales.
UBICACION	Estado	NVarChar (15)	15	Y	N	N	Indica si la ubicacion esta o no esta disponible para asignar a un articulo.

Tabla	Columna	Tipo de Dato	Tamaño	Valor Nullo	Llave Primaria	Llave Foranea	Descripción
VENTA	Codigoventa	Int	4	N	Y	N	Es el identificador de cada Venta, no se puede duplicar.
VENTA	Cantidadventa	Int	4	Y	N	N	Indica la cantidad vendida de articulos,
VENTA	Valorunitarioventa	Int	4	Y	N	N	Indica el valor unitario de cada articulo.
VENTA	Valortotalventa	Int	4	Y	N	N	Indica el valor total de la venta.
VENTA	Fechaventa	DateTime	8	Y	N	N	Indica la fecha de la venta.
VENTA	Codigoarticulo	VarChar (20)	20	N	Y	Y	Se relaciona con la tabla articulo, indica el codigo del articulo vendido.
VENTA	Codigocategoria	Int	4	N	Y	Y	Se relaciona con la tabla categoria, indica la categoria del articulo comprado.
VENTA	Codigoubicacion	Int	4	N	Y	Y	Se relaciona con la tabla ubicacion, indica la ubicacion a la cual se descargara el articulo.
VENTA	Codigoproveedor	Numeric	9	N	Y	Y	Se realciona con la tabla proveedor, indica el proveedor que vende el articulo.
VENTA	Codigofuncionario	Int	4	N	Y	Y	Se relaciona con la tabla funcionari, indica el usuario que realizo la venta.
VENTA	Codigoclient	Numeric	9	N	Y	Y	Se relaciona con la tabla cliente, indica el usuario que realizo la compra.

## ANEXO I Modelo Entidad Relación

