



**INSTITUTO SUPERIOR TECNOLÒGICO BOLIVARIANO DE  
TECNOLOGÌA**

**UNIDAD ACADÉMICA DE EDUCACIÓN COMERCIAL, ADMINISTRATIVA  
Y CIENCIAS**

**Proyecto de Investigación previo a la obtención del título de:**

**TECNÓLOGO EN ANALISIS DE SISTEMAS**

**TEMA:**

**DISEÑO DE UN SOFTWARE DE CONTROL DE INVENTARIOS DE  
COMPRA Y VENTA DE LICORES EN LA SALSOTECA BARRETO DE LA  
CIUDAD DE QUEVEDO EN EL 2019.**

**Autor: Briones Gonzabay Orlys Bryan**

**Tutor: Ing. Harold Ordaz Valdes**

**Guayaquil, Ecuador**

**2019**



## **DEDICATORIA**

En primer lugar agradezco a DIOS por darme la sabiduría y la fortaleza de permitirme haber llegado hasta este momento de mi vida, todo se lo debo a él, me puso duras pruebas que las fui superando una a una esto me ha permitido ir creciendo profesionalmente.

Seguidamente agradezco a mis padres, Orlys Briones y Dayci Gonzabay por su apoyo incondicional en cada paso que he dado en mis estudios y las decisiones que he tomado con aprobación de ellos y sus consejos para cada momento de mi vida en mis estudios.

**Briones Gonzabay Orlys Bryan**



## **AGRADECIMIENTO**

Agradezco a DIOS por guiarme en este camino largo lleno de obstáculos que los supere uno a uno gracias a su infinita bondad, y permitirme llegar hasta este momento de mi vida, dándome las fuerzas necesarias para cada dificultad que fui superando en mis estudios, sin su bendición y su compañía no hubiera llegado a este momento tan especial para mí.

**Briones Gonzabay Orlys Bryan**

**INSTITUTO SUPERIOR TECNOLÒGICO BOLIVARIANO DE TECNOLOGÌA**

**UNIDAD ACADÉMICA DE EDUCACIÓN COMERCIAL, ADMINISTRATIVA Y CIENCIAS**

**PROYECTO DE GRADO PREVIO A LA OBTENCION DEL TITULO DE: TECNÓLOGO EN ANÁLISIS DE SISTEMAS**

**TEMA:**

Diseño de un software de control de inventarios de compra y venta de licores en la Salsoteca Barreto de la ciudad de Quevedo en el 2019.

Autor: Orlys Bryan Briones Gonzabay

Tutor: Ing. Harold Ordaz Valdes

**RESUMEN**

El presente proyecto de investigación se basa en la importancia y la necesidad que tiene un software de control de inventarios, especialmente a los pequeños negocios o locales que empiezan en el mundo del negocio, donde se puede manifestar que son anexos entre los productos y las ventas, los sistemas de inventarios son de vital importancia para las personas que poseen negocios, y se encuentra en el mundo del negocio de la oferta y demanda.

Al tener un manejo adecuado garantiza un mejor desenvolvimiento en el control de ingreso y egreso de los licores, tantos nacionales como internacionales que se encuentran a la venta en la Salsoteca Barreto.

Dicha iniciativa del sistema de inventarios será de mucha ayuda, ya que con los datos investigados previamente, se ha podido recopilar falencias que servirán en el presente proyecto de diseñar un software de control de inventarios para poder tener un mejor manejo de la información en los productos de la Salsoteca Barreto, la cual se dedica a la compra y venta de distintas marcas de licores.

<b>PALABRAS CLAVES</b>		
Software Informático	Compra y Venta	Control de Inventarios

**INSTITUTO SUPERIOR TECNOLÒGICO BOLIVARIANO DE TECNOLOGÌA**

**UNIDAD ACADÉMICA DE EDUCACIÓN COMERCIAL, ADMINISTRATIVA Y CIENCIAS**

**PROYECTO DE GRADO PREVIO A LA OBTENCION DEL TITULO DE: TECNÓLOGO EN ANÁLISIS DE SISTEMAS**

**TEMA:**

Diseño de un software de control de inventarios de compra y venta de licores en la Salsoteca Barreto de la ciudad de Quevedo en el 2019.

**Autor:** Orlys Bryan Briones Gonzabay

**Tutor:** Ing. Harold Ordaz Valdes

**ABSTRACT**

This research project is based on the importance and need for inventory control software, especially to small businesses or stores that start in the business world, where it can be stated that they are annexes between products and sales, inventory systems are of vital importance for people who own businesses, and are in the world of the supply and demand business.

Having an adequate management guarantees a better development in the control of entry and exit of liquors, both national and international that are for sale in the Salsoteca Barreto.

This initiative of the inventory system will be very helpful, since with the data previously investigated, it has been possible to gather shortcomings that will serve in the present project to design an inventory control software to be able to have a better management of the information in the products of the Salsoteca Barreto, which is dedicated to the purchase and sale of different brands of spirits.

**KEYWORDS**

<b>KEYWORDS</b>		
Computer Software	Buy and sell	Inventory control



## ÍNDICE GENERAL

### Contenido

DEDICATORIA	ii
AGRADECIMIENTO	iii
CERTIFICACIÓN DE ACEPTACIÓN DEL TUTOR	iv
CLÁUSULA DE AUTORIZACIÓN PARA LA PUBLICACIÓN DE TRABAJOS DE TITULACIÓN	v
CERTIFICACIÓN DE ACEPTACIÓN DEL CEGESCIT	viii
RESUMEN	ix
ABSTRACT	x
ÍNDICE GENERAL	xi
CAPITULO I	1
EL PROBLEMA	1
PLANTEAMIENTO DEL PROBLEMA	1
1.1 UBICACIÓN DEL PROBLEMA EN UN CONTEXTO	1
1.2 SITUACIÓN CONFLICTO	2
1.3 FORMULACIÓN DEL PROBLEMA	3
1.4 DELIMITACIÓN DEL PROBLEMA	3
1.5 VARIABLES DE LA INVESTIGACIÓN	3
1.6 EVALUACIÓN DEL PROBLEMA	4
1.7 OBJETIVOS DE LA INVESTIGACIÓN	5
1.7.1 Objetivo General	5
1.7.2 Objetivos Específicos	5
1.8 JUSTIFICACIÓN DE LA INVESTIGACIÓN	6
CAPITULO II	7

MARCO TEORICO	7
2.1 FUNDAMENTACIÓN TEÓRICA	7
2.1.1 Antecedentes históricos .....	7
2.1.2 ANTECEDENTES REFERENCIALES.....	10
2.2 Aspectos teóricos fundamentales	13
2.2.1 Paradigmas de programación para sistemas de cómputo.....	13
2.2.2 Lenguajes y Tecnologías de Escritorio.....	16
2.2.2.1 Software de Aplicación para Escritorio.....	16
2.2.2.2 Lenguajes de Programación.....	17
2.2.2.3 Arquitectura de Software .....	27
2.2.2.4 Base de Datos .....	36
2.2.3 Frameworks de Escritorio (Marco de trabajo).....	43
2.2.4 Metodologías de Desarrollo de Software	49
2.2.4.1 Modelos y enfoques de desarrollo.....	49
2.2.4.2 Tradicionales .....	53
2.2.4.3 Ágiles.....	54
2.3 Fundamentación Legal	55
2.4 Variables de la Investigación (Conceptualización)	56
2.4.1 Variable Dependiente .....	56
2.4.2 Variable Independiente.....	57
2.5 Glosario de Términos	67
CAPITULO III	69
METODOLOGIA	69
3.1 PRESENTACIÓN DE LA EMPRESA .....	69
3.2 Diseño de la investigación.....	70



3.3 Tipos de investigación .....	71
3.3 La Encuesta.....	73
3.4 Solución Propuesta .....	74
3.5 Estudio de Factibilidad .....	74
3.6 Características Relevantes.....	75
3.7 Beneficios .....	77
<b>CAPITULO IV</b>	<b>78</b>
<b>DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA</b>	<b>78</b>
4.1 Descripción Del Sistema Propuesto .....	78
4.1.1 Alcances Del Proyecto .....	78
4.2 Requerimientos Del Software .....	79
4.2.1 Requerimientos Funcionales Del Sistema .....	79
4.2.2 Requerimientos No Funcionales.....	80
4.3 Diagrama de Caso de Uso General.....	82
4.4 Diagrama de Actividades de Caso de Uso .....	82
4.5 Mapa de navegación .....	95
4.6 Diseño de prototipado de la propuesta.....	96
4.6.1 Prototipado Pantallas De Usuario.....	96
4.6.2 Prototipado Pantallas De Administrador.....	100
4.7 Diagrama Entidad-Relación de la Base de Datos.....	102

4.8	Presupuesto.....	102
4.8.1	Requerimiento de software.....	102
4.8.2	Costo de Software .....	102
4.8.3	Costo del Sistema .....	103
4.8.4	Requerimiento del Personal .....	103
	CONCLUSIONES	104
	RECOMENDACIONES	105
	BIBLIOGRAFIA	106

## INDICE DE FIGURAS

<b>Títulos:</b>	<b>Páginas:</b>
Figura 1: Formulario de inicio de sesión en vista de diseño .	11
Figura 2: Ingreso al Sistema ..	11
Figura 3: Formulario de inicio de sesión en vista de diseño .	12
Figura 4: Paradigmas de la Programación ..	13
Figura 5: Paradigma Funcional .	14
Figura 6: Paradigma Lógico ...	14
Figura 7: Paradigma Imperativo	15
Figura 8: Paradigma Orientado a Objetos ...	15

Figura 9: Software de Aplicación para Escritorio	16
Figura 10: Lenguajes De Programación	17
Figura 11: Bajo Nivel	17
Figura 12: De Maquina	18
Figura 13: Ensamblador	18
Figura 14: De Alto Nivel	19
Figura 15: Prolog	20
Figura 16: Scala	21
Figura 17: Lisp	21
Figura 18: Scheme	22
Figura 19: Imperativos y Orientado a Objetos	23
Figura 20: C++	23
Figura 21: Java	24
Figura 22: C#	25
Figura 23: Visual .Net	25
Figura 24: Python	26
Figura 25: Ruby	27
Figura 26: Arquitectura de Software	27
Figura 27: Descomposición Modular	28
Figura 28: Cliente-Servidor	29
Figura 29: Arquitectura de Tres Niveles	30
Figura 30: Capa Presentación	31
Figura 31: Capa de Negocio	32
Figura 32: Capa de Acceso de Datos	32

Figura 33: Modelo Vista Controlador	33
Figura 34: Orientada a Servicios	34
Figura 35: Arquitectura de Micro servicios	34
Figura 36: Dirigida por Eventos	35
Figura 37: Base de Datos	36
Figura 38: Sistema de Administración de Base de Datos	36
Figura 39: SQL	37
Figura 40: SQL Server (Microsoft)	38
Figura 41: Oracle	39
Figura 42: MySQL	39
Figura 43: PostgreSQL	40
Figura 44: NoSQL	41
Figura 45: Redis	42
Figura 46: MongoDB	42
Figura 47: CouchDB	43
Figura 48: Framework Spring	44
Figura 49: Django	44
Figura 50: Prism	44
Figura 51: MVVM Light Toolkit	45
Figura 52: MVVM CROSS	45
Figura 53: ReactiveUI	46
Figura 54: .NET	46
Figura 55: Ruby On Rails	47
Figura 56: Sinatra	48

Figura 57: Hobbit	48
Figura 58: Modelo en Cascada	50
Figura 59: Modelo Prototipo	50
Figura 60: Modelo Incremental	51
Figura 61: Modelo en Espiral	51
Figura 62: Rapid Application Development	52
Figura 63: RUP	53
Figura 64: XP	54
Figura 65: SCRUM	55
Figura 66: Inventario físico	59
Figura 67: Organigrama de la empresa	69
Figura 68: Diagrama de caso de uso general	83
Figura 69: Pantalla principal	83
Figura 70: Menú de opciones	84
Figura 71: Licores ingresados en bodega	85
Figura 72: Características de los licores	86
Figura 73: Productos de venta de proveedores	87
Figura 74: Venta de productos diarios	88
Figura 75: Autenticación de usuario	89
Figura 76: Solicitar nuevo usuario	90
Figura 77: Gestionar Usuario	91
Figura 78: Gestionar menú de opciones	92
Figura 79: Gestionar compra de licores	93
Figura 80: Gestionar venta de licores	94

Figura 81: Pantalla inicio de sesión 96

Figura 82: Pantalla de registro nuevo usuario 97

Figura 83: Menú de opciones 97

Figura 84: Productos ingresados a bodega 98

Figura 85: Características de los productos 98

Figura 86: Productos de venta proveedores 99

Figura 87: Venta de productos diarios 100

Figura 88: Gestionar menú de opciones 100

Figura 89: Gestionar compra de productos 101

Figura 90: Gestionar venta de productos 101

Figura 91: Diagrama Entidad-Relación de la base de datos 102

## **CAPITULO I**

### **EL PROBLEMA**

#### **PLANTEAMIENTO DEL PROBLEMA**

##### **1.1 UBICACIÓN DEL PROBLEMA EN UN CONTEXTO**

En la actualidad, el mundo de la tecnología ha dado grandes avances más que en el siglo anterior, es una época moderna de cambios donde la tecnología busca facilitar o mejorar el estilo de vida de las personas, dejando de lado la ambigüedad de los procesos manuales que se realizaban, para dar paso a una era de procesos mejorados o automatizados en el vivir diario, es así como han ido cambiando todas las empresas a nivel mundial.

En su mayoría las grandes empresas, optaron por mejorar sus procesos que los venían realizando de manera manual, a una más factible como la automatización de los procesos a seguir según los requerimientos y necesidades que tenga cada empresa. Llevar un control de inventarios de los productos de una empresa ha sido llevado por mucho tiempo de manera manual, y gracias a la tecnología se pueden automatizar y guardar su información en los softwares desarrollados.

Las personas llevaban sus registros en forma física, pero no automatizada, no llevaban un control exacto de productos de ingresos como de egresos del negocio que realizan, y así con la automatización se puede saber cuánto ha tenido almacenado los licores en un local, centro comercial, o cualquier tipo de negocio sea grande o mediano en cualquier parte del mundo, en muchas ocasiones se pueden perder los registros de la bodega de una empresa si no están automatizados.

Los sistemas de inventario manuales pueden ser muy intensivos en el trabajo. Ellos requieren un seguimiento continuo para asegurarse que cada transacción se contabilice y que los productos se mantengan en los niveles de inventario adecuados. También es más difícil compartir información de inventario en toda la empresa, ya que la falta de informatización hace que el acceso al inventario sea un proceso más engorroso. (Joseph)

En Ecuador son muchos los negocios que usan control de inventarios manuales, bien sea en documentos digitales Excel, o en hojas de cuadernos o carpetas, no es recomendable llevar un control así, se debe automatizar para que la información no pueda llegar a perderse o ser alterada.

## **1.2 SITUACIÓN CONFLICTO**

La Salsoteca Barreto es un centro de diversión nocturno ubicado en la provincia de Los Ríos, Cantón Quevedo, parroquia San Camilo en la Avenida El Ejercito y Panamá, inaugurada en el mes de Junio del año 2019. La Salsoteca se caracteriza por la buena acogida que da a los clientes que acuden a pasar un momento de diversión en compañía de amigos y familiares. El local tiene un entorno agradable y climatizado.

En la Salsoteca Barreto, su principal problemática es la falta de un control de inventarios para los licores almacenados en la bodega, o la falta de conocimiento de cómo realizar dicho proceso, el mismo que reflejara los movimientos durante un periodo mensual, ya que con estos procesos que vienen realizando de forma manual puede generar errores humanos en el registro de los productos que llegan al establecimiento.

Actualmente la Salsoteca Barreto lleva todos los registros de los licores que compran a sus proveedores para su posterior comercialización en el centro nocturno de manera manual, en cuadernos o libretas, o usando programas

como Word o Excel, específicamente la salsoteca no cuenta con un control de inventarios para registrar las actividades que se realizan en la bodega.

La Salsoteca dentro de su problemática actualmente enfrenta los problemas de control de inventarios de los licores existentes en la bodega, de los cuales se detallan:

- Control de inventarios que le permita a la Salsoteca tener la información de los licores que almacena en la bodega, sean por marca de licor, tipo, tamaño, precio, o año del producto.
- Manejo adecuado del control de los licores que van ingresando a la bodega, realizar una sumatoria exacta y, así mismo, para retirar los licores de la bodega para llevarlos a la venta al público.

Entrega oportuna de información a los proveedores o a las autoridades cuando así lo requiera.

### **1.3 FORMULACIÓN DEL PROBLEMA**

¿Cómo influye la falta de un software para la gestión del control de inventarios de licores de la salsoteca Barreto, en la contabilidad de los productos existentes para la compra y venta?

### **1.4 DELIMITACIÓN DEL PROBLEMA**

- **Campo.-** Desarrollo de Software.
- **Área.-** Ingeniería de Software.
- **Aspecto.-** Fase de Diseño del software.
- **Periodo.-** 2019.

## 1.5 VARIABLES DE LA INVESTIGACIÓN

- **Variables Dependiente.-** Contabilidad de los productos existentes para la compra y venta.
- **Variables Independientes.-** Gestión del control de inventarios.

## 1.6 EVALUACIÓN DEL PROBLEMA

### **Delimitado**

El presente proyecto de grado se lo realizará en la provincia de Los Ríos, cantón Quevedo, parroquia San Camilo, barrio Promejoras, con la finalidad que la Salsoteca Barreto, tenga un software de control de inventarios de los licores que se almacenan en la bodega del local.

### **Claro**

Es importante aclarar que la Salsoteca Barreto no existe un software de inventarios que controle los licores que ingresan o salen del área de bodega.

### **Evidente**

Es evidente debido a la poca organización que tiene la Salsoteca Barreto en llevar los controles de inventarios de los licores que llegan a la bodega del local ya que los llevan de manera manual.

### **Relevante**

Es relevante porque este proyecto de investigación tiene como propósito beneficiar a la Salsoteca Barreto proponiendo el diseño de un software de control de inventarios para los licores de compra y venta que se almacenan en la bodega.

## **Factible**

Este diseño de software de control de inventarios permitirá que la Salsoteca Barreto, tenga un mejor control en los licores que se almacenan en bodega, dicho proceso lo lleva en la actualidad de manera manual.

## **Concreto**

Es concreto porque servirá para llevar de manera automatizada toda la información respecto a los licores que se almacenan en la bodega, ya que actualmente se realiza manualmente.

## **1.7 OBJETIVOS DE LA INVESTIGACIÓN**

### **1.7.1 OBJETIVO GENERAL**

- Diseñar un software para la gestión del control de inventarios de licores de la salsoteca Barreto, para la contabilidad de los productos existentes a la venta y los que sean necesarios para la compra.

### **1.7.2 OBJETIVOS ESPECÍFICOS**

- Identificar la información científica relacionada con softwares disponibles para el control de inventarios en tiendas de venta de licores y discotecas.
- Diagnosticar el estado actual del registro e inventario de compra y venta de licores de la SALSOTECA BARRETO.
- Proponer el diseño del software para el control de inventarios de compra y venta de los licores de la Salsoteca Barreto.

## **1.8 JUSTIFICACIÓN DE LA INVESTIGACIÓN**

Este proyecto de investigación es muy conveniente porque sirve para llevar un control en cuanto a los licores que ingresan a la bodega cuando compran a los proveedores y los que se salen o se venden a los clientes en la salsoteca, y es muy útil también para cambiar el antiguo control que se llevaba de manera manual escrita en cuadernos o libros que podrían extraviarse, y proceder a llevar el control de forma automatizada.

Los controles que se pueden llevar en la bodega de forma automatizada son los datos de los licores, en cuanto a marca del producto, precios, cantidades, tamaño, año de fabricación, se puede hacer una clasificación por cada uno de ellos. Así se está llevando una contabilidad más detallada de los licores, sin que esta información pueda perderse ya que estará almacenada en una base de datos, y ya no en cuadernos que por errores humanos pueden llegar a perderse.

Tiene trascendencia en la sociedad si se cumplen las necesidades y la propuesta es conveniente para la población en general o para el grupo de personas que maneje de manera manual el control de inventarios en las bodegas de todas las Salsotecas del país, dejando atrás este antiguo y dando paso a la innovación de llevar la contabilidad automatizada de todos los licores que estén en venta al público en general.

Se busca crear un control de inventarios automatizado que sea de fácil manipulación para los propietarios de los locales que guarden en la bodega sus productos y los expendan al público, y así sus negocios estén siempre encaminados a tener una mejor organización de sus productos en el área de almacenaje. Son muchos los problemas que se buscan resolver en distintos tipos de negocios comerciales, medianos y pequeños que aún no han automatizado sus procesos de inventarios de productos o mercaderías.

## **CAPITULO II**

### **MARCO TEORICO**

#### **2.1 FUNDAMENTACIÓN TEÓRICA**

##### **2.1.1 ANTECEDENTES HISTÓRICOS**

Según (Capielo & Pirela, 2015) `La historia de los inventarios, comienza desde la antigüedad, cuando los pueblos, debido a las épocas de escases, deciden almacenar grandes cantidades de alimentos, para hacer frente a ellas; así se idea un mecanismo de control para su reparto. Desde tiempos inmemorables, los egipcios y demás pueblos de la antigüedad, acostumbraban almacenar grandes cantidades de alimentos para ser utilizados en los tiempos de sequía o de calamidades. Es así como surge o nace el problema de los inventarios, como una forma de hacer frente a los periodos de escases. \_

Debemos tomar en cuenta que el control de inventarios es utilizado no solamente para el control del producto sino que el sistema nos ayuda a controlar diferentes clases de negocios, así mismo en las empresas que laboran gran cantidad de personas, en el control del negocio de licores de gran volumen a nivel mundial.

Según (Pinto F, 2017) `De acuerdo con las diferentes fuentes de investigación se ha podido determinar que los controles de inventarios son el desarrollo y la implementación de sistemas computarizados los que detallan información de diferentes niveles: \_

#### **A nivel mundial**

**“Sistema de información para el control de inventarios del almacén del ITS”.**

Autor (es): MC. Maricela Sánchez López, Lic. Marcelino Vargas López, Ing. Blanca Alicia Reyes Luna, MC. Olga Lidia Vidal Vásquez.

País: México

Año: 2011

Objetivos de la investigación: Diseñar, desarrollar e implementar un sistema de información que permita llevar el registro de artículos, el control de las entradas y salidas de los mismos, así como todas aquellas actividades requeridas en el almacén del Instituto Tecnológico de Saltillo.

Síntesis: En base a las Tecnologías de Información se desarrolló un sistema de información, el cual se encuentra en la etapa de implementación en un proceso de pruebas y correcciones, como un soporte de apoyo administrativo que permita llevar el control de inventarios del almacén del ITS con la finalidad de ofrecer rapidez y seguridad en el manejo del inventario.

Cabe mencionar que este proyecto se encuentra en la fase de implementación y se corren algunas pruebas de instalación en el área de trabajo del almacén, se están corrigiendo algunas consultas e informes acordes a los requerimientos y necesidades de los usuarios.

### **Latinoamérica**

Según (Pinto F, 2017) A nivel Latinoamericana las diversas responsabilidades en lo que respecta al software de inventario afectan a muchos departamentos de las empresas especialmente a la pequeña ya que estos ejercen diferentes controles sobre los diferentes productos que hace negociar las empresas en cada negocio y mercadería que trabajan y estas mueven a través de los distintos procesos.

**“Diseño e implementación de un software de registro y control de inventarios”.**

Autor (es): Christian Camilo Suarez & Oscar Andrés Cuellar.

País: Colombia-2012

Objetivos de la investigación: Diseñar e implementar un software de registro para la empresa CENTROSUR que permita ser más eficiente en el manejo de la información de los clientes, proveedores y artículos a través de la sistematización.

Síntesis: Los beneficios que se obtendrán al diseñar e implementar un software para el registro y control de procesos para la empresa CENTROSUR, son el manejo acertado de la información institucional, cuyos resultados pretenden:

- Un apoyo de amplia dimensión para el funcionario en el momento de realizar las actividades en el sistema y la red a través de la sistematización.
- Alto porcentaje de utilidad tanto para los directivos como para todo el personal.
- Satisfacción total y expectativas para la implementación de software.
- Control del inventario total de mercancías, costos y rotación.
- Se logró identificar la necesidad fundamental que existe en el almacén frente al manejo de inventario de productos disponibles para la venta.

Partiendo de un proceso investigativo interno, se logró identificar los ítems fundamentales que debe presentar el aplicativo web para el diseño para que supla la necesidad de control e inspección del inventario de la empresa.

## **Ecuador**

El software de control de inventarios tiene como propósito apoyar en el manejo de control de mercadería de bodega, tanto en el ingreso de licores como también los licores que salen o se venden de la salsoteca. Es decir que los controles de inventarios juegan un papel muy importante para el buen funcionamiento, acorde a las necesidades en el área de bodega de la salsoteca y así se pueden afrontar la oferta y demanda de licores y satisfacer a los diferentes tipos de clientes.

### **2.1.2 ANTECEDENTES REFERENCIALES**

De acuerdo con el análisis realizado a diferentes fuentes de información sobre los sistemas de control de inventario de almacenamientos de licores se ha podido recolectar varias investigaciones relacionadas con este tema del diseño de control para la Salsoteca Barreto.

Según (Franco Zerna & Escalera Gómez, 2013) `El proyecto de diseño de un sistema de gestión y control de los inventarios de producto, en una empresa productora de fertilizantes simples y compuestos. Este escenario ha obligado a la empresa a tener un mayor control y gestión de los inventarios de producto terminado. Este modelo visto en el marco teórico y ajustado al escenario actual de la empresa. \_

Según (GUTIÉRREZ & PATRICIA, 2009) `Una de las estrategias para enfrentar dicha complejidad consiste en la implementación de herramientas informáticas de software. Esta estrategia presenta frecuentemente fallas en su implementación debido a que el entorno en el que las herramientas se utilizan es diferente del entorno para las cuales fueron desarrolladas. \_

**Título:** Diseño de un software informático de inventario que contribuya a mejorar el control de los ingresos y egresos de los productos naturales de la

microempresa FRUTOTERAPIA, siendo su autor Fausto Fidel Pinto Condoy del Instituto Tecnológico Bolivariano de Ecuador en julio del 2019.

En este proyecto se propone buscar una solución a una problemática en una microempresa que no cuenta con el control de los ingresos y egresos de los productos naturales de este tema en particular. Se propone diseñar un software el cual determinara las fallas del sistema actual de registro de mercadería que afecta el desarrollo, mediante la investigación descriptiva, explicativa, observación, encuestas, para diseñar un software de control de inventario que permita optimizar el tiempo en los procesos de la microempresa FRUTOTERAPIA. (Condoy, 2019)



**Figura 1.** Formulario de inicio de sesión en vista de diseño.

**Título:** Construcción de una guía operativa basada en software para la gestión y control de la producción y comercialización de la asociación de productores bananeros de Ventanas Aprobaven, siendo su autor Varas Beltrán, César Fabián de la Universidad Técnica Estatal de Quevedo.

En este proyecto se lo realizo necesidad de contar con un aplicativo web para controlar las labores de cultivo banano para la Asociación de Productores bananeros de Ventanas `APROBAVEN\_ dise¿ada con los estándares de la entidad antes mencionada. El análisis de requerimientos se

lo hizo mediante información brindada por la directiva de APROBAVEN, se recopiló información realizando encuestas a personas involucradas en el argot bananero. (Fabián, 2014)



**Figura 2.** Ingreso al sistema.

Se ingresa el sistema y se escoge la opción Registrarme y se procede a registrar los datos solicitados para ingresar un nuevo usuario. (Fabián, 2014)

**Título:** Sistema para el control de ventas e inventarios de la empresa Antigua Arte Europeo S.A de C.V, siendo su autor Mederic Antonio Villa Gutiérrez de la Universidad Autónoma del Estado de Hidalgo del país de México

En este proyecto busca una aplicación informática que permita optimizar los procesos en ventas e inventarios haciendo uso del equipo de cómputo existente y con este facilitar el almacenamiento, control de la información que se maneja en cada uno de los procesos mencionados, y que además agilice las consultas a dicha información precisa y ordenada al momento que los administradores, la propietaria o el personal de la empresa la requieran. (Gutiérrez, 2007) **Figura 3.** Formulario de inicio de sesión en vista de diseño.



## 2.2 Aspectos teóricos fundamentales

### 2.2.1 Paradigmas de programación para sistemas de cómputo



**Figura 4.** Paradigmas de la Programación

[https://www.cienciasfera.com/materiales/informatica/tecnologiainformacion/tema21/12\\_paradigmas\\_de\\_la\\_programacin.html](https://www.cienciasfera.com/materiales/informatica/tecnologiainformacion/tema21/12_paradigmas_de_la_programacin.html)

Los Lenguajes de Programación (LP) proporcionan mecanismos para implementar una filosofía o paradigma de programación. Un paradigma es una forma de entender y representar la realidad: un conjunto de teorías, estándares y métodos que, juntos, representan un modo de organizar el pensamiento, es decir un modo de ver el mundo. (Ponce de León Amador & Saquete Boro, Introducción a la Programación Orientada a Objetos, 2006)

Cada nuevo paradigma responde a una necesidad real de nuevos modos de afrontar problemas. A menudo un nuevo paradigma es creado como respuesta a las deficiencias de paradigmas anteriores. (Ponce de León Amador & Saquete Boro, Introducción a la Programación Orientada a Objetos, 2006)

Un paradigma de programación es una forma de conceptualizar en que consiste la ejecución de un programa y cómo deben estructurarse y organizarse las tareas que se llevan a cabo en esa ejecución

(Ponce de León Amador & Saquete Boro, Introducción a la Programación Orientada a Objetos, 2006)

### **Paradigma Funcional.**



**Figura 5.** Paradigma Funcional

<https://www.udla.edu.ec/2017/06/16/paradigma-de-programacion-funcional/>

Un lenguaje de programación funcional describe procesos mediante la composición de funciones, entendidas éstas no como los clásicos subprogramas, sino como funciones puramente matemáticas. Entre las características propias de estos lenguajes está la no existencia de asignaciones de variables y la falta de estructuras de control lo que conlleva a menudo el uso de funciones recursivas para implementar la repetición de instrucciones. (Ponce de León Amador & Saquete Boro, Introducción a la Programación Orientada a Objetos, 2006)

## Paradigma Lógico

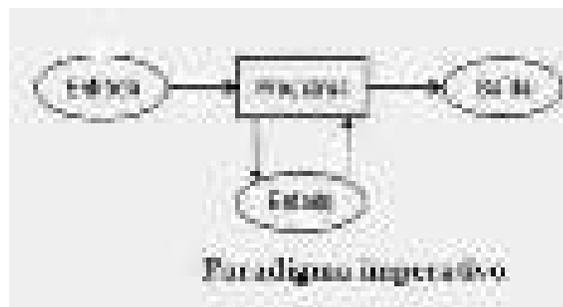


**Figura 6.** Paradigma Lógico

<https://es.slideshare.net/JFREDYOLAYARAMOS/paradigma-logico>

La paradigma lógica o declarativa consiste en la aplicación del corpus de conocimiento sobre lógica para el diseño en los lenguajes de programación. La mayoría de los lenguajes de programación lógica se basan en la teoría lógica de primer orden. (Ponce de León Amador & Saquete Boro, Introducción a la Programación Orientada a Objetos, 2006)

## Paradigma Imperativo



La programación imperativa es un paradigma de programación que describe la programación en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea. La mayoría de los lenguajes imperativos de alto nivel incorporan cuatro tipos

básicos de sentencias: asignaciones, ciclos, ramificaciones condicionales y ramificaciones incondicionales. (Ponce de León Amador & Saquete Boro, Introducción a la Programación Orientada a Objetos, 2006)

### **Paradigma Orientado a Objetos**



**Figura 8.** Paradigma Orientado a Objeto

[https://www.ecured.cu/Programaci%C3%B3n Orientada a Objetos](https://www.ecured.cu/Programaci%C3%B3n%20Orientada%20a%20Objetos)

El paradigma orientado a objetos (OO) define los programas en términos de comunidades de objetos. Los objetos son características comunes se agrupan en clases (un concepto asimilable al de tipo abstracto de dato (TAD)). Los objetos son entidades que combinan un estado (es decir datos) y un comportamiento (esto es, procedimiento o métodos). Estos objetos se comunican entre ellos para realizar tareas. (Ponce de León Amador & Saquete Boro, Introducción a la Programación Orientada a Objetos, 2006)

## 2.2.2 Lenguajes y Tecnologías de Escritorio.

### 2.2.2.1 Software de Aplicación para Escritorio.



**Figura 9.** Software de Aplicación para Escritorio

<https://promwebsoft.com/desarrollo-de-software>

Una aplicación de escritorio es aquella que se encuentra instalado en el ordenador o sistema de almacenamiento (USB) y podemos ejecutarlo sin internet en nuestro sistema operativo, al contrario que las aplicaciones en la nube que se encuentran en otro ordenador (servidor) al que accedemos a través de la red o internet a su software. (Wikipedia, La enciclopedia libre, 2019)

### 2.2.2.2 Lenguajes de Programación

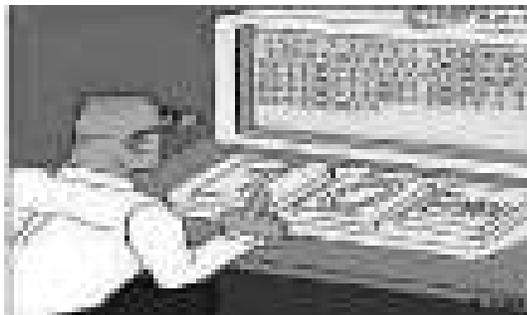


**Figura 10.** Lenguajes De Programación

<http://developeando.net/elegir-lenguaje-de-programacion/>

Un lenguaje de programación es un lenguaje formal que proporciona una serie de instrucciones que permiten a un programador escribir secuencias de órdenes y algoritmos a modo de controlar el comportamiento físico y lógico de una computadora con el objetivo de que produzca diversas clases de datos. A todo este conjunto de órdenes y datos escritos mediante un lenguaje de programación se le conoce como programa. (Wikipedia C. d., Lenguaje de Programación, 2019)

### **De Bajo nivel**

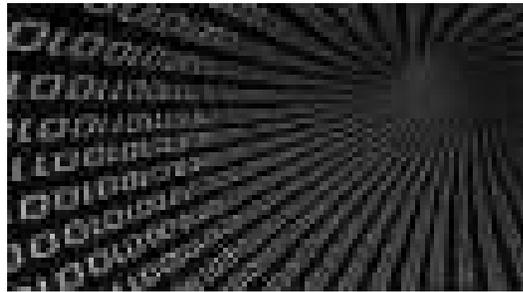


**Figura 11. Bajo Nivel**

<https://zimbronapps.com/sistemas-computacionales/fundamentos-de-programacion/tipos-de-lenguaje-de-programacion/>

Son mucho más fáciles de utilizar que el lenguaje máquina, pero dependen mucho de la máquina o computadora como sucedía con el lenguaje máquina.

## De Maquina



**Figura 12.** De Maquina

<https://www.timetoast.com/timelines/historia-del-lenguaje-de-maquina>

El lenguaje máquina contiene generalmente un número limitado de instrucciones, que incluyen un código de operación y una o varias direcciones de operandos y resultados en memoria central, codificados todos estos elementos de forma binaria. (Wikipedia C. d., Arquitectura de software, 2019)

Una vez que el ordenador se pone en marcha, el procesador va obteniendo, una tras otra, las instrucciones que se suponen almacenadas en posiciones sucesivas de memoria, las analiza y las ejecuta. Así, empezará por la palabra de la dirección 0, luego la de la dirección 1. (Wikipedia C. d., Arquitectura de software, 2019)

## Ensamblador



**Figura 13.** Ensamblador

<https://jrking95.wixsite.com/isc6semestre/single-post/2016/1/27/%C2%BFQu%C3%A9-es-el-Lenguaje-Ensamblador>

La misión de un ensamblador es la de simplificar la programación de un determinado ordenador, manteniendo un control directo del hardware del mismo, por lo que cada tipo de ordenador tiene su propio ensamblador, que no está demasiado lejos del código máquina.

### De Alto nivel



**Figura 14.** De Alto Nivel

<https://conociendoalgoritmo.weebly.com/lenguaje-de-alto-nivel.html>

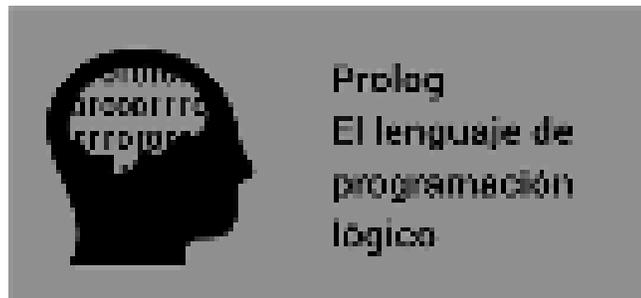
Su desarrollo comenzó a mediados de los años 50. Se había puesto de manifiesto que la mayor limitación de los ordenadores se encontraba a nivel software y no a nivel hardware, pues escribir programas en lenguajes de bajo nivel era una tarea larga, difícil y cara. (Wikipedia C. d., Arquitectura de software, 2019)

Un lenguaje de alto nivel es un lenguaje orientado hacia la resolución de una determinada clase de problemas, mientras que un lenguaje de bajo nivel es un lenguaje orientado a una determinada clase de máquinas. (Wikipedia C. d., Arquitectura de software, 2019)

## Lógico

La programación lógica los programas se consideran como una serie de aseveraciones lógicas. De esta forma, el conocimiento se representa mediante reglas, tratándose de sistemas declarativos.

## Prolog



**Figura 15.** Prolog

<https://www.somosbinarios.es/prolog-el-lenguaje-de-programacion-logico/>

Prolog es un lenguaje de programación seminterpretado. Su funcionamiento es muy similar a Java. El código fuente se compila a un código de byte el cuál se interpreta en una máquina virtual denominada Warren Abstract Machine (comúnmente denominada WAM).

Por eso, un entorno de desarrollo Prolog se compone de:

- Un compilador: Transforma el código fuente en código de byte. A diferencia de Java, no existe un Standard al respecto. Por eso, el código de byte generado por un entorno de desarrollo no tiene por qué funcionar en el intérprete de otro entorno.
- Un intérprete: Ejecuta el código de byte. Un shell o top-level. Se trata de una utilidad que permite probar los programas, depurarlos, etc. Su funcionamiento es similar a los interfaces de línea de comando de los sistemas operativos.

- Una biblioteca de utilidades: Estas bibliotecas son, en general, muy amplias. Muchos entornos incluyen (afortunadamente) unas bibliotecas standard-ISO que permiten funcionalidades básicas como manipular cadenas, entrada/salida, etc. (EcuRed, Prolog (lenguaje de programación), 2019)

## Funcionales

### Scala



**Figura 16.** Scala

<https://andalinux.wordpress.com/2016/07/05/scala-un-lenguaje-que-esta-dando-que-hablar/>

Scala es un lenguaje de programación multi-paradigma diseñado para expresar patrones comunes de programación en forma concisa, elegante y con tipos seguros. Integra sutilmente características de lenguajes funcionales y orientados a objetos. La implementación actual corre en la máquina virtual de Java y es compatible con las aplicaciones Java existentes. (Wikipedia C. d., Scala, 2019)

## Lisp



**Figura 17.** Lisp

<https://www.timetoast.com/timelines/evolucion-de-los-lenguajes-de-programacion-bea8fb3c-869d-41b2-80a2-5aff8339a337>

Lisp (históricamente LISP) es una familia de lenguajes de programación de computadora de tipo multiparadigma con larga historia y una inconfundible y útil sintaxis basada en la notación polaca.

Desarrollado originalmente en 1958 por John McCarthy y sus colaboradores en el Instituto Tecnológico de Massachusetts, Lisp es el segundo lenguaje de programación de alto nivel de mayor antigüedad entre los que continúan teniendo un uso extendido en la actualidad; únicamente FORTRAN es anterior. (Wikipedia C. d., Lisp, 2019)

## Scheme



**Figura 18.** Scheme

<https://www.ecured.cu/Scheme>

Scheme es un lenguaje de programación funcional (si bien impuro pues sus estructuras de datos no son inmutables) y un dialecto de Lisp. Fue desarrollado por Guy L. Steele y Gerald Jay Sussman en la década de los setenta e introducido en el mundo académico a través de una serie de artículos conocidos como los Lambda Papers de Sussman y Steele.

La filosofía de Scheme es minimalista. Su objetivo no es acumular un gran número de funcionalidades, sino evitar las debilidades y restricciones que hacen necesaria su adición. Así, Scheme proporciona el mínimo número posible de nociones primitivas, construyendo todo lo demás a partir de un reducido número de abstracciones. Por ejemplo, el mecanismo principal para el control de flujo son las llamadas recursivas. (Wikipedia C. d., Scheme, 2019)

### **Imperativos y orientado a objetos**

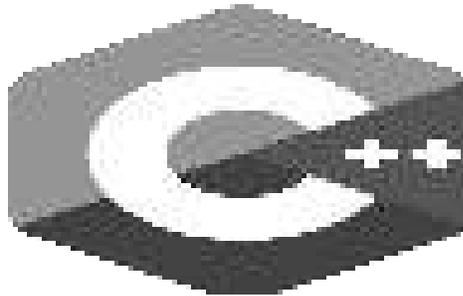


**Figura 19.** Imperativos y Orientado a Objetos

<https://styde.net/que-es-la-programacion-orientada-a-objetos/>

Un lenguaje imperativo es un tipo de lenguaje de programación. Es este tipo de lenguajes, las instrucciones se ejecutan unas tras otras, de manera secuencial, salvo cuando se encuentran estructuras de control condicionales o bucles. Hay declaración de variables, tipos y procedimientos, aunque esto varía notablemente en función del lenguaje utilizado, pues los hay que exigen las declaraciones mientras que otros permiten que esos elementos no sean declarados.

C/C++



**Figura 20. C++**

<https://es.wikipedia.org/wiki/C%2B%2B>

El éxito comercial de la tecnología de objetos se basa en la aparición y tremendo éxito comercial de lenguajes que añadieron extensiones OO a un lenguaje no OO muy difundido: C.

El primer esfuerzo de este tipo que atrajo la atención fue Objective-C. El más conocido hoy en día es C++, que fue diseñado originariamente por Bjarne Stroustrup en los laboratorios Bell de AT&T. C++ comienza a desarrollarse en 1980 y se implanta rápidamente como modo de obtener beneficios de programación OO manteniendo al mismo tiempo compatibilidad con C. El mérito de C++ es el impulso histórico que dio a la tecnología de objetos, haciéndola fácil de presentar a toda la comunidad, que no habría aceptado esas ideas bajo una apariencia menos convencional. Que C++ no sea el LOO ideal, comentario hecho frecuentemente por autores y conferenciantes, no debería oscurecer su contribución (Ponce de León Amador & Saquete Boro, Introducción a la programación orientada a objetos, 2006) (Stroustrup, 2001).

## JAVA



**Figura 21.** Java

<http://www.ictea.com/cs/knowledgebase.php?action=displayarticle&id=8790>

En 1996 un equipo de SUN Microsystems presentó un nuevo LEO desarrollado a partir de una versión simplificada de C++: Java. La principal contribución de Java radica en la tecnología de implementación. Construido con ideas ya presentes en otros muchos entornos OO, estas ideas son recogidas aquí bajo un nuevo enfoque, ya que la ejecución de Java se basa en dos pilares:

1. Código con formato de bajo nivel, portable e interpretable, cuya especificación es de dominio público.
2. Máquina virtual fácilmente disponible que interpreta los programas en este código. (Ponce de León Amador & Saquete Boro, Introducción a la programación orientada a objetos, 2006)

**C#**



**Figura 22. C#**

<https://lenguajedeprogramacionblog.wordpress.com/c-2/>

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 (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. (Wikipedia C. d., C Sharp, 2019)

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. (Wikipedia C. d., C Sharp, 2019)

### **Visual Basic .Net**



**Figura 23. Visual .Net**

<https://reyesjoseling.wordpress.com/historia-de-visual-studio/>

Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET. Su introducción resultó muy controvertida, ya que debido a cambios significativos en el lenguaje VB.NET no es retro compatible con Visual Basic, pero el manejo de las instrucciones es similar a versiones anteriores de Visual Basic, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas. Para mantener eficacia en el desarrollo de las aplicaciones. La gran mayoría de programadores de VB.NET utilizan el entorno de desarrollo integrado Microsoft Visual Studio en alguna de sus versiones (desde el primer Visual Studio .NET hasta Visual Studio .NET 2019, que es la última versión de Visual Studio para la plataforma .NET), aunque existen otras alternativas, como SharpDevelop (que además es libre).

## Python



**Figura 24.** Python

<https://es.wikipedia.org/wiki/Python>

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida,

programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. (Wikipedia C. , Python, 2019)

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores. (Wikipedia C. , Python, 2019)

## **Ruby**



**Figura 25.** Ruby

<http://educagratis.cl/moodle/course/view.php?id=208>

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre. (Wikipedia C. , Ruby, 2019)

### 2.2.2.3 Arquitectura de Software



**Figura 26.** Arquitectura de Software

<https://escuela.it/cursos/curso-arquitectura-software>

La arquitectura de software es el diseño de más alto nivel de la estructura de un sistema.

Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software. (Wikipedia C. d., Arquitectura de software, 2019)

Una arquitectura de software se selecciona y diseña con base en objetivos (requisitos) y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real. (Wikipedia C. d., Arquitectura de software, 2019)

La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la

comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea. (Wikipedia C. d., Arquitectura de software, 2019)

### **Descomposición Modular**



**Figura 27.** Descomposición Modular

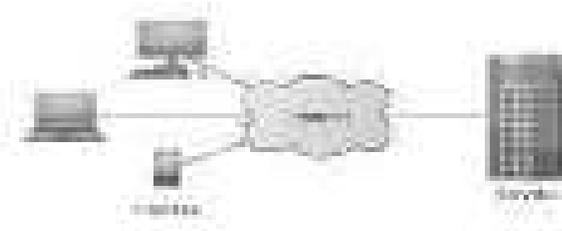
<https://prezi.com/tobju9knuloi/descomposicion-modular-y-patrones-de-diseno/>

Uno de los métodos fundamentales para resolver un problema es dividirlo en problemas más pequeños, llamados subproblemas, pudiendo éstos, a su vez, ser subdivididos repetidamente en problemas más pequeños hasta llegar a obtener subprogramas lo suficientemente reducidos como para que resuelvan una única tarea y sean, por tanto, programados de forma más sencilla. Esto facilita, además, la detección de errores y la elaboración del programa, pues es más fácil pensar cómo resolver muchas tareas pequeñas, que pensar cómo resolver una sola más grande y compleja. (Wikipedia C. d., Arquitectura de software, 2019)

Esta técnica de dividir el problema principal en subproblemas se denomina frecuentemente divide y vencerás o programación modular. Además, si hay varias personas para hacer un programa, al dividir éste en fragmentos, hacemos posible que cada programador se pueda dedicar a uno o varios fragmentos, que posteriormente serán enlazados, reduciendo

considerablemente el tiempo de programación. (Wikipedia C. d., Arquitectura de software, 2019)

### **Cliente/Servidor**



**Figura 28.** Cliente-Servidor

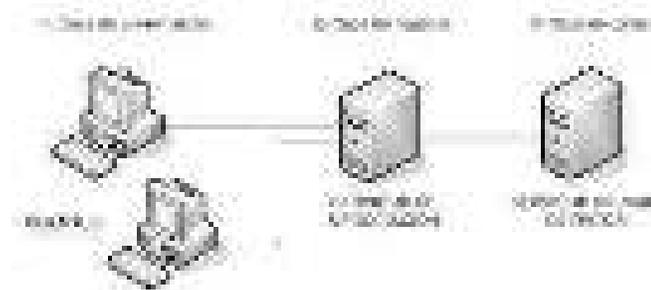
<https://es.wikipedia.org/wiki/Cliente-servidor>

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores estos últimos responden a la demanda del cliente que la produjo. (EcuRed, Arquitectura Cliente Servidor, 2019)

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa una red mundial como lo es la Internet. (EcuRed, Arquitectura Cliente Servidor, 2019)

Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura. (EcuRed, Arquitectura Cliente Servidor, 2019)

## Arquitectura de Tres Niveles



**Figura 29.** Arquitectura de Tres Niveles

[https://www.ecured.cu/Arquitectura de tres niveles](https://www.ecured.cu/Arquitectura_de_tres_niveles)

Arquitectura de tres niveles. La arquitectura de software incluye los aspectos estáticos y dinámicos más significativos del software que se desea crear. De acuerdo Robert Pressman, la arquitectura de software no es otra cosa que "una descripción de los subsistemas y los componentes de un sistema informático y las relaciones entre ellos". De igual manera, la arquitectura de software de tres niveles, incluye todos estos aspectos, y además, brinda mejores opciones para proyectos informáticos de gran alcance y complejidad. (EcuRed, Arquitectura de Tres Niveles, 2019)

Para enfrentarse a estos temas, la comunidad de software desarrolló la noción de una arquitectura de tres niveles. La aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido. La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo. (EcuRed, Arquitectura de Tres Niveles, 2019)

La capa intermedia, o capa de empresa, consiste en la aplicación o lógica de empresa, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación. La capa intermedia (lógica de aplicación) es básicamente el código al que recurre la capa de presentación para recuperar los datos

deseados. La capa de presentación recibe entonces los datos y los formatea para su presentación. (EcuRed, Arquitectura de Tres Niveles, 2019)

### **Capa Presentación**



**Figura 30.** Capa Presentación

<https://www.slideshare.net/robertovasconezbarragan7/capa-de-presentacin-modelo-osi>

Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario, le presenta la información y obtiene la información del usuario en un mínimo de proceso. En el mundo de la informática es conocida como interfaz gráfica y debe tener la característica de ser amigable, o sea, entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa intermedia o de negocio. (EcuRed, Arquitectura de Tres Niveles, 2019)

## Capa de Negocio

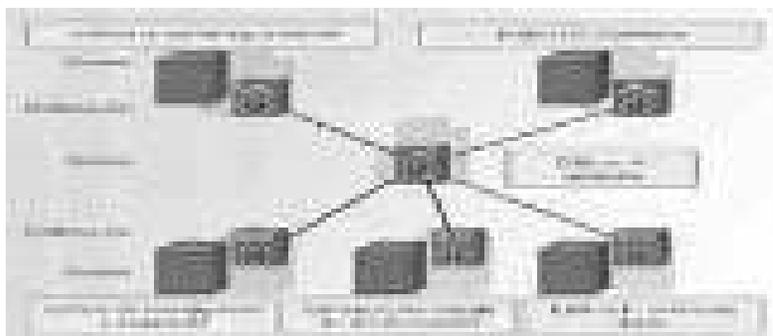


**Figura 31.** Capa de Negocio

<https://sites.google.com/site/tuxnots/materias-de-la-facu/metodologia-de-sistemas/arquitecturaencapaspatronesparacapadenegocioyaccesoadatosdomaindrivendesignsoftwareparaormscaracteristicas>

Es donde residen las funciones que se ejecutan, se reciben las peticiones del usuario, se procesa la información y se envían las respuestas tras el proceso. Se denomina capa de negocio o capa de lógica del negocio, porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. (EcuRed, Arquitectura de Tres Niveles, 2019)

## Capa de Acceso de Datos

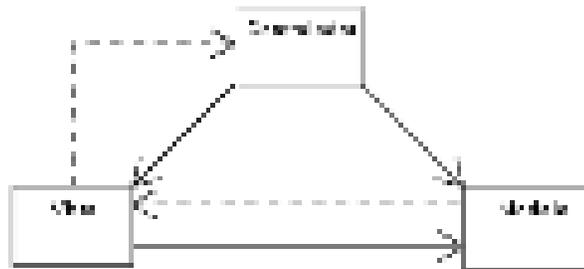


**Figura 32.** Capa de Acceso de Datos

<https://interpolados.wordpress.com/2017/05/01/capas-de-acceso-distribucion-y-de-nucleo-central/>

Esta capa es la encargada de almacenar los datos del sistema y de los usuarios. Su función es almacenar y devolver datos a la capa de negocio aunque para esto también es necesario en algunos casos, que tengan procedimientos almacenados y funciones dentro de la capa. En una arquitectura de tres capas, esta capa es la única que puede acceder a los mismos. Está formada por uno o varios sistemas gestores de bases de datos, localizados en un mismo servidor o en varios. (EcuRed, Arquitectura de Tres Niveles, 2019)

### **Modelo Vista Controlador (MVC)**



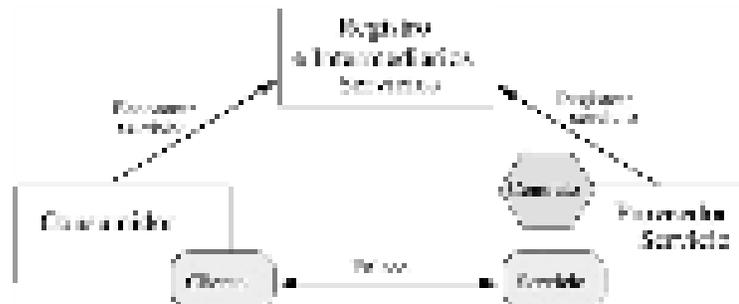
**Figura 33.** Modelo Vista Controlador

<https://es.wikipedia.org/wiki/Modelo%20%80%93vista%20%80%93controlador>

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de

conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. (Wikipedia C. d., Modelo-Vista-Controlador, 2019)

### Orientada a Servicios



**Figura 34.** Orientada a Servicios

[https://www.researchgate.net/figure/Figura-25-Modelo-conceptual-de-una-arquitectura-orientada-a-servicios\\_fig3\\_33399284](https://www.researchgate.net/figure/Figura-25-Modelo-conceptual-de-una-arquitectura-orientada-a-servicios_fig3_33399284)

Arquitectura orientada al servicio (SOA). Es la primera arquitectura de Tecnologías de Información (TI) que asume lo que los negocios han sabido desde hace mucho tiempo. Se trata esencialmente de un set de servicios sueltos, donde cada uno es relativamente económico para construirlo o reemplazarlo si es necesario. (EcuRed, Arquitectura Orientada a Servicios (SOA), 2019)

### Arquitectura de Microservicios



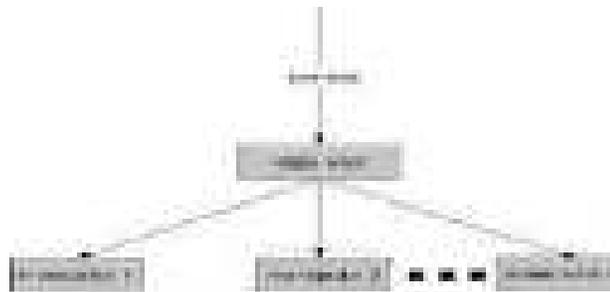
**Figura 35.** Arquitectura de Microservicios

<https://www.youtube.com/watch?v=TSiso1F6oqw>

La arquitectura de microservicios (en inglés, Micro Services Architecture, MSA) es una aproximación para el desarrollo de software que consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso y se comunican con mecanismos ligeros (normalmente una API de recursos HTTP). Cada servicio se encarga de implementar una funcionalidad completa del negocio. Cada servicio es desplegado de forma independiente y puede estar programado en distintos lenguajes y usar diferentes tecnologías de almacenamiento de datos. (Wikipedia C. d., Arquitectura de microservicios, 2019)

Se suele considerar la arquitectura de microservicios como una forma específica de realizar una arquitectura orientada a servicios. (Wikipedia C. d., Arquitectura de microservicios, 2019)

### **Dirigida por Eventos**



**Figura 36.** Dirigida por Eventos

<https://www.genbeta.com/desarrollo/introduccion-a-la-programacion-dirigida-por-eventos>

Un evento puede ser definido como "un cambio significativo en un estado". Por ejemplo, cuando un consumidor compra un coche, el estado del coche

pasa de "se vende" a "vendido". La arquitectura del sistema del vendedor de coches debe tratar este cambio de estado como un evento, cuyo suceso puede ser conocido en otras aplicaciones en la arquitectura. Desde una perspectiva formal, lo que es producido, publicado, propagado, detectado o consumido es un mensaje (típicamente asíncrono) llamado notificación del evento, y no el evento en sí mismo, el cual es el cambio de estado que disparó la emisión del evento. Los eventos no viajan, solamente ocurren. Por otro lado, el término evento es frecuentemente usado para denotar el mensaje de notificación en sí mismo, lo cual puede llevar a algún tipo de confusión. (Wikipedia C. d., Arquitectura dirigida por eventos, 2019)

#### 2.2.2.4 Base de Datos



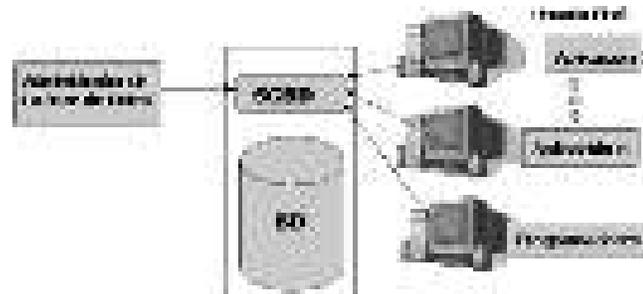
**Figura 37.** Base de Datos

<https://mistertek.com/captar-base-datos-empresas-mail-marketing/>

El diseño de bases de datos es un proceso complejo que permite obtener una implementación de una base de datos que satisface los requisitos informacionales de un sistema de información. El diseño se realiza a partir de los requisitos del sistema de información. Este proceso guía al diseñador de bases de datos por varias etapas con el objetivo de segmentar un problema de una complejidad considerable en diferentes sub-problemas de menor complejidad hasta llegar a la solución final: la implementación de una base

de datos que satisface los requerimientos funcionales y no funcionales de un sistema de información. (Casas Roma & Conesa i Caralt, 2013)

### Sistema de Administración de Base de Datos



**Figura 38.** Sistema de Administración de Base de Datos

<http://www.mailxmail.com/curso-diseno-creacion-bases-datos/sistemas-gestion-bases-datos-concepto-diseno-mer-normalizacion-1>

Un Sistema de Administración de Bases de Datos (DBMS) es software que permite a una organización centralizar los datos, administrarlos en forma eficiente y proveer acceso a los datos almacenados mediante programas de aplicación. El DBMS actúa como una interfaz entre los programas de aplicación y los archivos de datos físicos. Cuando el programa de aplicación solicita un elemento de datos, como el sueldo bruto, el DBMS lo busca en la base de datos y lo presenta al programa de aplicación. Si utilizara archiva de datos tradicionales, el programador tendría que especificar el tamaño y formato de cada elemento de datos utilizado en el programa y después decir a la computadora en dónde están ubicados. (GUIDOBEYMARPEREZLOPEZ, 2015)

## SQL (Lenguaje de Consultas y Transacciones a Bases de Datos)



**Figura 39.** SQL

<https://tutobasico.com/introduccion-lenguaje-sql/>

El SQL es el lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales. Es un lenguaje declarativo: sólo hay que indicar qué se quiere hacer. En cambio, en los lenguajes procedimentales es necesario especificar cómo hay que hacer cualquier acción sobre la base de datos. El SQL es un lenguaje muy parecido al lenguaje natural; concretamente, se parece al inglés, y es muy expresivo. Por estas razones, y como lenguaje estándar, el SQL es un lenguaje con el que se puede acceder a todos los sistemas relacionales comerciales. (Martín Escofet, 2002)

## SQL Server (Microsoft)



**Figura 40.** SQL Server (Microsoft)

<https://aws.amazon.com/es/rds/sqlserver/>

Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL). (Wikipedia C. d., Microsoft SQL Server, 2019)

Dentro de los competidores más destacados de SQL Server están: Oracle, MariaDB, MySQL, PostgreSQL. SQL Server ha estado tradicionalmente disponible solo para sistemas operativos Windows de Microsoft, pero desde 2016 está disponible para GNU/Linux, y a partir de 2017 para Docker también. (Wikipedia C. d., Microsoft SQL Server, 2019)

Puede ser configurado para utilizar varias instancias en el mismo servidor físico, la primera instalación lleva generalmente el nombre del servidor, y las siguientes - nombres específicos (con un guion invertido entre el nombre del servidor y el nombre de la instalación). (Wikipedia C. d., Microsoft SQL Server, 2019)

## **Oracle**



**Figura 41.** Oracle

<https://trellat.es/exportar-importar-estadisticas-en-oracle/>

Oracle Database es un sistema de gestión de base de datos de tipo objeto-relacional (ORDBMS, por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation. (Wikipedia C. d., Oracle Database, 2019)

Su dominio en el mercado de servidores empresariales había sido casi total hasta que recientemente tiene la competencia del Microsoft SQL Server y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. (Wikipedia C. d., Oracle Database, 2019)

Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux. (Wikipedia C. d., Oracle Database, 2019)

## **MySQL**



**Figura 42.** MySQL

<https://www.muysseguridad.net/2016/09/13/mysql-dos-vulnerabilidades-criticas/>

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. (Wikipedia C. d., MySQL, 2019)

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL. (Wikipedia C. d., MySQL, 2019)

### **PostgreSQL**



**Figura 43.** PostgreSQL

<https://www.enterprisedb.com/fr/products/edb-postgresql>

El lenguaje estructurado de consultas (SQL) es un lenguaje de base de datos normalizado, utilizado por la gran mayoría de los servidores de bases de datos que manejan bases de datos relacionales u objeto-relacionales.

Es un lenguaje declarativo en el que las órdenes especifican cual debe ser el resultado y no la manera de conseguirlo (como ocurre en los lenguajes procedimentales). Al ser declarativo es muy sistemático, sencillo y con una curva de aprendizaje muy agradable ya que sus palabras clave permiten escribir las órdenes como si fueran frases en las que se especifica (en inglés) que es lo que queremos obtener. (Sarría)

## NoSQL



**Figura 44.** NoSQL

<https://pragatisoftware.com/an-introduction-to-nosql/>

Es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico de SGBDR (Sistema de Gestión de Bases de Datos Relacionales) en aspectos importantes, siendo el más destacado que no usan SQL como lenguaje principal de consultas. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad) y habitualmente escalan bien horizontalmente. Los sistemas NoSQL se denominan a veces "no sólo SQL" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo SQL. (Wikipedia C. d., NoSQL, 2019)

Aunque muchas empresas no gestionaban las escalas de datos de las empresas antes mencionadas, sin embargo, muchas de ellas empezaron a diseñar sus aplicaciones para ejecutarse en estos ambientes distribuidos y usar este tipo de bases de datos al descubrir las ventajas que ofrecían. (Wikipedia C. d., NoSQL, 2019)

## Redis



**Figura 45.** Redis

<http://www.securitybydefault.com/2014/02/12-seguridad-en-redis-fortificacion.html>

Redis es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes (clave/valor) pero que opcionalmente puede ser usada como una base de datos durable o persistente. Está escrito en ANSI C por Salvatore Sanfilippo, quien es patrocinado por Redis Labs. Está liberado bajo licencia BSD por lo que es considerado software de código abierto. (Wikipedia C. d., Redis, 2019)

## MongoDB



**Figura 46.** MongoDB

<https://magazine.odroid.com/es/article/mongodb-on-a-c2-with-ubuntu-18-04/>

MongoDB (del inglés humongous, "enorme") es un sistema de base de datos NoSQL orientado a documentos de código abierto. (Wikipedia C. d., MongoDB, 2019)

En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. (Wikipedia C. d., MongoDB, 2019)

MongoDB es una base de datos adecuada para su uso en producción y con múltiples funcionalidades. Esta base de datos se utiliza mucho en la industria, contando con implantaciones en empresas como MTV Network, Craigslist, BCI o Foursquare. (Wikipedia C. d., MongoDB, 2019)

El código binario está disponible para los sistemas operativos Windows, GNU/Linux, OS X y Solaris. (Wikipedia C. d., MongoDB, 2019)

## **CouchDB**



**Figura 47.** CouchDB

<http://couchdb.apache.org/>

Es un gestor de bases de datos de código abierto, cuyo foco está puesto en la facilidad de su uso y en ser "una base de datos que asume la web de manera completa". Se trata de una base de datos NoSQL que emplea JSON para almacenar los datos, JavaScript como lenguaje de consulta por medio de MapReduce y HTTP como API. Una de sus características más peculiares es la facilidad con la que permite hacer replicaciones. CouchDB fue liberada por primera vez en 2005, transformándose en un proyecto Apache en 2008. (Wikipedia C. d., CouchDB, 2019)

### 2.2.3 Frameworks de Escritorio (Marco de trabajo)

Un framework, entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. (Wikipedia C. d., 2019)

- Java: Es un lenguaje de programación orientado a objetos. Su principal intención es poder escribir aplicaciones que se ejecuten en cualquier dispositivo (WORA, Write Once, Run Anywhere).

**Frameworks:** Spring, Dropwizard, Blade.



**Figura 48.** Framework Spring

<http://www.javahispano.org/espanha/2014/5/7/charla-going-real-time-with-websockets-and-spring.html>

- Python: Es un lenguaje de programación de alto nivel creado por Guido van Rossum en 1991. Entre sus características está lo fácil de usar y leer bajo su filosofía de DRY (Don't Repeat Yourself).

**Frameworks:** Django, Pyramide, Web2py.



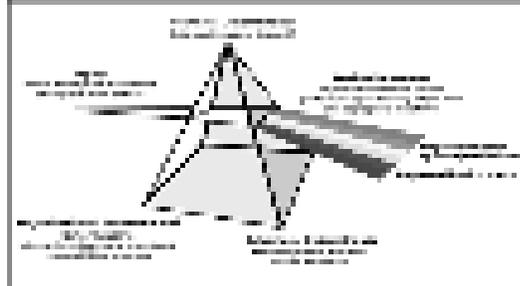
**Figura 49.** Django

[https://es.wikipedia.org/wiki/Django\\_\(framework\)](https://es.wikipedia.org/wiki/Django_(framework))

Propietaria o licencias de código abierto GPL y LGPL, hace posible tanto el uso libre como la explotación comercial. (Guide, 2019)

**C#**

**Prism**



**Figura 50.** Prism

[https://www.researchgate.net/figure/Prism-Framework-for-understanding-Health-Information-System-performance\\_fig1\\_7200428](https://www.researchgate.net/figure/Prism-Framework-for-understanding-Health-Information-System-performance_fig1_7200428)

Por un lado tenemos Prism, uno de los frameworks de este tipo más antiguos, con más solera y uno de los que están más testeados. Además es uno de los frameworks que recomienda Microsoft. (Ortega, 2019)

**MVVM Light Toolkit**



**Figura 51.** MVVM Light Toolkit

<http://www.mvvmlight.net/installing/>

Otro de los más años lleva funcionando es MVVM Light Toolkit, que además funciona tanto para C# como para Xamarin. (Ortega, 2019)

## MVVM Cross



**Figura 52. MVVM CROSS**

<https://javersuarezruiz.wordpress.com/2017/11/21/xamarin-forms-utilizando-mvvmcross/>

Este framework se desarrolló inicialmente para Xamarin, aunque en la actualidad se puede desarrollar también con Windows Presentation Foundation y demás. (Ortega, 2019)

Es un framework bastante potente, mucho más que los dos anteriores, tiene una comunidad muy fuerte y está en constante actualización. (Ortega, 2019)

## REACTIVE UI



**Figura 53. ReactiveUI**

<https://reactiveui.net/>

El más reciente, pero no por ello menos importante, es ReactiveUI. Es un framework MVVM un poco más pequeño que el resto, ya que no tiene tanto soporte, a nivel de framework MVVM a nivel, de la comunidad, pero no por ello deja de ser bastante potente también. (Ortega, 2019)

Su principal virtud es que no solo ayuda con la arquitectura MVVM, sino que también está orientado a la programación reactiva, lo que nos permite ser más eficientes a la hora de desarrollar código. (Ortega, 2019)

## **.NET**



**Figura 54. .NET**

<https://www.howtogeek.com/253588/what-is-the-microsoft-net-framework-and-why-is-it-installed-on-my-pc/>

.NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones. Basada en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado. (Wikipedia C. d., Microsoft .Net, 2019)

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Oracle Corporation 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. (Wikipedia C. d., Microsoft .Net, 2019)

## Python

### Los 4 frameworks web más populares para Python

- Pyramid - Flexible. Pyramid nació de una fusión entre Pylons 1.0 y repoze.bfg.
- Bottle - Simple. Bottle es un microframework muy simple que proporciona un mínimo de herramientas al desarrollador (enrutamiento, plantillas y una pequeña abstracción sobre WSGI).
- Django - Potente.
- Flask - Ágil. (Programación.Net)

## Ruby

### Ruby On Rails



**Figura 55.** Ruby On Rails

<https://www.aprendum.com/curso-online-ruby-on-rails/>

Si ya estás familiarizado con Ruby es probable que ya hayas escuchado sobre esta framework o incluso ya la estés utilizando. Ruby on Rails es una framework de código abierto. Es una de las frameworks más populares para Ruby. El objetivo principal de Rails es favorecer la convención antes de la configuración. Como muchas otras frameworks, se especializa en disminuir la repetición de código. (Informativa, 2019)

## Sinatra



**Figura 56.** Sinatra

<https://medium.com/@hanna.soloman/step-by-step-guide-to-setting-up-a-sinatra-framework-for-your-app-2997ad9d5642>

Sinatra es un lenguaje específico de dominio (DSL) para crear aplicaciones web en Ruby de manera sencilla. Sinatra es en sí, una librería, pero debido a su popularidad la hemos incluido como una opción más en esta lista. De hecho, algunos desarrolladores la consideran una framework completa y ha servido como base para la creación de otras frameworks que mencionaremos más adelante. (Informativa, 2019)

## Hobbit



**Figura 57.** Hobbit

[https://www.slideshare.net/hobbit\\_project/hobbit-in-a-nutshell-edf2016](https://www.slideshare.net/hobbit_project/hobbit-in-a-nutshell-edf2016)

Esta micro-framework ha sido desarrollada teniendo como inspiración el DSL de Sinatra. Se caracteriza por ser rápida y requerir una configuración mínima. El objetivo por el cual se ha creado esta framework es para evitar que el

usuario cree líneas de código innecesarias. Asimismo, al usar Hobbit podrás entender el uso de Rack y sus extensiones. (Informativa, 2019)

#### **2.2.4 Metodologías de Desarrollo de Software**

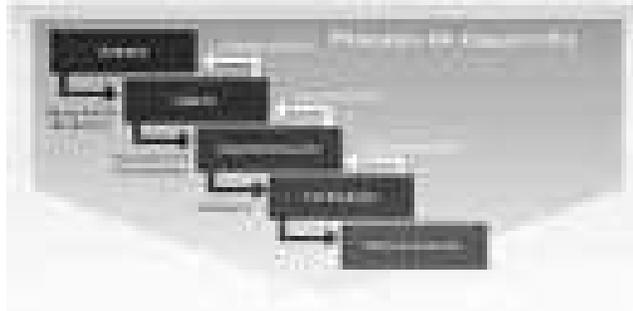
Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas. (EcuRed, Metodologías de desarrollo de Software, 2019)

##### **2.2.4.1 Modelos y enfoques de desarrollo**

Cada metodología de desarrollo de software tiene más o menos su propio enfoque para el desarrollo de software. Estos son los enfoques más generales, que se desarrollan en varias metodologías específicas. Estos enfoques son los siguientes:

- Modelo en cascada: Framework lineal.
- Prototipado: Framework iterativo.
- Incremental: Combinación de framework lineal e iterativo.
- Espiral: Combinación de framework lineal e iterativo.
- RAD: Rapid Application Development, framework iterativo.

## Modelo en Cascada

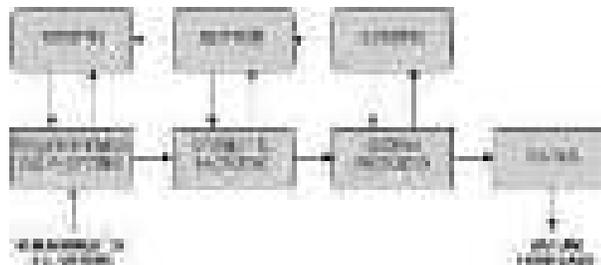


**Figura 58.** Modelo en Cascada

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>

Es un proceso secuencial, fácil de desarrollo en el que los pasos de desarrollo son vistos hacia abajo (como en una cascada de agua) a través de las fases de análisis de las necesidades, el diseño, implantación, pruebas (validación), la integración, y mantenimiento. (Wikipedia C. d., Metodología de desarrollo de software, 2019)

## Prototipo



**Figura 59.** Modelo Prototipo

<http://rquerrero334.blogspot.es/1192897080/>

El prototipo permite desarrollar modelos de aplicaciones de software que permiten ver la funcionalidad básica de la misma, sin necesariamente incluir toda la lógica o características del modelo terminado. (Wikipedia C. d., Metodología de desarrollo de software, 2019)

## Incremental

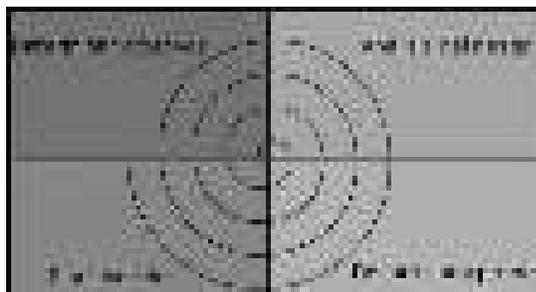


**Figura 60.** Modelo Incremental

<http://isw-udistrital.blogspot.com/2012/09/ingenieria-de-software-i.html>

Provee una estrategia para controlar la complejidad y los riesgos, desarrollando una parte del producto software reservando el resto de aspectos para el futuro. (Wikipedia C. d., Metodología de desarrollo de software, 2019)

## Espiral



**Figura 61.** Modelo en Espiral

[https://es.wikipedia.org/wiki/Desarrollo\\_en\\_espiral](https://es.wikipedia.org/wiki/Desarrollo_en_espiral)

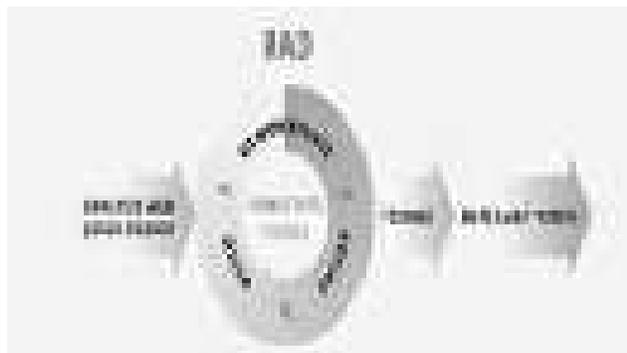
Los principios básicos son:

La atención se centra en la evaluación y reducción del riesgo del proyecto dividiendo el proyecto en segmentos más pequeños y proporcionar más facilidad de cambio durante el proceso de desarrollo, así como ofrecer la

oportunidad de evaluar los riesgos y con un peso de la consideración de la continuación del proyecto durante todo el ciclo de vida. (Wikipedia C. d., Metodología de desarrollo de software, 2019)

- Cada viaje alrededor de la espiral atraviesa cuatro cuadrantes básicos: (1) determinar objetivos, alternativas, y desencadenantes de la iteración; (2) Evaluar alternativas; Identificar y resolver los riesgos; (3) desarrollar y verificar los resultados de la iteración, y (4) plan de la próxima iteración. (Wikipedia C. d., Metodología de desarrollo de software, 2019)
- Cada ciclo comienza con la identificación de los interesados y sus condiciones de ganancia, y termina con la revisión y examinación. (Wikipedia C. d., Metodología de desarrollo de software, 2019)

### **Rapid Application Development (RAD)**



**Figura 62.** Rapid Application Development

<https://blog.capterra.com/what-is-rapid-application-development/>

El desarrollo rápido de aplicaciones (RAD) es una metodología de desarrollo de software, que implica el desarrollo iterativo y la construcción de prototipos. El desarrollo rápido de aplicaciones es un término originalmente utilizado para describir un proceso de desarrollo de software introducido por James

Martin en 1991. (Wikipedia C. d., Metodología de desarrollo de software, 2019)

#### **2.2.4.2 Tradicionales**

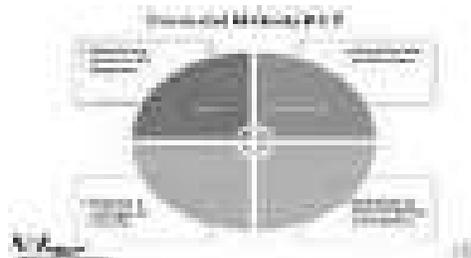
El enfoque tradicional de desarrollo de sistemas es estudiado ampliamente en medios académicos de la Ingeniería de software y es el denominado ciclo de vida de desarrollo de sistemas.

Bajo este enfoque:

Se realiza el Análisis de requerimientos, Diseño, Desarrollo e Implantación secuencialmente.

La fase siguiente no comienza hasta completar la anterior. (PMOinformatica.com, 2011)

#### **RUP**



**Figura 63. RUP**

<https://dtyoc.com/2016/06/07/modelo-rup-ibm/>

Es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de

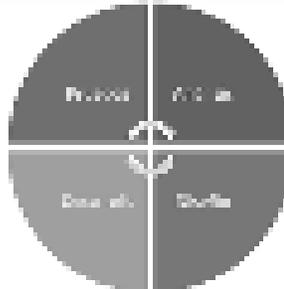
sistemas orientados a objetos. (Wikipedia C. d., Proceso Unificado de Rational, 2019)

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. También se conoce por este nombre al software, también desarrollado por Rational, que incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el Rational Method Composer (RMC), que permite la personalización de acuerdo con las necesidades. (Wikipedia C. d., Proceso Unificado de Rational, 2019)

#### **2.2.4.3 Ágiles**

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto organizado y multidisciplinario, inmerso en un proceso compartido de toma de decisiones a corto plazo. (Wikipedia C. d., Desarrollo ágil de software, 2019)

## XP



**Figura 64. XP**

<https://managementplaza.es/blog/sabes-como-funciona-xp/>

La metodología XP define cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance. El método especifica que de estas cuatro variables, tres de ellas podrán ser fijadas arbitrariamente por actores externos al grupo de desarrolladores (clientes y jefes de proyecto), y el valor de la restante deberá ser establecida por el equipo de desarrollo, quien establecerá su valor en función de las otras tres. (Grau)

## SCRUM



**Figura 65. SCRUM**

<http://www.diegocalvo.es/metodologia-scrum-metodologia-agil/>

Scrum es un marco de trabajo para desarrollo ágil de software.

Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo y obtener el mejor resultado posible de proyectos, caracterizado por optar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto. (Wikipedia C. d., Scrum (desarrollo de software), 2019)

- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados. (Wikipedia C. d., Scrum (desarrollo de software), 2019)
- Solapar las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o en cascada. (Wikipedia C. d., Scrum (desarrollo de software), 2019)

### **2.3 Fundamentación Legal**

De acuerdo (Registro Oficial Órgano del Gobierno del Ecuador, 2006) El estado reconoce, regula y garantiza la propiedad intelectual adquirida de conformidad con la ley, las decisiones de la Comisión de la Comunidad Andina y los convenios internacionales vigentes del Ecuador.

La propiedad intelectual comprende:

- Los derechos de autor y derechos conexos.
- La propiedad industrial que abarca, entre otros elementos los siguientes.
- Las invenciones
- Los dibujos y modelos industriales.
- Los esquemas de trazado (topografía) de circuitos integrados

- La información no divulgada y los secretos comerciales
- Las apariencias distintivas de los negocios y establecimientos de comercio
- Los nombres comerciales
- Las indagaciones geográficas
- Cualquier otra creación intelectual que se destine a un uso agrícola, industrial, o comercial.
- Art. 28 (Registro Oficial Órgano del Gobierno del Ecuador, 2006) `Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independiente de que hayan sido incorporadas en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el nombre (código fuente) o forma legible por maquina (código objeto) ya sea programas operativos y programas aplicativos, incluyendo diagramas de flujos, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.
- Art. 29 Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considera titular salvo prueba lo contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual. \_

## **2.4 Variables de la Investigación (Conceptualización)**

### **2.4.1 Variable Dependiente**

#### **Contabilidad de los productos existentes para la compra y venta.**

La Venta de Mercaderías es un tipo de transacción existente en la contabilidad que se basa en la salida o entrega por parte de una empresa en particular de bienes o servicios propios de la actividad que desarrolla y a un precio definido. (Gálan, 2015)

Según señala la propia definición de Mercaderías, estos productos o bienes fueron adquiridos por la compañía con la finalidad de emplearlos en transacciones de venta y, por tanto, integrarlos en su actividad o trabajo. A través de la venta de mercaderías la empresa desarrolla su negocio y recibe ingresos por su trabajo. (Gálan, 2015)

Existen diferentes formas a través de las cuales se realizan las ventas de mercaderías en la vida empresarial. Dependiendo de la forma y naturaleza de estas transacciones, y de su forma de contabilización, se pueden clasificar este tipo de ventas de mercaderías en Venta de mercaderías al contado, Venta de mercaderías con pago mediante entidad bancaria, y Venta de mercaderías con pago mediante efectos comerciales. (Gálan, 2015)

La diferencia existente entre la venta de mercaderías y la de otros activos en contabilidad se basa en el objeto protagonista de la venta en sí. La venta de mercaderías tal cual se centra en transacciones con elementos que no han sufrido una transformación o alteración en su composición o proceso de producción. (Gálan, 2015)

Existen también separaciones para elementos que sí han podido sufrir este cambio y son objeto de la venta o para productos considerados como

semiterminados, los cuales han sufrido cambios en su composición, pero no en su proceso productivo. (Gálan, 2015)

## **2.4.2 Variable Independiente**

### **Gestión de Control de Inventarios**

#### **Inventario**

La salsoteca Barreto no cuenta con un software de control de inventarios el mismo que debería formar parte del área de bodega del local, ya que llevar todo el registro de los licores que entran al local de manera manual podría ocasionar la pérdida de la información, todas sus características por cada uno de ellos según sea la marca, el tamaño, o clasificarlos por año de fabricación.

El inventario es el conjunto de mercancías o artículos que tiene la empresa para comerciar con aquellos, permitiendo la compra y venta o la fabricación primero antes de venderlos, en un periodo económico determinados. Deben aparecer en el grupo de activos circulantes.

Según (Guarango, 2015) `La base fundamental de toda empresa comercial es la compra y venta de bienes y productos terminados; de aquí la importancia del manejo de inventario por parte de la misma, es por eso que es de suma importancia conocer e implementar los mecanismos y técnicas adecuadas`.

El control de inventarios es una herramienta fundamental en la administración moderna, ya que esta permite a las empresas y organizaciones conocer las cantidades existente de productos disponibles para la venta, en un lugar y tiempo determinado, así como las condiciones de almacenamiento aplicables en las industrias. (EZPINOZA, 2013)

Según (Pinto F, 2017) `La base de datos que debe tener toda empresa basado en la compra y venta de diferentes artículos en los bienes y servicios es donde nace la necesidad de un control de inventario para el mejor desempeño de la misma en el manejo contable ya que este software facilitara el control oportuno y confiable en el final de cada periodo de la economía de la microempresa. \_

### **Inventario físico**



**Figura 66.** Inventario Físico

<http://www.cerouno.com.mx/administracion-del-activo-fijo/inventario-fisico/index.html>

De acuerdo (Juárez, 2007) Todas las entradas o salidas que se produzcan durante la realización del inventario deberán ser comunicadas a los auditores, facilitándoles todo tipo de documentación que acredite dichos movimientos. Se prestará especial atención a la posible duplicación o a la no consideración de partidas de tránsito, al comienzo del inventario físico deberán estar actualizadas todas las tarjetas de control de inventario permanente.

El inventario físico es el conteo manual de todas las existencias de una empresa. Esto, considerando los insumos, productos en proceso, bienes finales, activo fijo, entre otros.

Es decir, efectuar un inventario físico es similar a organizar un censo sobre la población de un país. Ambos, además, tienen como objetivo servir de base para la toma de decisiones. Cabe recordar que el inventario físico forma parte de la gestión de inventarios. (Westreicher, 2019)

### **Inventario mixto**

Según (Martínez, 2014) "Un inventario mixto se ve a partir de una clase de mercancía, cuya partida no se ha identificado o no puede identificarse con un lote en particular, entre sus características tenemos inventario de productos terminados, inventario en materia prima inventario en tránsito, inventario en proceso, inventario en consignación, inventario mínimo, inventario máximo, inventario disponible, inventario en línea, inventario agregado e inventario en cuarentena."

### **Inventario de consignación**

Según (Córdova, 2014) "se refiere a aquellos bienes o artículos que una persona o empresa (consignador), envía a otra persona física o moral, denominado comisionista, destinatario o consignatario, a fin de que venda dichos bienes a nombre y cuenta del primero y de acuerdo con las instrucciones hubieren acordado ambos

Son aquellas mercancías que un proveedor deja a un vendedor, pero el dueño sigue siendo el proveedor, sin tener que pagar en su totalidad el vendedor, es decir lo que se venda pagará y lo que no se venda será devuelto.

### **Inventario máximo**

Según (Aguirre, Mendoza Hernández, Robello, & Rivera, 2011) "El proceso de planeación de inventarios forma parte de un macro proceso de planeación

de la cadena de suministro y depende en gran medida de la estrategia de inventarios que se elija. \_

### **Inventario mínimo**

Según (Samuel, 2017) `El mínimo vendría a ser el stock de seguridad que debe tener una empresa de un artículo para poder seguir ofreciendo el servicio a sus clientes, de acuerdo con las previsiones y estadísticas que ha realizado la empresa. \_

### **Inventario de previsión**

Según (Reich, 2009) `El inventario de anticipación o inventario de previsión es utilizado por las empresas para absorber las irregularidades que se pueden presentar en la demanda o suministro que se presentará a futuro y que se encuentra, en muchas ocasiones, bien definida. \_

### **Inventario de reserva**

Según (Albizhita, 2015) `Los inventarios de reserva a veces son mantenidos en forma de artículos semiterminados para balancear los requerimientos de producción de los diferentes procesos o departamentos de que consta la producción para poder ajustar las programaciones de la producción y surtir a tiempo. \_

### **Inventario de mercaderías**

Según (Domingo, 2016) `Señala que el inventario de las mercaderías de las empresas debe ser gestionado y en ese marco planeado, organizado, controlado y retroalimentado, de tal forma que ayude a la gestión integral. \_

### **Inventario de fluctuación**

Según (Wendy, 2010) `Estos inventarios se llevan porque la cantidad y ritmo de las ventas y de producción no pueden pedirse con exactitud. Los inventarios de fluctuación llamados stocks de estabilización pueden incluirse en el plan de producción de manera que los niveles de producción no tengan que cambiar para enfrentar las variaciones aleatorias de la demanda. \_

### **Inventario periódico**

Según (Ortiz, 2014) `Realiza un control cada determinado tiempo o periodo, y para eso es necesario hacer un conteo físico. Para poder establecer con exactitud la cantidad de inventarios disponibles en una fecha determinada. La empresa solo puede saber tanto el inventario exacto como el costo de venta, en el momento de hacer un conteo físico, lo cual por lo general se hace al final de un periodo, que puedes ser mensual, semestral o anual. \_

### **Inventario por productos en procesos de fabricación**

Según (Betancur, 2013) `Consiste en la producción parcialmente manufacturada; y su costo comprende materiales, mano de obra y gastos indirectos de fabricación que les son aplicables. \_

### **Inventario de materias primas**

Según (Vázquez G. G., 2011) `Representan existencias de los insumos básicos de materiales que abran de incorporarse al proceso de fabricación de una compañía. Son aquellos materiales que son directamente utilizados en la fabricación del producto. Lo conforman todos los materiales con los que se elaboran los productos, pero que todavía no han recibido procesamiento.

### **Inventario de previsión**

Según (Vázquez G. G., 2011) Se tienen con el fin de cubrir una necesidad futura perfectamente definida. Se diferencia con el respecto a los de seguridad, en que los de previsión se tienen a la luz de una necesidad que se conoce con certeza razonable y por lo tanto, involucra un menor riesgo. \_

### **Costos en los inventarios**

Según (Flores & Parra, 2012) En cuanto a sus costos, puede decirse que los inventarios son uno de los activos más caros para muchas empresas, a veces llega a representar el 40% del capital total invertido. Por otra parte, más del 50% de los costos totales en las empresas industriales son costos de materiales, mientras que en las empresas comerciales o de ventas al detalle los costos de las mercancías o mercaderías giran en torno al 70% de los costos totales. \_

#### 1) Costo de Pedido

- ✓ Es independiente de la cantidad pedida
- ✓ Incluye los gastos administrativos
- ✓ Los costos de pedidos se formulan normalmente

#### 2) Costo de mantenimiento de un inventario

- ✓ Incluye el costo del espacio físico
- ✓ Roturas, deterioro y obsolescencia
- ✓ Oportunidad para una inversión alternativa

#### Costo de Escases

- ✓ Costos por no tener los bienes en el momento requerido

- ✓ Puede causar el aplazamiento de un pedido Se pierde el pedido cuando no se puede aplazar

#### 4) Costo de mantenimiento de un inventario

- ✓ Incluye el costo del espacio físico
- ✓ Roturas, deterioro y obsolescencia
- ✓ Oportunidad para una inversión alternativa

Estos costos de inventario no son más que piezas o recursos que a la vez son utilizados en unas empresas por medio de un sistema de Inventario ya que son los que cruzan los niveles de inventario y llegan a determinar a aquellos a mantener el momento que es necesario reabastecerlo.

#### **Manejo de inventario**

Según (Peza & Silva, 2015) `La gestión de sistemas de inventarios constituye una de las funciones más complejas de las organizaciones, ya que implica mantener existencias para protegerse contra incertidumbres al menor costo. \_

Básicamente el manejo de inventario es el que nos proporcionara una gran mejoría en la salsoteca Barreto en lo que respecta al servicio de los distribuidores y clientes, además nos permitirá tener un mejor control en todos las operaciones que se realicen a los productos que se ingrese a la bodega de la salsoteca, mejorando los registros con mayor eficiencia en el control de los licores para la venta al público.

#### **Métodos de valoración o valuación de inventario**

Según (López, 2016) `Los métodos de valoración o métodos de valuación de inventarios son técnicas utilizadas con el objetivo de seleccionar y aplicar una base específica para evaluar los inventarios en términos monetarios. La

valuación de inventarios es un proceso vital cuando los precios unitarios de adquisición han sido diferentes. \_

Estos métodos son las técnicas que se manejan en un negocio que tenga un control de inventario con el ánimo de seleccionar y aplicar una base que detalla que nos sirva para valorar los inventarios, ya que es un asunto importante especialmente cuando los valores o costos unitarios de compra son otros.

Existen muchas técnicas para valorar inventarios, pero los más utilizados por las diferentes empresas o microempresas son las siguientes:

- ✓ Identificación Especifica
- ✓ Promedio Ponderado o Costo promedio constante
- ✓ Últimos en entrar, Primeros en Salir UEPS

Primeros en entrar, Primeros en salir PEPS

### **Método de identificación específica**

Según (Guingue, 2014) Es una forma de valorar los inventarios de mercancías de la empresa, calculando el costo, determinando su nivel de utilidad y fijando la producción con su respectivo nivel de ventas. \_

Este método de identificación específica es la forma o manera de valorar los inventarios de la mercadería que llegan a las empresas, el cual calcula los costos que determinan su nivel de utilidad y a las fija la producción con su respectivo nivel de ventas al mercado en la sociedad.

Se puede decir que es el método más simple, pero es el que demanda mayor tiempo por el cual es el que determina el costo de los materiales utilizados y el costo del inventario final.

### **Promedio ponderado o costo promedio constante**

Según (Damaris, 2011) Se basa en el costo promedio del inventario durante el período, este método se utiliza para la valoración de un inventario en un negocio se grande o pequeño el cual toma valores promedios tanto para las mercaderías que se encuentran en stock, como también para los costos de mercaderías que se hayan vendidas.

### **Método de control de inventario**

Según (Jiménez, 2012) El control sobre los inventarios es importante porque las mercaderías son el alma de un comercializador. Las compañías exitosas cuidan mucho de proteger sus inventarios. \_

Este método se lo realiza con la finalidad de pronosticar un desarrollo de ventas o presupuesto, para llegar a determinar los costos de inventario, ventas o a la vez la obtención, recepción, almacenaje, producción etc.

La predisposición en lo que respecta al nivel de inventario es de que la empresa se mantenga en un nivel bajo, llegando a controlar y detener así la cantidad de dinero que deben comprometerse en os inventario.

Entre estos se encuentran dos métodos de control de inventario:

### **Control Operativo**

Según (Aguirre R. , Control Administrativo y Operativo, 2015) El control se refiere a los mecanismos utilizados para garantizar que conductas y desempeño se cumplan con las reglas y procedimientos de una empresa.

El término control tiene una connotación negativa para la mayoría de las personas, pues se le asocia con restricción, imposición, delimitación, vigilancia o manipulación. \_

## **Control preventivo**

Según (OVIEDO, 2015) indica que Según Neto (2008) se realiza con el fin de prevenir la ocurrencia de fallas, y mantener en un nivel determinado a los equipos en la organización, se conoce como control preventivo directo o periódico, sus actividades están controladas por el tiempo; se basa en la confiabilidad de los equipos. \_

Es aquella modalidad que se encarga de vigilar el operativo de los mismos que se fundamenta en recepciones verídicas ajustándose a las necesidades, impidiendo así aglomeraciones excesivas de stock.

a. Un propio control preventivo de inventarios debe manejar los siguientes elementos.

- ✓ Control Contable: Kardex o software
- ✓ Control Físico: Almacén
- ✓ Control de Nivel de Inversión: Índice de Rotación

b. Técnicas existentes de control

- ✓ Máximos y Mínimos
- ✓ Índice de Rotación
- ✓ Aplicación de criterio especialmente cuando las especulaciones entran en juego
- ✓ Control Presupuesto

El inventario es aquel que consiste en planificar y a la vez llegar a un control del volumen de flujo que vota una empresa con respecto a su producción o venta de mercadería adquirida, hasta llegar a la venta y compra por parte de sus distribuidores.

En toda empresa existen cuatro funciones las mismas que se detallan a continuación.

- ✓ Compras
- ✓ Producción
- ✓ Finanzas
- ✓ Ventas

### **Sistemas**

Según (ZOOOLCO, 2012) `Por inventario se define al registro total de los bienes y demás cosas pertenecientes a una persona o comunidad, hecho con orden y precisión. Por extensión, se denomina inventario a la comprobación y recuento, de las existencias físicas en sí mismas y/o con las teóricas documentadas.

### **Sistema conceptual o ideal**

Este sistema es un conjunto organizado de distintas definiciones, como símbolos u otros instrumentos del pensamiento, tales como:

- ✓ Las matemáticas
- ✓ Notas musicales
- ✓ Lógica Formal

### **Sistema real**

Según (Arévalo, 2015) `Cualquier sistema, el cual el tiempo de la salida es significativo. Esto generalmente es porque la entrada corresponde a algún Movimiento en el mundo físico, y la salida está relacionada con dicho (Wikipedia C. d., C Sharp, 2019)

## 2.5 Glosario de Términos

**Paradigma:** El concepto de paradigma es utilizado comúnmente como sinónimo de "ejemplo", también se puede decir que es hacer referencia en caso de algo que se toma como "modelo".

**Recursividad:** Es la forma en la cual se especifica un proceso basado en su propia definición.

**Ensamblador:** Se refiere a un tipo de programa informático que se encarga de traducir un fichero fuente escrito en un lenguaje ensamblador, a un fichero objeto que contiene código máquina.

**Declarativo:** Es una de las tres maneras en que se almacena la información en la memoria a largo plazo.

**Multi-paradigma:** Los paradigmas de programación son una forma de clasificar los lenguajes de programación en función de sus características. Los idiomas se pueden clasificar en múltiples paradigmas.

**Inmutabilidad:** La inmutabilidad es la cualidad de aquello que no cambia. Se dice que es inmutable lo que no está sometido a la condición temporal.

**Abstracciones:** Es una operación mental destinada a aislar conceptualmente una propiedad o función concreta de un objeto, y pensar qué es, ignorando otras propiedades del objeto en cuestión.

**Gestionar:** Es asumir y llevar a cabo las responsabilidades sobre un proceso, esto puede ser empresarial o personal, lo que incluye: La preocupación por la disposición de los recursos y estructuras necesarias para que tenga lugar.

**Asíncrono:** Hace referencia al suceso que no tiene lugar en total correspondencia temporal con otro suceso.

**Complejidad:** Es la cualidad de lo que está compuesto de diversos elementos interrelacionados.

**Consignación:** Es el acto de consignar, el acto de entregar a otra persona o agente el cargo, la custodia o el cuidado de cualquier material o bienes.

**Consignatario:** Persona, entidad o empresa a quien se consigna una mercancía; en especial la que en los puertos representa al armador de un buque para tratar los asuntos administrativos relacionados con su carga y pasaje.

**Stock:** Registro documental de los bienes y demás cosas pertenecientes a una persona, empresa o comunidad.

**Fluctuación:** Es la diferencia existente entre el contenido de los libros de inventarios menos la existencia física real de las mercancías o productos que se tienen dentro de un negocio o establecimiento en cuestión.

## **CAPITULO III METODOLOGIA**

### **3.1 PRESENTACIÓN DE LA EMPRESA**

La Salsoteca Barreto es un negocio de venta de licores en un ambiente musical, gracias a la dedicación y esfuerzo de sus propietarios, Bolivar Barreto Falcones, Lisette Barreto Pineda, y Otto Barreto, fue creada a mediados del año 2019, y está ubicada en la ciudad de Quevedo, parroquia San Camilo de la provincia de Los Ríos.

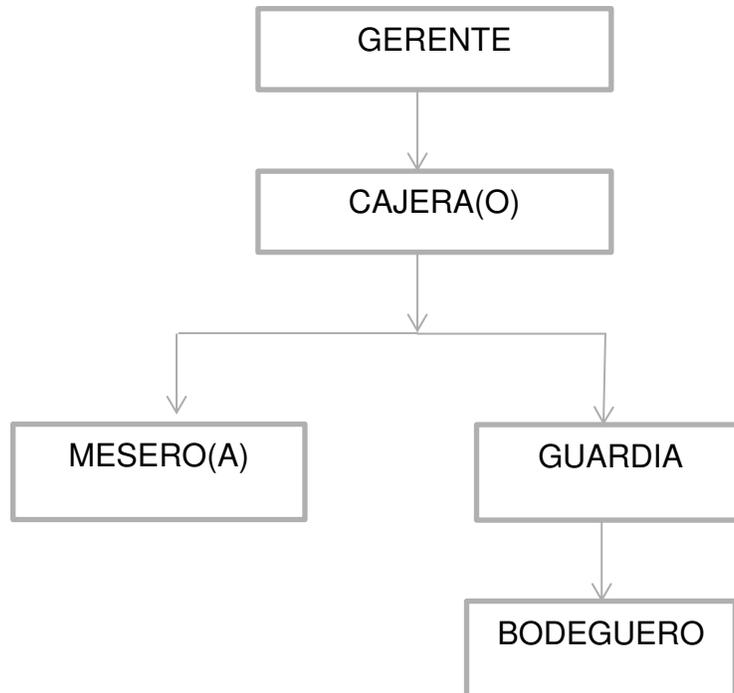
#### **Misión**

Ser una Salsoteca de alto nivel en la provincia de Los Ríos, y estar a la altura de las demás Salsotecas a nivel de la provincia.

#### **Visión**

Enfocar el negocio a futuro con diversas sucursales dentro y fuera de la provincia de Los Ríos.

## Estructura Organizativa



Por medio de esta estructura organizacional se muestra por nivel jerárquico los cargos actuales de la Salsoteca Barreto.

Elaborado por: Orlys Bryan Briones Gonzabay.

**Figura 67.** Organigrama de la empresa.

### Clientes

- Personas adultas a partir de los 18 años de edad.

### Productos y servicios.

**Servicios:** Venta de bebidas alcohólicas de distintas marcas, en un buen ambiente musical.

## **Descripción del proceso objeto de estudio**

La Salsoteca tiene como función principal la venta al público de licores exclusivamente dentro del establecimiento, para lo cual requiere de un buen manejo inventarios de sus productos que son adquiridos a sus proveedores y almacenados en la bodega del local, y así es como se mantiene abastecido los licores que son para la venta de nuestros clientes acorde a la demanda de la clientela.

### **3.2 Diseño de la investigación**

El presente proyecto de investigación se desarrollara en la Salsoteca Barreto, dicho diseño está enfocado en dos aspectos: enfoque cualitativo y enfoque cuantitativo.

#### **Enfoque cualitativo**

Según (Blasco y Pérez, 2007), señalan que la investigación cualitativa estudia la realidad en un contexto natural y cómo sucede, sacando e interpretando fenómenos de acuerdo con las personas implicadas.

Según (Lincoln y Denizn, 1994), la investigación cualitativa es un campo interdisciplinar, transdisciplinar y en muchas ocasiones contradisciplinar.

Atraviesa las humanidades, las ciencias sociales y físicas. La investigación cualitativa es muchas cosas al mismo tiempo. Es multiparadigmática en su enfoque.

Según (Sandín Esteban, 2003), la investigación cualitativa es una actividad sistemática orientada a la comprensión en profundidad de fenómenos educativos y sociales, a la transformación de prácticas y escenarios socio educativos y sociales, a la transformación de prácticas y escenarios

socioeducativos, a la toma de decisiones y también hacia el descubrimiento y desarrollo de un cuerpo organizado de conocimientos.

### **Enfoque cuantitativo**

Según (Tamayo, 2007), consiste en el contraste de teorías ya existentes a partir de una serie de hipótesis surgidas de la misma, siendo necesario obtener una muestra, ya sea en forma aleatoria o discriminada, pero representativa de una población o fenómeno objeto de estudio.

Según (Fernández, P. y Díaz, P. 2002), la investigación cuantitativa trata de determinar la fuerza de asociación o correlación entre variables, la generalización y objetivación de los resultados a través de una muestra para hacer inferencia a una población de la cual toda muestra procede.

Según (Hernández, 2006), en la investigación Cuantitativa Los estudios que utilizan este enfoque confían en la medición numérica, el conteo, y en uso de estadística para establecer indicadores exactos.

### **3.3 Tipos de investigación**

El presente proyecto de investigación tiene como objeto de estudio aplicar varios tipos de investigaciones.

#### **Investigación exploratoria**

La investigación exploratoria es aquella que se efectúa sobre un tema u objeto desconocido o poco estudiado, por lo que sus resultados constituyen una visión aproximada de dicho objeto, es decir, un nivel superficial de conocimientos. (Arias, 2006)

De modo tal, que constituye un hecho que la misma apuntará conocimientos ciertos o al menos aproximaciones, respecto a un tema en especial, debiendo este constituir en un conjunto de fundamentos que brindarán luces

o bien datos para las investigaciones posteriores, las cuales se encargarán de profundizar al respecto. (Arias, 2006)

La investigación se desarrolla en la Salsoteca Barreto, se encuentra ubicada en la provincia de Los Ríos, cantón Quevedo, parroquia San Camilo, para el desarrollo de esta investigación se realizó mediante encuestas a los propietarios del local y a los proveedores sobre los pedidos y con qué frecuencia realizan compras de los productos.

Una vez realizadas las encuestas se determinó el verdadero problema de la Salsoteca, para lo cual se tomaran algunos aspectos que permitirán llegar a tomar la decisión para el diseño del software planteado.

### **Investigación explicativa**

La investigación explicativa se encarga de buscar el porqué de los hechos mediante el establecimiento de relaciones causa-efecto. En este sentido, los estudios explicativos pueden ocuparse tanto de la determinación de las causas (investigación post facto), como de los efectos (investigación experimental), mediante la prueba de la hipótesis. Sus resultados y conclusiones constituyen el nivel más profundo de conocimientos. (Arias, El Proyecto de Investigación, 2006)

Para la mayoría de los expertos o bien de los dedicados a los estudios metodológicos, se considera que la investigación explicativa, es una de las más complejas de desarrollar, y así no los coloca de relieve Arias. (Arias, El Proyecto de Investigación, 2006)

### **Investigación correlacional**

Son aquellos trabajos donde nuestra preocupación se centra en determinar los orígenes o causas de un determinado conjunto de fenómenos. Su

objetivo, por lo tanto, es conocer por qué suceden ciertos hechos, analizando las relaciones causales existentes o, al menos, las condiciones en que ellos se producen. (Sabino, 2014)

Lo que busca es entablar el porqué de los fenómenos, es decir, busca descubrir porque estos suceden o se originan. O lo que es igual, este proceso de investigación tiene a lugar, estudiar el fenómeno, pero en un aspecto muy concreto, cual es el conjunto de circunstancias, hechos o demás fenómenos que lo hacen existente. (Sabino, 2014)

### **Técnicas de investigación**

Para este proyecto de investigación se aplicara la técnica de encuestas a los propietarios y trabajadores del local, la cual sirvió para determinar las necesidades tecnológicas que requiere el local.

### **Instrumentos de investigación**

En el presente proyecto de investigación para obtener información real y clara se ha utilizado una encuesta dirigida a los propietarios de la Salsoteca y del personal que labora.

### **3.3 La Encuesta**

Este proceso es importante debido a que esta adecuado al proposito del presente proyecto de investigacion, lo cual nos permitirá ver los productos que tienen en stock en la bodega de la Salsoteca.

Se ha creado una encuesta a los propietarios y trabajadores del local para poder saber y obtener información de su manera de llevar los registros de los productos en la bodega del local.

La encuesta permitirá conocer detalladamente el registro que llevan el personal que labora en la bodega del local, conocer sus productos como son clasificados una vez que son adquiridos a los mayoristas.

Esta propuesta tiene su fundamento en el estudio de las preguntas realizadas a los propietarios y empleados de la Salsoteca.

### 3.4 Solución Propuesta

Una vez realizado el análisis de la problemática, se procederá a diseñar un software acorde a las necesidades de los propietarios y empleados del local, el mismo que será desarrollado en Visual Studio bajo una plataforma de base de datos en SQL Server, la cual permitirá una solides en la manipulación de la información. A continuación, se detalla la solución propuesta.

PROBLEMA	SOLUCIÓN
No existe un software de control de inventarios	Se diseñara un software en Visual Studio que reciba la información de la base de datos.
La Salsoteca tiene un registro de productos manual (no automatizado)	Se diseñara una base de datos con la información de los productos clasificándolos por la marca.
No registran todos los productos que llegan a la bodega	Diseñar una base de datos dentro de la bodega, para así todo producto que llega sea registrado.

### 3.5 Estudio de Factibilidad

#### Factibilidad Técnica

- La Salsoteca Barreto no cuenta en la actualidad con ningún tipo de software.

- La falta de capacitación al personal impide que puedan manejar cualquier tipo de software informático.

### **Factibilidad Económica**

- Para el diseño de este proyecto de investigación se determinó el presupuesto para diseñar el software.
- No habrán costos adicionales de instalaciones eléctricas, o algún tipo de bien inmueble, ya que el local si cuenta con el equipamiento físico y tecnológico necesario.

### **Factibilidad Operacional**

- Al momento de ser concluido el proceso de diseño del software de control de inventarios, se propone implementarlo a futuro si así lo desean los propietarios.

## **CONCLUSION DEL ESTUDIO DE FACTIBILIDAD**

Se concluye que, aunque la Salsoteca no cuenta, con ciertos instrumentos tecnológicos, el sistema propuesto es factible en el ámbito financiero, técnico y operacional.

### **3.6 Características Relevantes**

El software contará con un diseño de interfaz gráfica muy amigable y de muy fácil manejo para los empleados y propietarios del local, lo cual será un beneficio adicional ya que la mayoría de los programas del mercado ofrecen interfaces pobres y sin ningún tipo de contenido.

## **Ventajas**

Dentro de las ventajas del presente proyecto podemos destacar las siguientes:

- Mayor seguridad en el manejo de información de los productos.
- Ahorro de tiempo y dinero.
- Facilidad de manejo del software.

## **Control y políticas del proyecto**

Todo software de inventario consistirá en el diseño de herramientas informáticas para diseñar bases de datos y sistemas de información de los procesos de registros de inventarios que se realizan en el local.

- El diseño de los softwares de inventarios contratados serán realizados únicamente por Orlys Bryan Briones Gonzabay.
- Los tiempos serán calculados con base a un análisis preliminar de requerimientos, los cuales dependerán de la magnitud del diseño solicitado.
- Los avances en el diseño del software estarán sujetos a las cargas de trabajo de las áreas involucradas.
- Al comenzar la etapa de diseño, los propietarios de la Salsoteca deberá cancelar el 60% del valor total del proyecto, esto para cubrir gastos operativos y administrativos que se generan al dar inicio un proyecto nuevo.

## **Entregables**

### **Del estudio de viabilidad**

- Descripción breve del software propuesto y sus características.

- Descripción breve de las necesidades del negocio en el software propuesto.
- Propuesta de organización del diseño.
- Estudio de los costes.
- Estudio de los beneficios que producirá el software.

### **De la fase de análisis**

- Captura de requisitos
  - Análisis del proceso manual que se automatizara.
  - Requisitos de los propietarios para diseñar el software.
  - Descripción del software propuesto.
- Especificación del software
  - Requisitos de datos.
  - Requisitos de hardware.

### **De la fase de diseño**

- Programas, módulos reutilizables y objetos.
- Ficheros y bases de datos.
- Diccionario de datos
- Procedimientos
  - Interfaces, tanto humanos como de máquinas
- Documentos del diseño final del sistema y de cada programa.
- Diagramas definitivos del sistema y de los programas.
- Descripción detallada de la lógica de cada programa.
- Descripción de las Entradas y Salidas (ficheros, pantallas, listados, etc.).

### **3.7 Beneficios**

El diseño de este software tiene varios beneficios, los cuales se detallan a continuación:

**Elevar el nivel de calidad de registros de datos de los productos:**

Reducir el nivel de pérdida de información, o datos de productos que no sean registrados al llegar a la bodega del local, así evitar el desabastecimiento de productos.

**Mejorar el uso de información de los productos en el software**

**propuesto:** Al momento de realizar ventas de productos dentro del local, se busca capacitar a las personas que utilicen el software para que puedan llevar un buen control de mercadería y evitar la escases.

## **CAPITULO IV**

### **DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA**

#### **4.1 Descripción Del Sistema Propuesto**

El presente proyecto está basado en el diseño de un software de escritorio para el control de inventarios para los licores de compra y venta en la Salsoteca Barreto, utilizando como lenguaje de programación Visual Studio 2017, IDE para programar como Visual Studio con un framework 4.7.

La interfaz a utilizar para el almacenamiento de bases de datos será en, SQL Server Management Studio.

La arquitectura a utilizar será cliente-servidor, debido que será la interacción entre el usuario y el sistema, donde el sistema mostrara como llevar el control de los datos, mostrara la cantidad de productos que dispone el local, como a su vez, mostrara los que ya han sido vendidos.

El sistema podrá registrar los datos de los productos que son adquiridos a los proveedores.

De dicha manera, el administrador del sistema podrá gestionar los distintos menús de opciones del sistema y comprobar información como la siguiente:

- Detectar que licores son los más indispensables, es decir, los que más ventas tienen.
- Detectar que licores están próximos a su fecha de caducidad o ya han caducado.
- Detectar con exactitud los licores existentes.

#### **4.1.1 Alcances Del Proyecto**

El diseño del software incluirá el ingreso de los licores que llegan a la bodega del local, serán registrados por marca de producto y año de fabricación.

Se diseñara una interfaz gráfica de acceso al software, compuesta por un usuario y contraseña, la cual nos enlazara con otra interfaz gráfica donde podremos seleccionar el nuevo registro de los licores que llegan a la bodega de local, o eliminar los licores que salen de la bodega del local.

Sistema de ingreso de productos

- Ingreso de usuario y contraseña
- Ingreso de nuevos productos que llegan al local
- Ingreso de cada producto por marca y año de fabricación.

Sistema de salida de productos

- Ingreso a la opción de eliminación de cada producto que sale de la bodega del local al expendio al público.
- Ingreso a la opción de eliminación de cada producto que se vende dentro del local.

#### **4.2 Requerimientos Del Software**

Los requerimientos de software son las condiciones o capacidades que debe cumplir el software para el control de inventarios de los licores de compra y venta, de la Salsoteca Barreto.

##### **4.2.1 Requerimientos Funcionales Del Sistema**

- **< Vista De Inicio >**
  - **Visualizar La Pantalla Inicio de Sesión:** Se podrá visualizar el inicio de sesión al sistema, estará compuesto por la dirección de correo electrónico, y la contraseña como mínimo de 8 caracteres,

en caso de no tener una cuenta se procederá a crear una nueva en la opción `Registrarse`.

- **< Vista De Ventana De Opciones >**
  - **Visualizar Todas Las Opciones Del Software:** Se podrá visualizar tres opciones: La primera será el ingreso a la opción de los licores que se encuentran al momento en la bodega del local, la segunda opción será el ingreso a la opción de los licores que son adquiridos a los proveedores para su posterior almacenamiento en la bodega, y la tercer ventana será el registro de los licores que se vendan dentro del local.
- **< Vista De Ventana De Ingreso De Licores En Bodega>**
  - **Visualizar Todos Datos De Los Productos:** Se podrá visualizar datos como la marca de cada producto, su fecha de elaboración, fecha de caducidad, y la cantidad de dicho producto que se encuentran al momento disponibles en la bodega.
- **< Vista De Ventana De La Cantidad De Productos Que Venden Los Proveedores Al Local>**
  - **Visualizar Todos Los Datos Que Dan Los Proveedores De Los Productos:** Se podrá visualizar la cantidad de productos que el propietario compra al proveedor, la marca de cada uno, fechas de elaboración y caducidad, y el valor económico que se le cancela al proveedor.
- **< Vista De Ventana De Licores Que Se Venden Dentro Del Local >**

**Visualizar Las Cantidades De Productos Que Se Venden Los Días Que Atiende El Local:** Se podrá visualizar la cantidad de productos que se venden al público, los días laborables del local, y así, saber el déficit de

productos que aún quedan guardados en la bodega, para su posterior reabastecimiento.

#### **4.2.2 Requerimientos No Funcionales**

Los requerimientos no funcionales (en lo adelante RNF) son propiedades o cualidades que el producto debe tener. Representan las características del producto a desarrollar.

##### **RNF De Software**

La computadora debe tener sistema operativo Windows 7 en adelante, para el diseño del software de control de inventarios.

##### **RFN De Hardware**

- Intel 2955U 1.4 GHz.
- 2 GB de RAM.
- 500 GB memoria interna
- Pantalla 14.0 HD LED LCD

##### **RFN De Diseño De Software**

- Lenguaje de programación Visual Studio 2015.
- Sistema Gestor De Base De Datos: SQL Server.

##### **RFN De Pantallas De Usuario**

- El diseño de las pantallas debe ser como sugiere el usuario.
- El software proporcionara claridad y correcta organización de los productos que se compran y venden en el local.
- Diseñar pantallas sencillas, de fácil interacción entre el usuario y el sistema.

### **RNF De Confiabilidad y Seguridad**

- Restringir el acceso a usuarios no autorizados, para que no puedan realizar ningún tipo de modificación o alteración de datos de los productos del local.
- Seguridad de acceso y administración de usuarios: Los niveles de accesos están determinados por usuario y contraseña, solo tendrá acceso el personal que esté autorizado y se hayan asignado sus respectivos usuarios para que pueda acceder al sistema.

Garantizar y dar fe que la información que sea alterada, cambiada o editada sea únicamente por el personal que esté autorizado y posea permisos para aquello.

### **RNF De Usabilidad**

- El software de control de inventarios solo podrá ser usado por personal que tenga su cuenta de usuario y contraseña, y posea conocimientos sobre el uso del software y las funciones que realiza.
- El software debe tener acceso al menú principal desde cualquiera de sus pantallas.
- Los elementos gráficos como los botones de opciones deberán señalar al que se refiere.

### **RNF De Portabilidad**

- El software de control de inventario solo puede ser usado en el local, bajo ningún motivo está autorizado instalar el software en ordenadores ajenos al negocio.

### **RNF De Funcionabilidad**

- Reducir al mínimo el tiempo en que se carga el software en iniciarse.

#### **RNF De Rendimiento**

- Tiempo de respuestas rápidas al igual que la velocidad de procesamiento de la información.

#### **RNF Legales, Derechos De Autor, Otros**

- Una vez terminado el producto, el software debe ser sometido a una evaluación y certificación por parte del cliente del mismo.

#### **RNF De Disponibilidad**

Los usuarios autorizados deben tener acceso al software en todo momento a la información solicitada.



### 4.3 Diagrama de Caso de Uso General

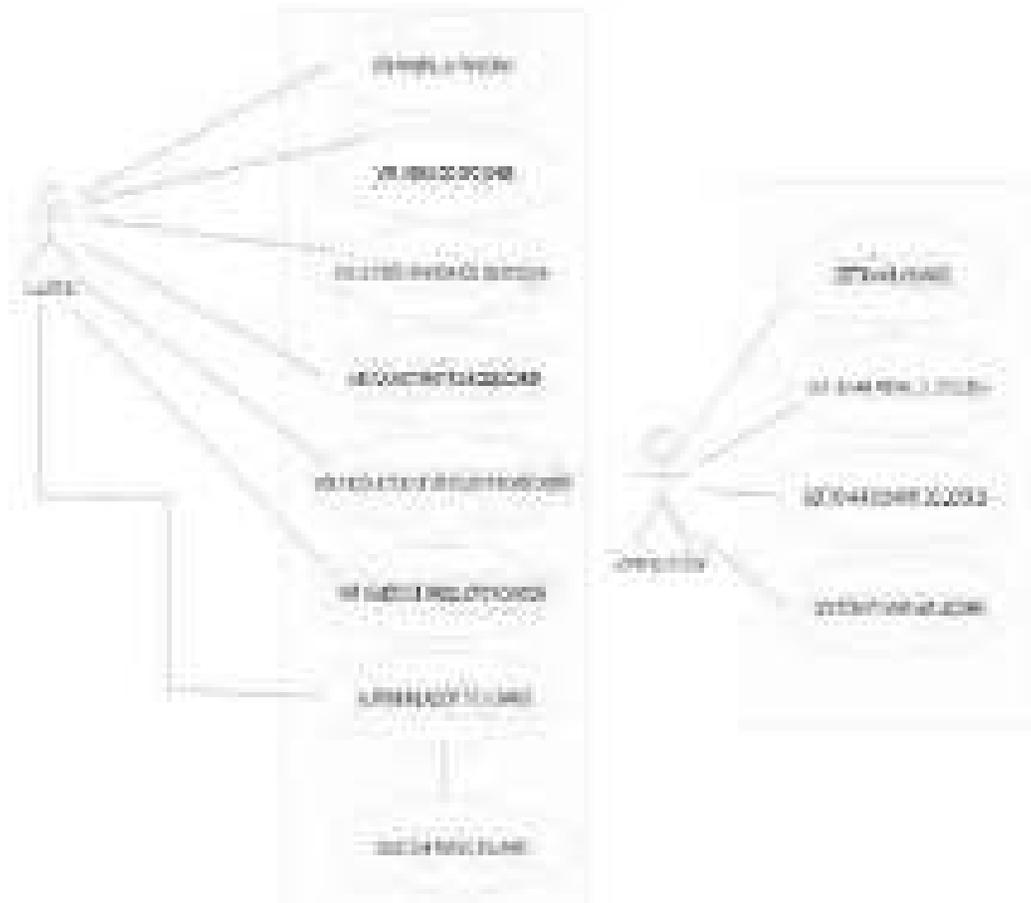


Figura 68. Diagrama de Caso de Uso General

#### 4.4 Diagrama de Actividades de Caso de Uso

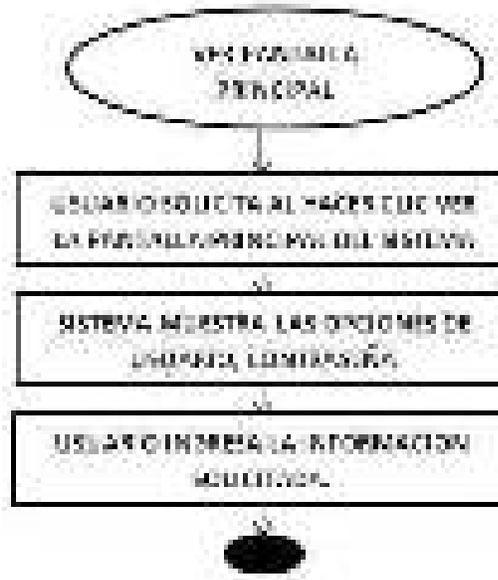


Figura 69. Pantalla Principal

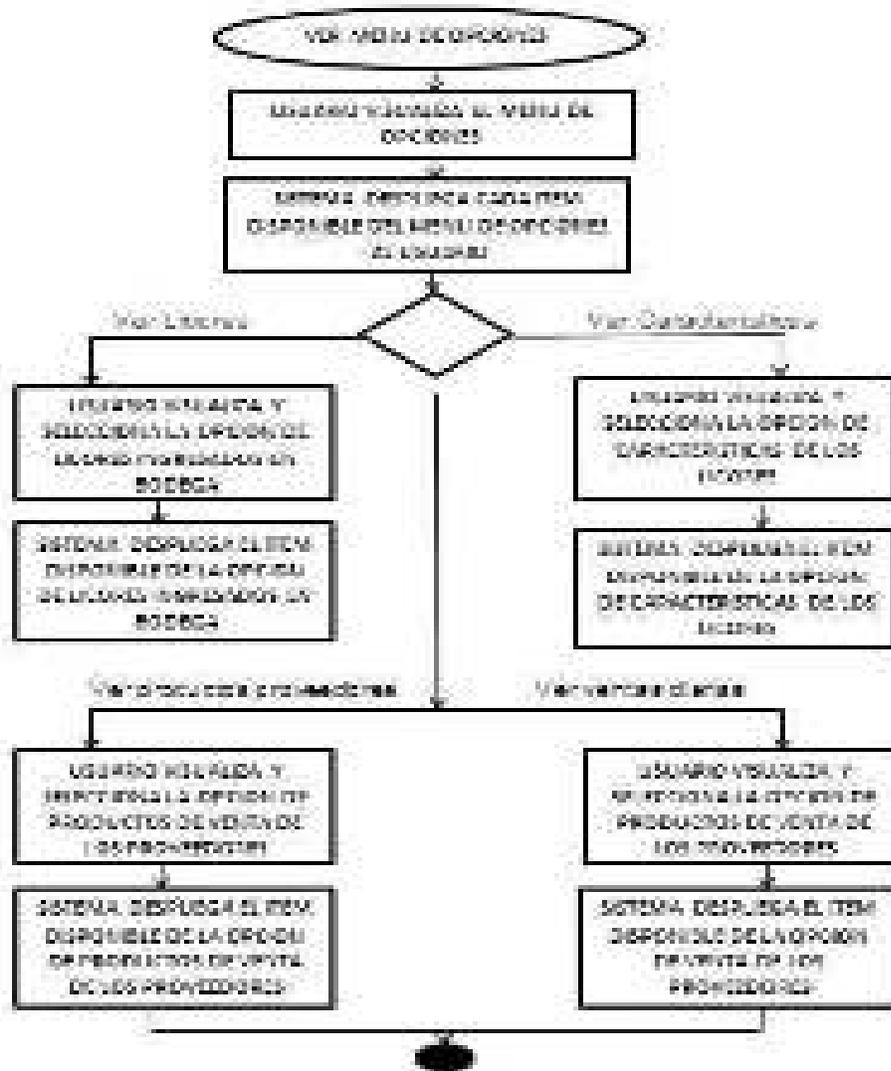


Figura 70. Menú de Opciones.

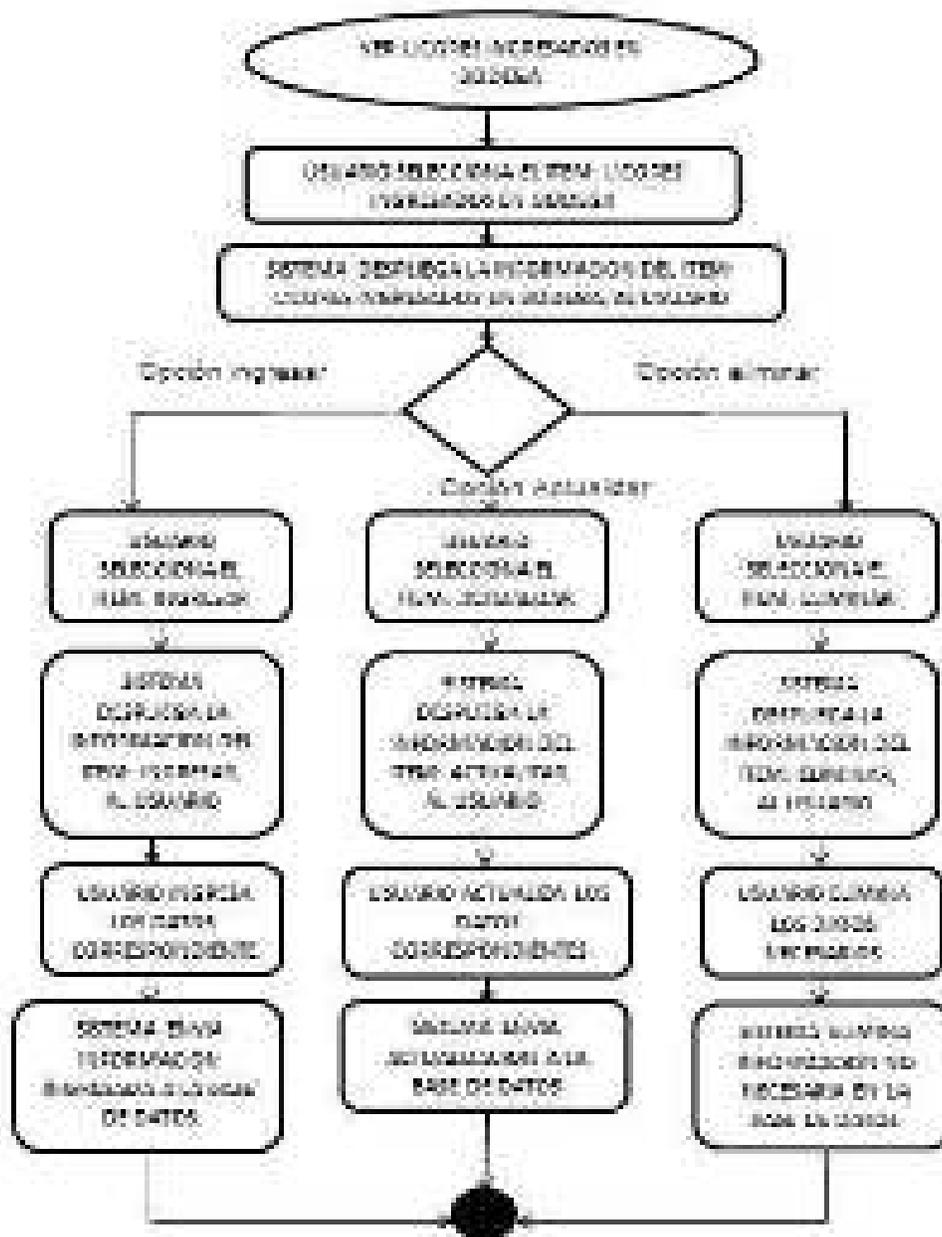


FIGURA 71. Licores ingresados en bodega.

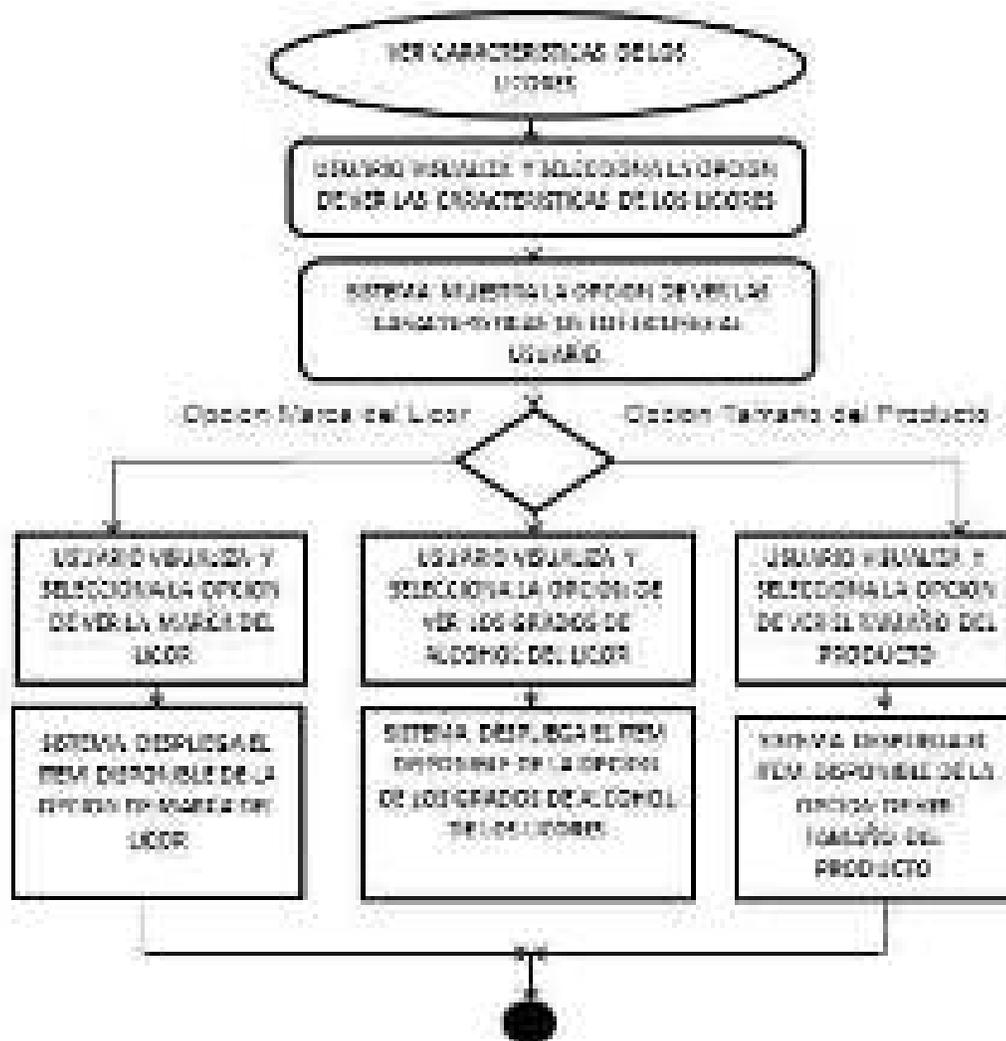


FIGURA 72. Características de los licores.

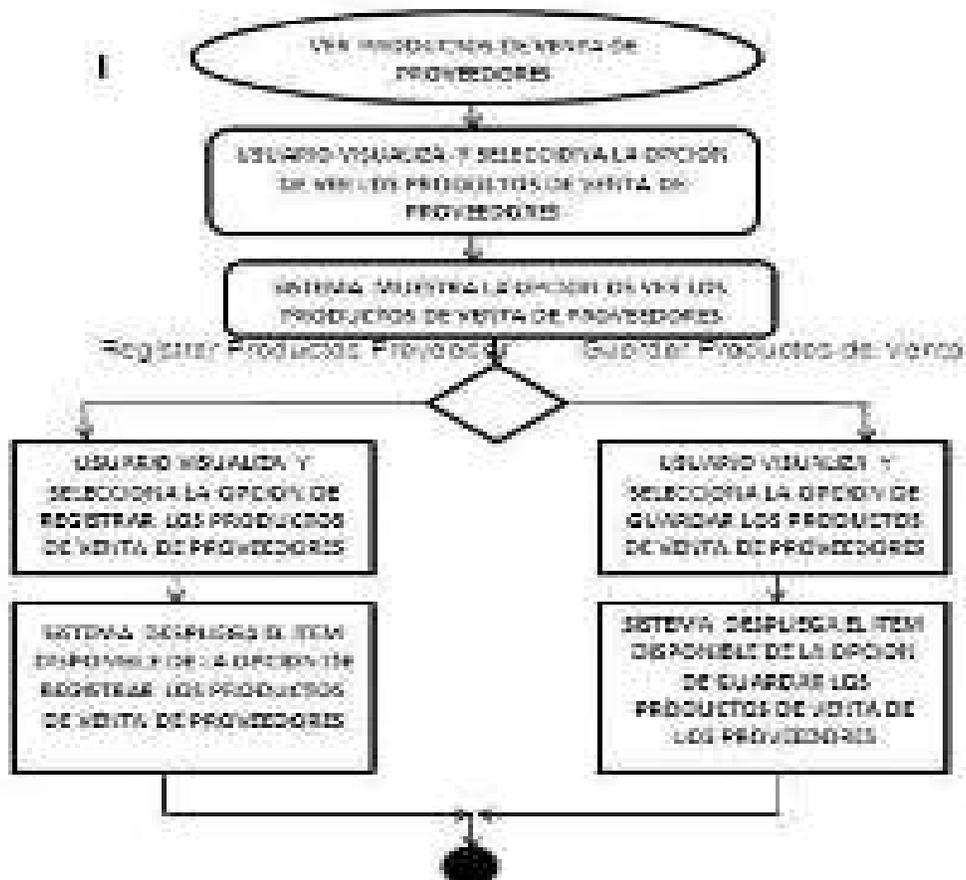


FIGURA 73. Productos de venta de proveedores.

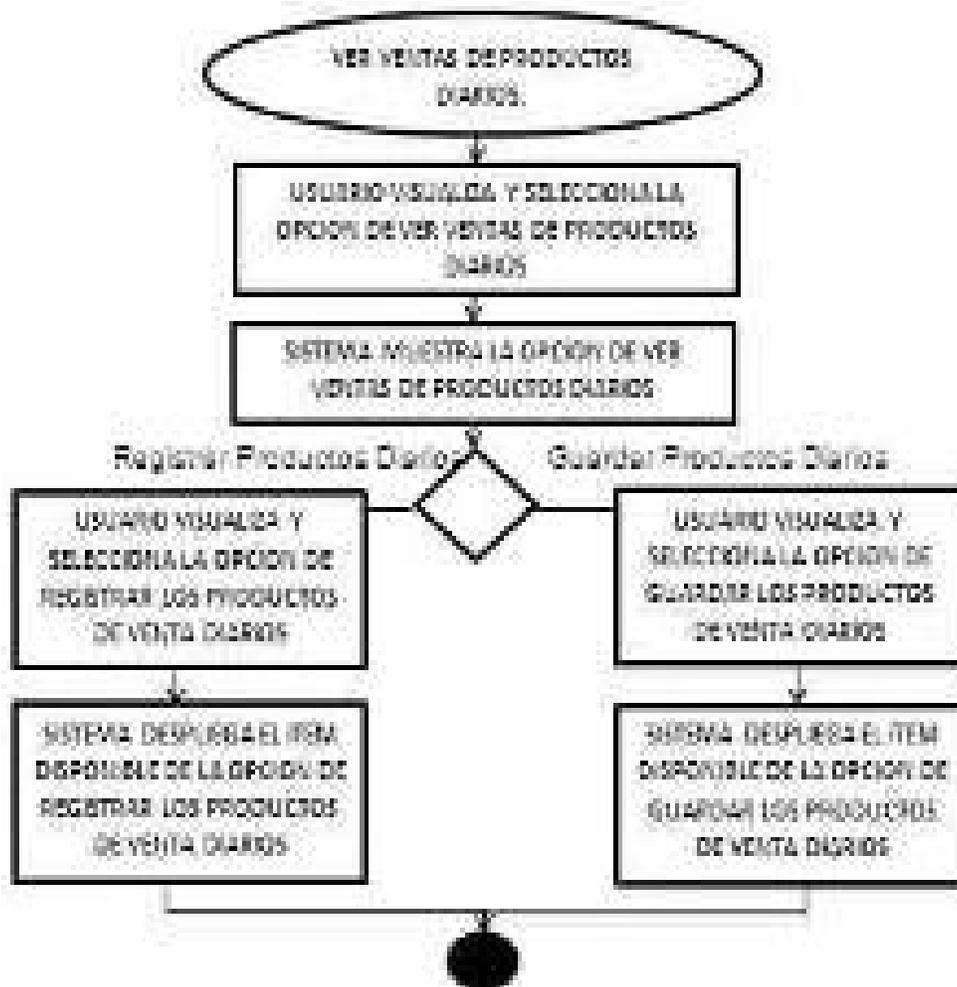


FIGURA 74. Ventas de productos diarios.

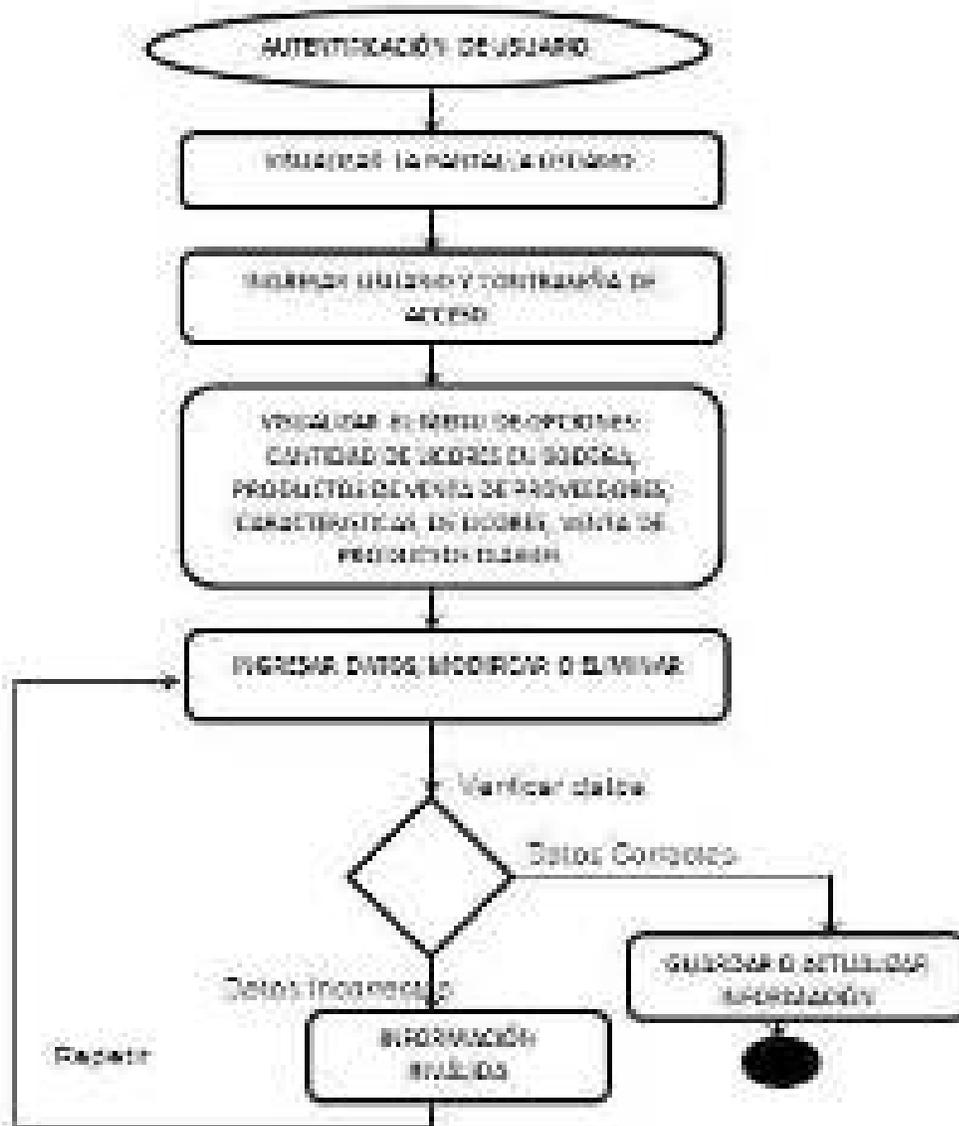


Figura 75. Autenticación de Usuario.

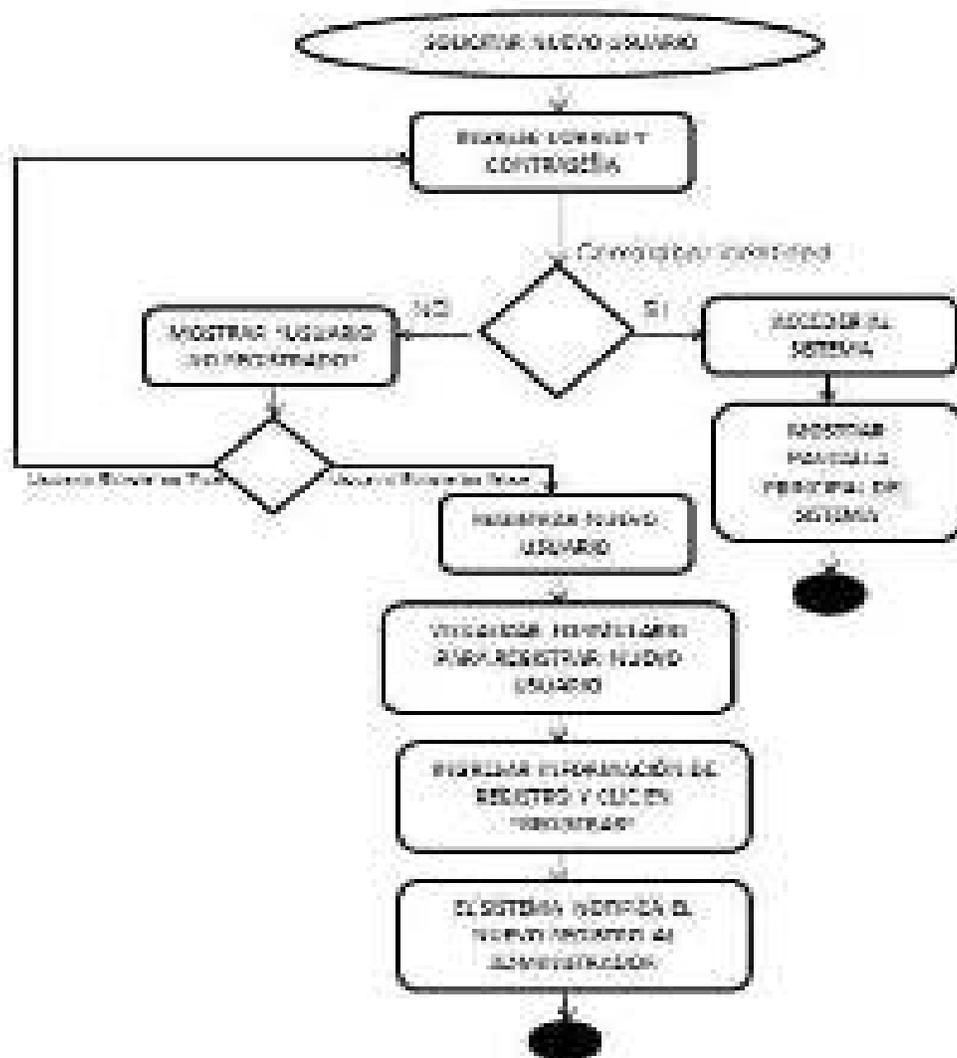


Figura 76. Solicitar nuevo usuario.

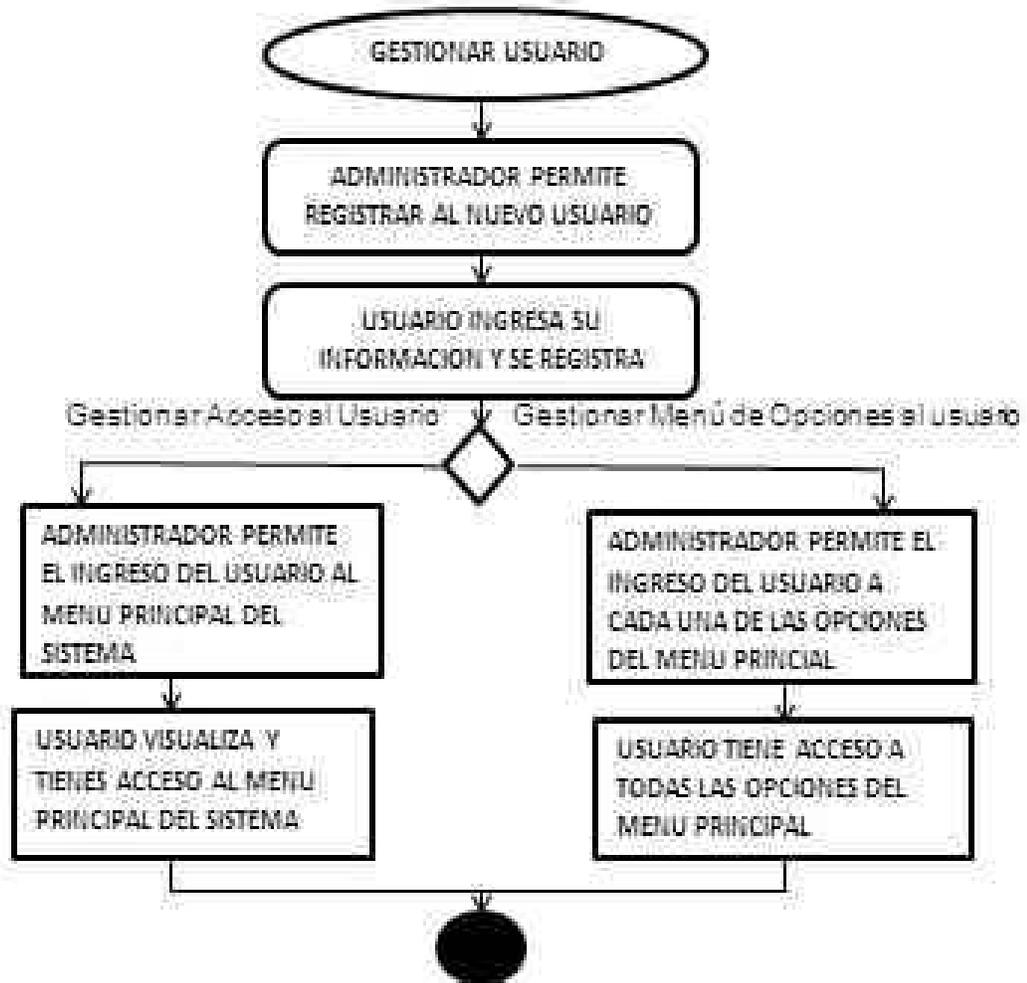


Figura 77. Gestionar Usuario

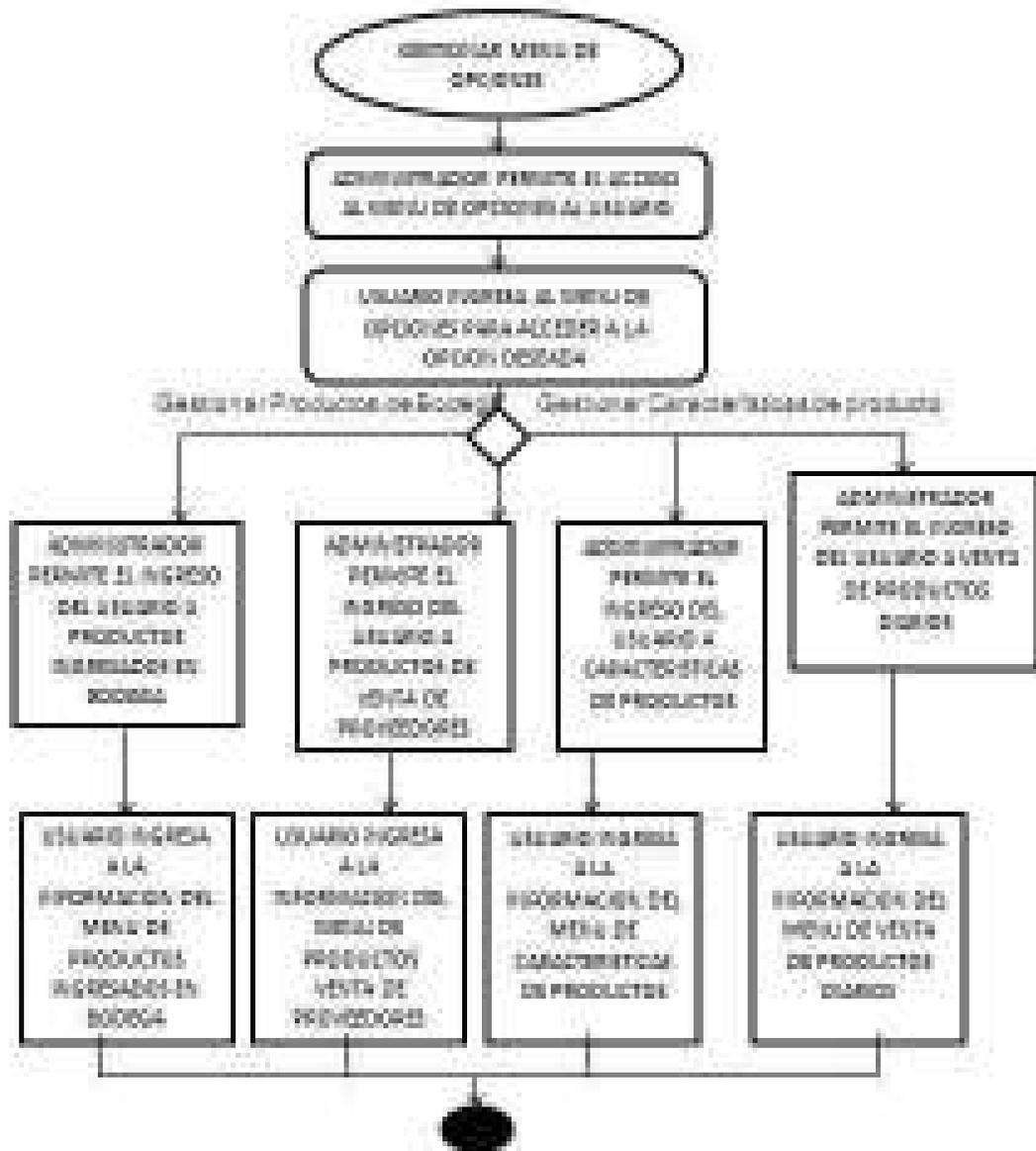


Figura 78. Gestionar Menú de Opciones

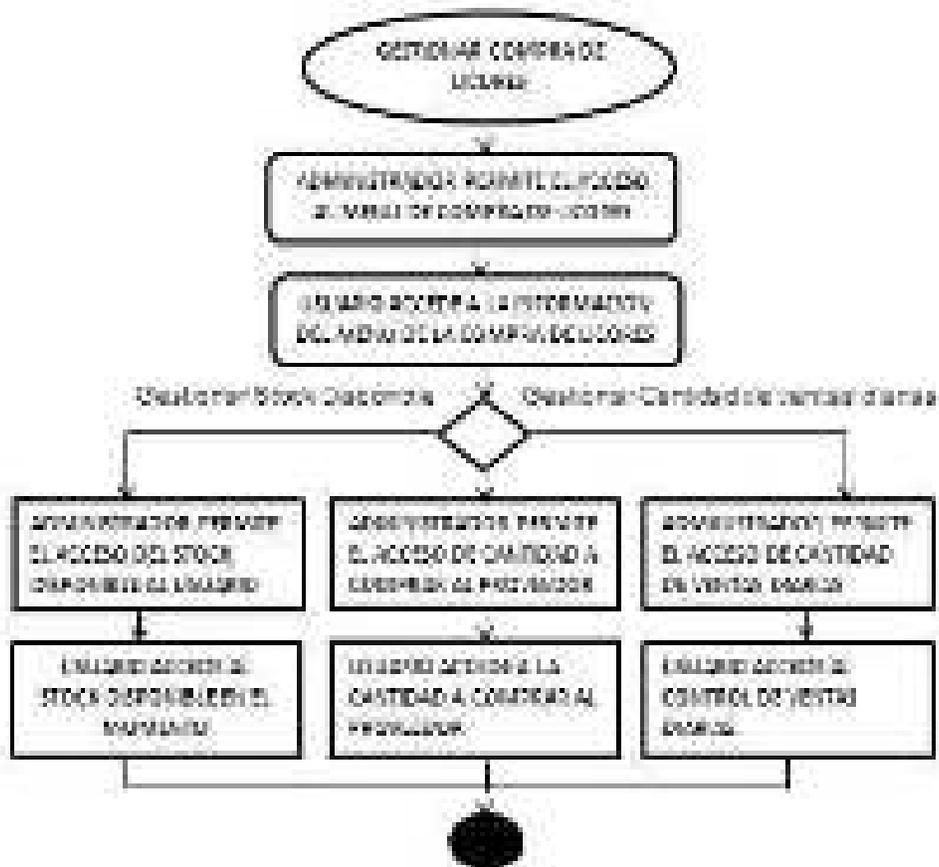


Figura 79. Gestionar compra de licores

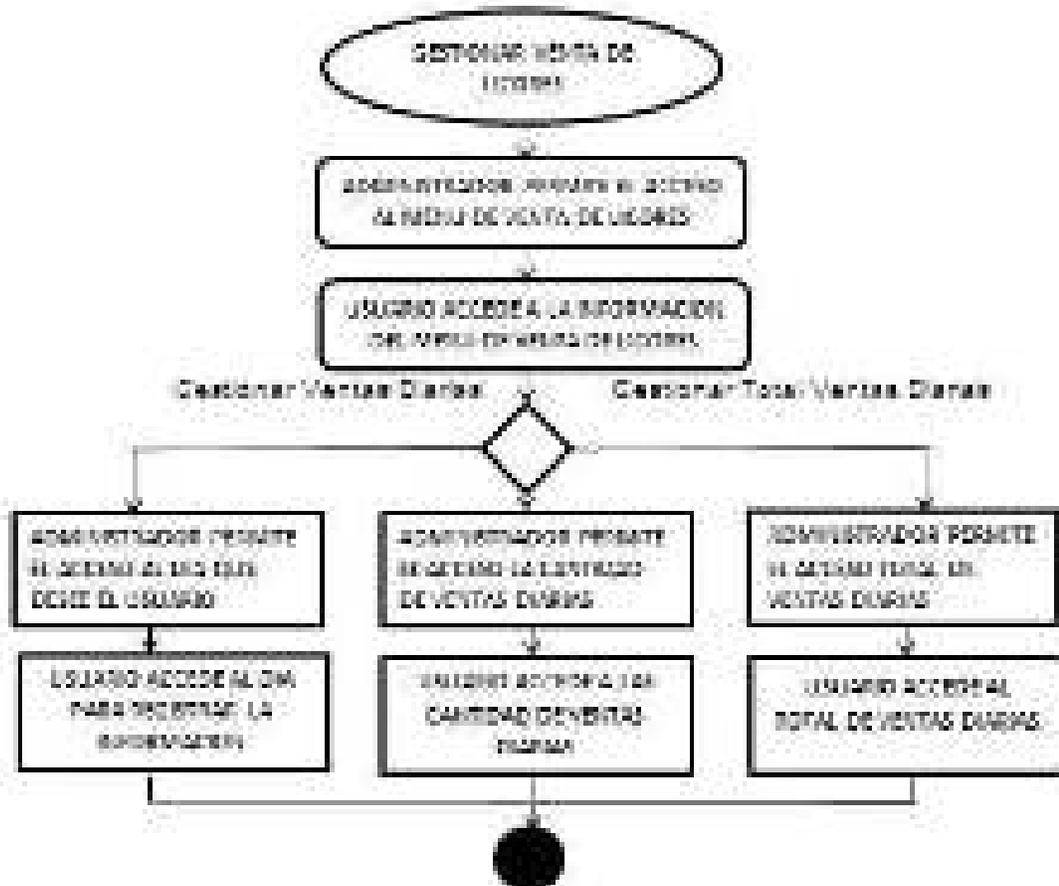
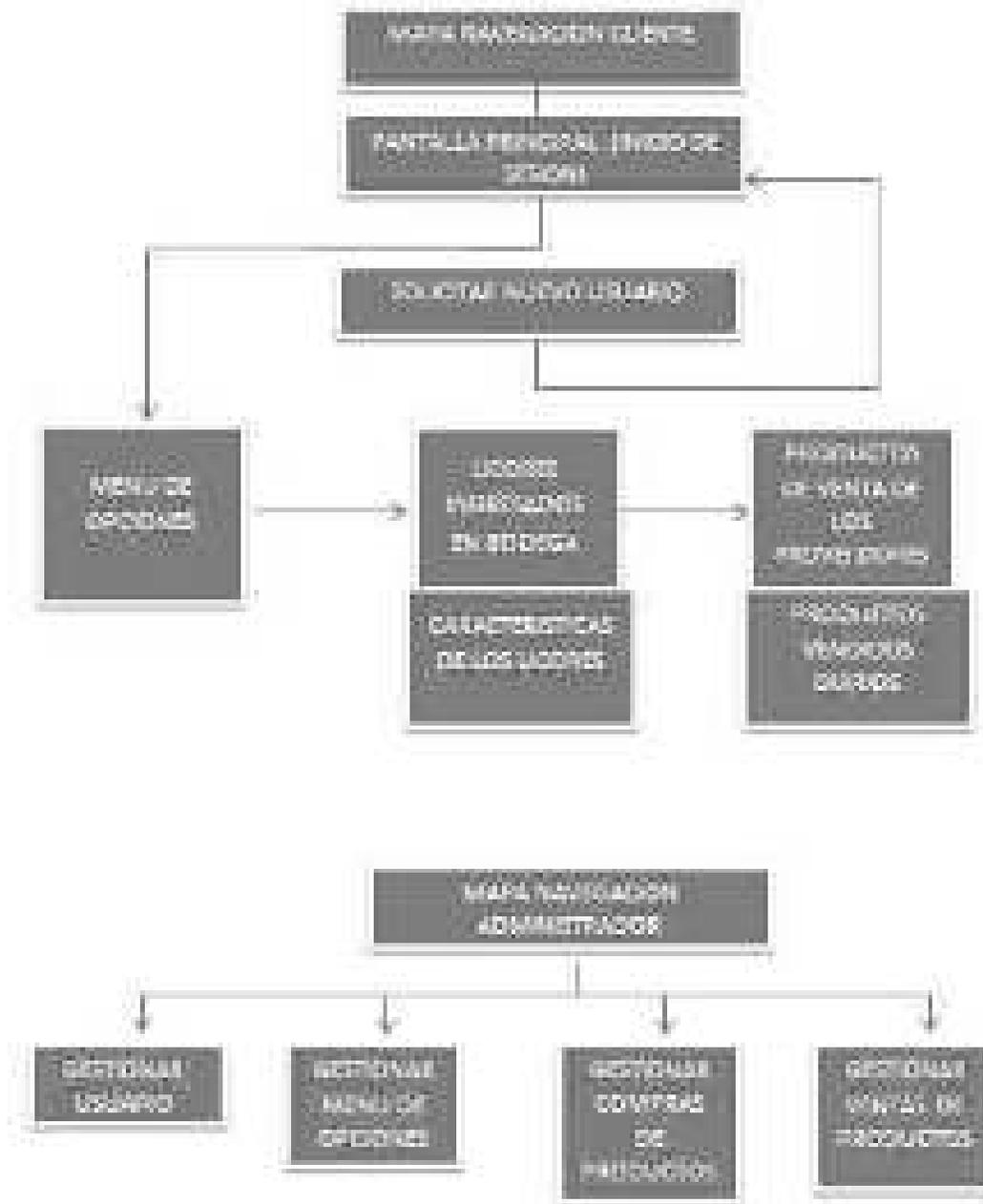


Figura 80. Gestionar venta de licores



#### 4.5 Mapa de navegación



## 4.6 Diseño de prototipado de la propuesta

### 4.6.1 Prototipado Pantallas De Usuario

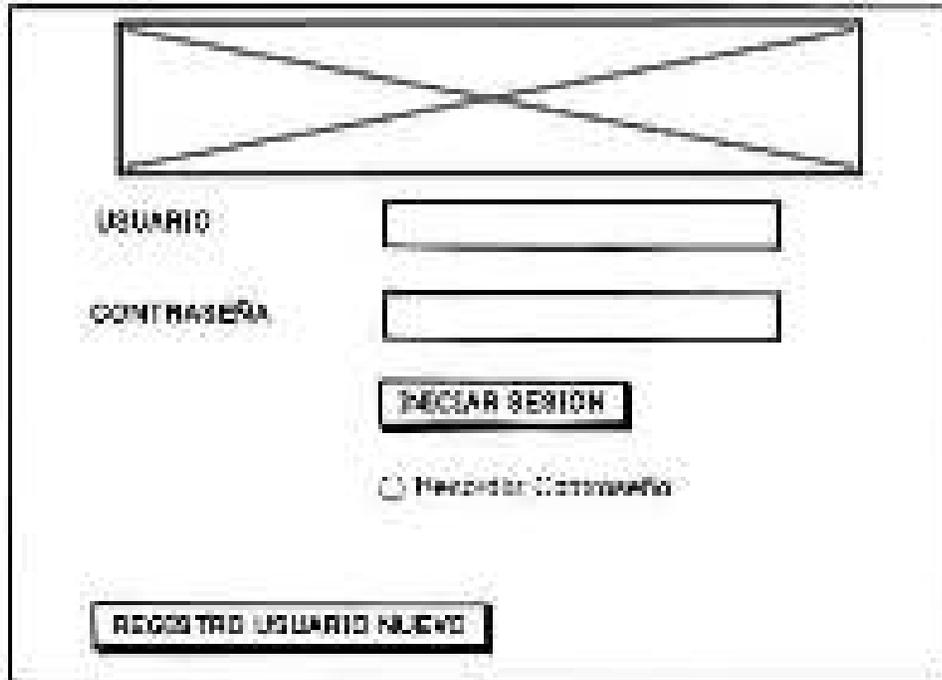


Diagrama de un prototipo de pantalla de inicio de sesión. El diseño incluye:

- Un recuadro rectangular con una 'X' diagonal que indica un espacio reservado para una imagen o logo.
- Etiquetas para los campos de texto: "USUARIO" y "CONTRASEÑA".
- Campos de entrada de texto rectangulares.
- Un botón rectangular con el texto "INICIAR SESION".
- Un enlace con un ícono de flecha y el texto "Recuperar Contraseña".
- Un botón rectangular con el texto "REGISTRO USUARIO NUEVO" en la parte inferior.

Figura 81. Pantalla Inicio de Sesión.



Apellidos

Nombres

Correo Electrónico

Contraseña

**REGISTRAR** **CANCELAR**

Figura 82. Pantalla Registro Nuevo Usuario.

**PRODUCTOS INGRESADOS A BODEGA**

**PRODUCTOS DE VENTA DE PROVEEDORES**

**CARACTERISTICAS DE PRODUCTOS**

**VENTA PRODUCTOS DIARIOS**

← ATRAS

Figura 83. Menú de Opciones



### PRODUCTOS INGRESADOS A BODEGA

CANTIDAD	DESCRIPCIÓN	AÑO FABRICACIÓN	AÑO CADUCIDAD	P.P.P
<input type="text" value="1"/>	<input type="text"/>	<input type="text" value="1991"/>	<input type="text" value="2007"/>	<input type="text"/>
<input type="text" value="2"/>	<input type="text"/>	<input type="text" value="1991"/>	<input type="text" value="2008"/>	<input type="text"/>
<input type="text" value="3"/>	<input type="text"/>	<input type="text" value="1993"/>	<input type="text" value="2009"/>	<input type="text"/>
<input type="text" value="4"/>	<input type="text"/>	<input type="text" value="1993"/>	<input type="text" value="2010"/>	<input type="text"/>

Figura 84. Productos Ingresados a Bodega.

### CARACTERÍSTICAS DE LOS PRODUCTOS

MARCA DEL PRODUCTO	<input type="text"/>	<input type="button" value="TEMPORAL"/>
GRADOS DE ALCOHOL	<input type="text"/>	<input type="button" value="LIMPIAR"/>
LEAVAR DEL PRODUCTO	<input type="text"/>	

Figura 85. Características de los Productos.



### PRODUCTOS DE VENTA DE PROVEEDORES

PRODUCTO	CANTIDAD	PRECIO POR UNIDAD	IVA	TOTAL A PAGAR
<input type="text"/>	<input type="text" value="1"/> <input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text" value="2"/> <input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text" value="3"/> <input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text" value="4"/> <input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura 86. Producto de Venta de Proveedores

### VENTA DE PRODUCTOS DIARIOS

REGIA	CANTIDAD	PRECIO POR UNIDAD	IVA	TOTAL A PAGAR
<input type="text" value="1"/> <input type="text" value="0"/>	<input type="text" value="1"/> <input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="2"/> <input type="text" value="0"/>	<input type="text" value="2"/> <input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="3"/> <input type="text" value="0"/>	<input type="text" value="3"/> <input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="4"/> <input type="text" value="0"/>	<input type="text" value="4"/> <input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura 87. Venta de Productos Diarios.



#### 4.6.2 Prototipado Pantallas De Administrador

**GESTIONAR USUARIO**

REGISTRO DE USUARIO

APPELLIDO:	<input type="text"/>
NOMBRE:	<input type="text"/>
CORREO ELECTRONICO:	<input type="text"/>
CONTRASEÑA:	<input type="text"/>

---

SEGURIDAD DE USUARIO

USUARIO:	<input type="text"/>
CONTRASEÑA:	<input type="text"/>

< ATRAS

Figura 87. Gestionar Usuario

**GESTIONAR MENU DE OPCIONES**

PRODUCIDOS ENFERMERA A BORDA:	<input type="button" value="PERMITIR MENÚ"/>
PRODUCIDOS DE AYUDA DE PROVEEDORES:	<input type="button" value="PERMITIR MENÚ"/>
ENFERMEROS DE PRODUCTOS:	<input type="button" value="PERMITIR MENÚ"/>
SEÑA DE PRODUCTOS DADOS:	<input type="button" value="PERMITIR MENÚ"/>

< ATRAS

Figura 88. Gestionar Menú de Opciones.



**GESTIONAR COMPRA DE PRODUCTOS**

STOCK DISPONIBLE	CANTIDAD A COMPRAR, PROVEEDOR	CANTIDAD VENDIDA DESDE
<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="2"/>
<input type="text" value="3"/>	<input type="text" value="3"/>	<input type="text" value="3"/>

[← ATRAS](#)

Figura 89. Gestionar Compra de Productos.

**GESTIONAR VENTAS DE PRODUCTOS**

USU	CANTIDAD	TOTAL INGRESO DIARIO
LINER	<input type="text" value="1"/>	<input type="text"/>
MARIN	<input type="text" value="2"/>	<input type="text"/>
HIDROGEN	<input type="text" value="3"/>	<input type="text"/>
PLASMA	<input type="text" value="4"/>	<input type="text"/>
VIDEOS	<input type="text" value="5"/>	<input type="text"/>
SABADO	<input type="text" value="6"/>	<input type="text"/>

[← ATRAS](#)

Figura 90. Gestionar Ventas de Productos.

#### 4.7 Diagrama Entidad-Relación de la Base de Datos

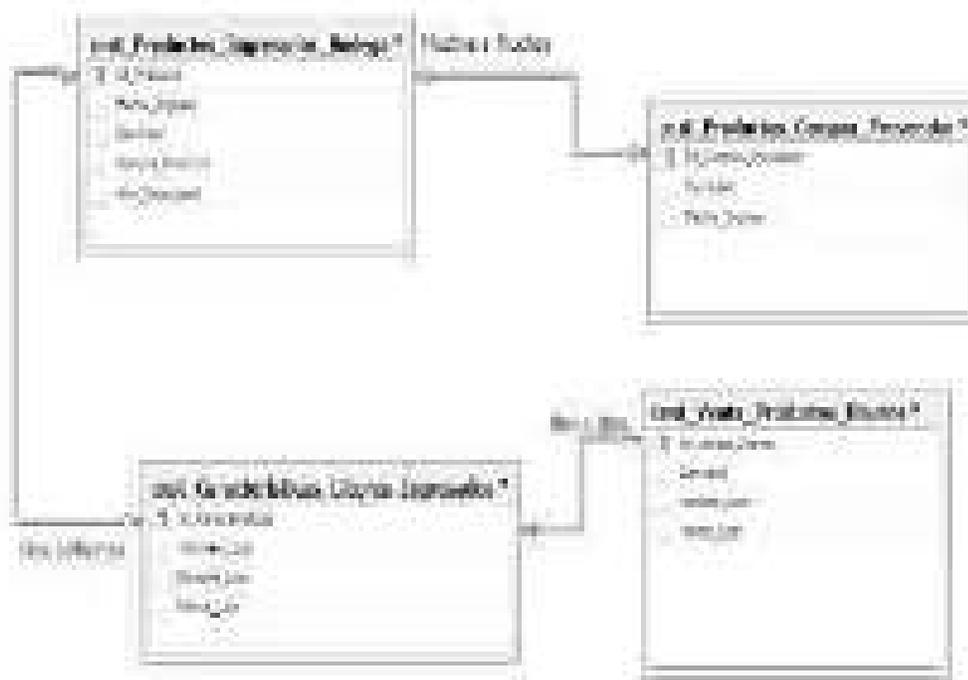


Figura 91. Diagrama Entidad-Relación de la Base de Datos

#### 4.8 Presupuesto

##### 4.8.1 Requerimiento de software

Cantidad	Programa	Descripción
1	Windows 7 Pro	Sistema Operativo
1	Servicio de Internet Mensual	Internet
1	Visual Studio 2015	Interfaz de diseño
1	Balsamiq Wireframes	Prototipos de software
1	SQL Server	Bases de Datos



#### 4.8.2 Costo de Software

Cantidad	Descripción	Precio Unitario	Precio Total
1	Licencia Windows 7 Pro	\$130	\$130
6	Servicio de Internet Mensual (6 Meses)	\$25	\$175
1	SQL Server	\$12	\$ 12
<b>TOTAL</b>			<b>\$317</b>

#### 4.8.3 Costo del Sistema

Descripción	Precio Total
Precio del software	\$317
Precio del trabajo	\$400
<b>TOTAL</b>	<b>\$717</b>

#### 4.8.4 Requerimiento del Personal

Etapa	Personal
Diseño	1 Analista/ Diseñador
Desarrollo	1 Programador

## CONCLUSIONES

La salsoteca Barreto propondrá un diseño de software de escritorio para el control de inventarios en los licores de compra y venta, para satisfacer las necesidades de los propietarios en cuanto al manejo de inventarios y evitar la pérdida de información de los licores.

Al administrador permitirá un manejo y control permanente de los licores que llegan y se venden en la salsoteca, mejorando su servicio de control de mercadería dentro del local y evitar escases de productos.

El usuario registra, guarda, actualiza o elimina la información, y esta información la gestiona el administrador del software para controlar los productos y mantener el negocio hacia el éxito comercial. El usuario, por otra parte, no puede gestionar el software ya que sus obligaciones laborales solo se centran en mantener el control de los productos y que estos sean oportunamente notificados al administrador.

Finalmente cabe mencionar que todo el diseño del software fue diseñado en software libre, sin involucrar a la salsoteca a tener que cancelar licencias por distintos softwares adicionales.

## **RECOMENDACIONES**

Una vez culminado este proyecto de investigación, y realizado el diseño de este software como es el diseño de control de inventarios para los licores de compra y venta de la Salsoteca Barreto, se recomienda que se continúe con el desarrollo con las fases que quedaron diseñadas.

Se recomienda además, que para tener un mejor y buen control no se debe dejar extender por mucho tiempo los controles y conteos en la bodega del negocio, tanto para los productos ya vendidos como para los existentes.

También se recomienda que para evitar pérdidas innecesarias dentro de la salsoteca, se deben controlar frecuentemente las fechas de elaboración y fecha de caducidad de cada uno de los productos que son adquiridos a los proveedores.

Ofrecer los mejores productos a los clientes, que sean de calidad y que estén registrados en el sistema, evitar que se dañe o se pierda los productos por fechas cercanas a su caducidad.

## BIBLIOGRAFIA

- Arias, F. G. (2006). El Proyecto de Investigación. En F. G. Arias, *El Proyecto de Investigación*. Epitesme.
- Arias, F. G. (2006). El Proyecto de Investigación. En F. G. Arias, *El Proyecto de Investigación*. Episteme.
- Casas Roma, J., & Conesa i Caralt, J. (2013). Diseño conceptual de bases de datos en UML. En J. Casas Roma, & J. Conesa i Caralt, *Diseño conceptual de bases de datos en UML* (pág. 152). Barcelona, España: Universitat Oberta de Catalunya.
- Condoy, F. F. (14 de 07 de 2019). *Diseño de un sistema informático de inventario que contribuya a mejorar el control de los ingresos y egresos de los productos naturales de la microempresa Frutoterapia. (Proyecto de Grado)*. Recuperado el 05 de 09 de 2019, de Diseño de un sistema informático de inventario que contribuya a mejorar el control de los ingresos y egresos de los productos naturales de la microempresa Frutoterapia. (Proyecto de Grado): <http://repositorio.itb.edu.ec/handle/123456789/1218>
- contributors, E. (03 de 10 de 2015). *Lenguaje imperativo*. Recuperado el 09 de 09 de 2019, de Lenguaje imperativo: [http://enciclopedia.us.es/index.php?title=Lenguaje\\_imperativo&oldid=334626](http://enciclopedia.us.es/index.php?title=Lenguaje_imperativo&oldid=334626)
- EcuRed. (18 de 07 de 2019). *Arquitectura Cliente Servidor*. Recuperado el 10 de 09 de 2019, de Arquitectura Cliente Servidor: [https://www.ecured.cu/index.php?title=Arquitectura\\_Cliente\\_Servidor&oldid=3461890](https://www.ecured.cu/index.php?title=Arquitectura_Cliente_Servidor&oldid=3461890)
- EcuRed. (21 de 06 de 2019). *Arquitectura de Tres Niveles*. Recuperado el 10 de 09 de 2019, de Arquitectura de Tres Niveles: [https://www.ecured.cu/index.php?title=Arquitectura\\_de\\_tres\\_niveles&oldid=3418301](https://www.ecured.cu/index.php?title=Arquitectura_de_tres_niveles&oldid=3418301).
- EcuRed. (18 de 07 de 2019). *Arquitectura Orientada a Servicios (SOA)*. Recuperado el 10 de 09 de 2019, de Arquitectura Orientada a Servicios (SOA): [https://www.ecured.cu/index.php?title=Arquitectura\\_Orientada\\_a\\_Servicios\\_\(SOA\)&oldid=3461867](https://www.ecured.cu/index.php?title=Arquitectura_Orientada_a_Servicios_(SOA)&oldid=3461867)
- EcuRed. (06 de 07 de 2019). *Metodologías de desarrollo de Software*. Recuperado el 11 de 09 de 2019, de Metodologías de desarrollo de Software: [https://www.ecured.cu/index.php?title=Metodologias\\_de\\_desarrollo\\_de\\_Software&oldid=3443438](https://www.ecured.cu/index.php?title=Metodologias_de_desarrollo_de_Software&oldid=3443438)
- EcuRed. (31 de 08 de 2019). *Prolog (lenguaje de programación)*. Recuperado el 18 de 09 de 2019, de Prolog (lenguaje de programación):

[https://www.ecured.cu/index.php?title=Prolog\\_\(lenguaje\\_de\\_programaci%C3%B3n\)&oldid=3536173](https://www.ecured.cu/index.php?title=Prolog_(lenguaje_de_programaci%C3%B3n)&oldid=3536173)

escritorio, A. d. (28 de 07 de 2019). *Wikipedia, La enciclopedia libre*. Recuperado el 09 de 09 de 2019, de Wikipedia, La enciclopedia libre:

[https://es.wikipedia.org/w/index.php?title=Aplicaci%C3%B3n\\_de\\_escritorio&oldid=117789344](https://es.wikipedia.org/w/index.php?title=Aplicaci%C3%B3n_de_escritorio&oldid=117789344).

EZPINOZA, O. (04 de 02 de 2013). *La administracion eficiente de los inventarios*.

Recuperado el 05 de 09 de 2019, de La administracion eficiente de los inventarios: <http://inventariosautores.blogspot.com/2013/02/control-de-inventarios-segun-autores.html>

Fabián, V. B. (2014). *Tesis de Pregrado - Ingeniería en Sistemas*. Recuperado el 09 de 09 de 2019, de Tesis de Pregrado - Ingeniería en Sistemas:

<http://repositorio.uteq.edu.ec/handle/43000/135>

Gálan, J. S. (30 de 10 de 2015). *Venta de Mercaderías*. Recuperado el 12 de 09 de 2019, de Venta de Mercaderías: <https://economipedia.com/definiciones/venta-de-mercaderias.html>

Grau, J. L. (s.f.). *¿SABES CÓMO FUNCIONA XP?* Recuperado el 12 de 09 de 2019, de ¿SABES CÓMO FUNCIONA XP?: <https://managementplaza.es/blog/sabes-como-funciona-xp/>

Guide, D. (23 de 01 de 2019). *Qt: el framework en C++ para el desarrollo de software multiplataforma*. Recuperado el 11 de 09 de 2019, de Qt: el framework en C++ para el desarrollo de software multiplataforma:

<https://www.ionos.es/digitalguide/servidores/know-how/qt/>

GUIDOBEYMARPEREZLOPEZ. (06 de 01 de 2015). *SISTEMAS DE ADMINISTRACIÓN DE BASES DE DATOS DBMS*. Recuperado el 10 de 09 de 2019, de SISTEMAS DE ADMINISTRACIÓN DE BASES DE DATOS DBMS:

<https://guidobeymarperezlopez.wordpress.com/2015/01/06/sistema-de-administracion-de-base-de-datos-dbms/>

Gutiérrez, M. A. (03 de 2007). *Sistema para el Control de Ventas de Inventarios de la Empresa Antiguo Arte Europeo S.A de C.V*. Recuperado el 09 de 09 de 2019, de Sistema para el Control de Ventas de Inventarios de la Empresa Antiguo Arte Europeo S.A de C.V:

<https://www.uaeh.edu.mx/docencia/Tesis/icbi/licenciatura/documentos/Sistema%20para%20el%20control%20de%20ventas%20e%20inventarios.pdf>

Informativa, A. (06 de 07 de 2019). *Listado de rápidas Ruby frameworks para optimizar tu proceso de desarrollo*. Recuperado el 11 de 09 de 2019, de Listado de rápidas

- Ruby frameworks para optimizar tu proceso de desarrollo:  
<https://blog.aulaformativa.com/listados-ruby-frameworks/>
- Joseph, C. (s.f.). *Ventajas y desventajas de un sistema manual de control de inventario*. Recuperado el 17 de 09 de 2019, de Ventajas y desventajas de un sistema manual de control de inventario: <https://pyme.lavoztx.com/ventajas-y-desventajas-de-un-sistema-manual-de-control-de-inventario-6045.html>
- JUNCO, Y. G. (15 de 07 de 2019). *Diseño de un software de inventario y control de mercadería de la tienda el Idolo del Cantón Samborondón 2017. (Proyecto de Grado)*. Instituto Superior Tecnológico Bolivariano de Tecnología. Guayaquil, Ecuador. Recuperado el 05 de 09 de 2019, de Diseño de un software de inventario y control de mercadería de la tienda el Idolo del Cantón Samborondón 2017. (Proyecto de Grado). Instituto Superior Tecnológico Bolivariano de Tecnología. Guayaquil, Ecuador.:  
<http://repositorio.itb.edu.ec/handle/123456789/1250>
- Llopis Pascual, F., Pérez López, E., & Ortuño Ortín, F. (2000). Introducción a la programación, algoritmos y C/C++. En F. Llopis Pascual, E. Pérez López, & F. Ortuño Ortín, *Introducción a la programación, algoritmos y C/C++* (pág. 400). Alicante, España: Publicaciones de la Universidad de Alicante.
- Márquez, A. G. (s.f.). *Inventarios, procesos de control y gestión*. Recuperado el 94 de 09 de 2019, de Inventarios, procesos de control y gestión: <https://www.monografias.com/trabajos93/inventarios-procesos-control-y-gestion/inventarios-procesos-control-y-gestion.shtml#bibliogara>
- Martín Escofet, C. (2002). El lenguaje SQL. En C. Martín Escofet, *El lenguaje SQL* (pág. 62). Barcelona, España: UOC Papers.
- Ortega, J. M. (07 de 02 de 2019). *Los 4 frameworks más usados en MVVM para .NET*. Recuperado el 11 de 09 de 2019, de Los 4 frameworks más usados en MVVM para .NET: <https://openwebinars.net/blog/los-4-frameworks-mas-usados-en-mvvm-para-net/>
- Pirela, C. &. (2015). Recuperado el 09 de 09 de 2019, de <https://es.slideshare.net/lfcf27/sistema-de-control-de-inventario-investigacion-de-operaciones-ii-lfcf>
- PMOinformatica.com. (22 de 03 de 2011). *El enfoque tradicional de desarrollo de sistemas*. Recuperado el 12 de 09 de 2019, de El enfoque tradicional de desarrollo de sistemas: [pmoinformatica.com/2011/03/el-enfoque-tradicional-de-desarrollo-de.html](http://pmoinformatica.com/2011/03/el-enfoque-tradicional-de-desarrollo-de.html)

- Ponce de León Amador, P. J., & Saquete Boro, E. (2006). Introducción a la programación orientada a objetos. En P. J. Ponce de León Amador, & E. Saquete Boro, *Introducción a la programación orientada a objetos* (pág. 260). Alicante, España: Publicaciones de la Universidad de Alicante.
- Ponce de León Amador, P. J., & Saquete Boro, E. (2006). Introducción a la Programación Orientada a Objetos. En P. J. Ponce de León Amador, & E. Saquete Boro, *Introducción a la Programación Orientada a Objetos* (pág. 260). ALICANTE ESPAÑA: Publicaciones de Universidad de Alicante.
- Programación.Net. (s.f.). *Los 4 frameworks web más populares para Python*. Recuperado el 11 de 09 de 2019, de Los 4 frameworks web más populares para Python: [https://programacion.net/articulo/los\\_4\\_frameworks\\_web\\_mas\\_populares\\_para\\_python\\_1069](https://programacion.net/articulo/los_4_frameworks_web_mas_populares_para_python_1069)
- Sabino, C. (2014). El Proceso de Investigación. En C. Sabino, *El Proceso de Investigación* (pág. 262). Episteme.
- Saigua, C. L. (11 de 07 de 2019). *Diseño de software de control de inventarios para el local comercial Súper Bazar en Guayaquil, 2017. (Proyecto de Grado)*. Instituto Superior Tecnológico Bolivariano de Tecnología. Guayaquil, Ecuador. Recuperado el 05 de 09 de 2019, de Diseño de software de control de inventarios para el local comercial Súper Bazar en Guayaquil, 2017. (Proyecto de Grado). Instituto Superior Tecnológico Bolivariano de Tecnología. Guayaquil, Ecuador.: <http://repositorio.itb.edu.ec/handle/123456789/1196>
- Sarría, F. A. (s.f.). *Programación en SQL con PostgreSQL*. Recuperado el 11 de 09 de 2019, de Programación en SQL con PostgreSQL: <https://www.um.es/geograf/sigmur/sigpdf/postgresql.pdf>
- Westreicher, G. (13 de 07 de 2019). *Economipedia.com*. Recuperado el 05 de 09 de 2019, de Economipedia.com: <https://economipedia.com/definiciones/inventario-fisico.html>
- Wikipedia, C. (05 de 09 de 2019). *Python*. Recuperado el 09 de 09 de 2019, de Python: <https://es.wikipedia.org/w/index.php?title=Python&oldid=118951866>
- Wikipedia, C. (31 de 08 de 2019). *Ruby*. Recuperado el 09 de 09 de 2019, de Ruby: <https://es.wikipedia.org/w/index.php?title=Ruby&oldid=118773761>
- Wikipedia, C. (13 de 05 de 2019). *Visual Basic .Net*. Recuperado el 09 de 09 de 2019, de Visual Basic .Net: [https://es.wikipedia.org/w/index.php?title=Visual\\_Basic\\_.NET&oldid=115906065](https://es.wikipedia.org/w/index.php?title=Visual_Basic_.NET&oldid=115906065)
- Wikipedia, C. d. (18 de 07 de 2019). *Arquitectura de microservicios*. Recuperado el 10 de 09 de 2019, de Arquitectura de microservicios:

[https://es.wikipedia.org/w/index.php?title=Arquitectura\\_de\\_microservicios&oldid=117508515](https://es.wikipedia.org/w/index.php?title=Arquitectura_de_microservicios&oldid=117508515)

Wikipedia, C. d. (16 de 09 de 2019). *Arquitectura de software*. Recuperado el 18 de 09 de 2019, de Arquitectura de software:  
[https://es.wikipedia.org/w/index.php?title=Arquitectura\\_de\\_software&oldid=119415165](https://es.wikipedia.org/w/index.php?title=Arquitectura_de_software&oldid=119415165)

Wikipedia, C. d. (03 de 09 de 2019). *Arquitectura dirigida por eventos*. Recuperado el 10 de 09 de 2019, de Arquitectura dirigida por eventos:  
[https://es.wikipedia.org/w/index.php?title=Arquitectura\\_dirigida\\_por\\_eventos&oldid=118865060](https://es.wikipedia.org/w/index.php?title=Arquitectura_dirigida_por_eventos&oldid=118865060)

Wikipedia, C. d. (29 de 07 de 2019). *C Sharp*. Recuperado el 09 de 09 de 2019, de C Sharp: [https://es.wikipedia.org/w/index.php?title=C\\_Sharp&oldid=117811087](https://es.wikipedia.org/w/index.php?title=C_Sharp&oldid=117811087)

Wikipedia, C. d. (02 de 09 de 2019). *CouchDB*. Recuperado el 11 de 09 de 2019, de CouchDB:  
<https://es.wikipedia.org/w/index.php?title=CouchDB&oldid=118861213>

Wikipedia, C. d. (30 de 08 de 2019). *Desarrollo ágil de software*. Recuperado el 12 de 09 de 2019, de Desarrollo ágil de software:  
[https://es.wikipedia.org/w/index.php?title=Desarrollo\\_%C3%A1gil\\_de\\_software&oldid=118724800](https://es.wikipedia.org/w/index.php?title=Desarrollo_%C3%A1gil_de_software&oldid=118724800)

Wikipedia, C. d. (30 de 08 de 2019). *Framework*. Recuperado el 11 de 09 de 2019, de Framework:  
<https://es.wikipedia.org/w/index.php?title=Framework&oldid=118729557>

Wikipedia, C. d. (29 de 08 de 2019). *Lenguaje de Programación*. Recuperado el 09 de 09 de 2019, de Lenguaje de Programación:  
[https://es.wikipedia.org/w/index.php?title=Lenguaje\\_de\\_programaci%C3%B3n&oldid=118692733](https://es.wikipedia.org/w/index.php?title=Lenguaje_de_programaci%C3%B3n&oldid=118692733)

Wikipedia, C. d. (29 de 07 de 2019). *Lisp*. Recuperado el 09 de 09 de 2019, de Lisp:  
<https://es.wikipedia.org/w/index.php?title=Lisp&oldid=117810177>

Wikipedia, C. d. (04 de 09 de 2019). *Metodología de desarrollo de software*. Recuperado el 11 de 09 de 2019, de Metodología de desarrollo de software:  
[https://es.wikipedia.org/w/index.php?title=Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software&oldid=118933128](https://es.wikipedia.org/w/index.php?title=Metodolog%C3%ADa_de_desarrollo_de_software&oldid=118933128)

Wikipedia, C. d. (20 de 08 de 2019). *Microsoft .Net*. Recuperado el 11 de 09 de 2019, de Microsoft .Net:  
[https://es.wikipedia.org/w/index.php?title=Microsoft\\_.NET&oldid=118380456](https://es.wikipedia.org/w/index.php?title=Microsoft_.NET&oldid=118380456)

- Wikipedia, C. d. (27 de 08 de 2019). *Microsoft SQL Server*. Recuperado el 10 de 09 de 2019, de Microsoft SQL Server:  
[https://es.wikipedia.org/w/index.php?title=Microsoft\\_SQL\\_Server&oldid=118564299](https://es.wikipedia.org/w/index.php?title=Microsoft_SQL_Server&oldid=118564299)
- Wikipedia, C. d. (30 de 08 de 2019). *Modelo-Vista-Controlador*. Recuperado el 10 de 09 de 2019, de Modelo-Vista-Controlador:  
<https://es.wikipedia.org/w/index.php?title=Modelo%E2%80%933vista%E2%80%933controlador&oldid=118724645>
- Wikipedia, C. d. (22 de 08 de 2019). *MongoDB*. Recuperado el 18 de 09 de 2019, de MongoDB:  
<https://es.wikipedia.org/w/index.php?title=MongoDB&oldid=118443090>
- Wikipedia, C. d. (21 de 08 de 2019). *MySQL*. Recuperado el 11 de 09 de 2019, de MySQL:  
<https://es.wikipedia.org/w/index.php?title=MySQL&oldid=118411299>
- Wikipedia, C. d. (08 de 08 de 2019). *NoSQL*. Recuperado el 18 de 09 de 2019, de NoSQL:  
<https://es.wikipedia.org/w/index.php?title=NoSQL&oldid=118103658>
- Wikipedia, C. d. (14 de 08 de 2019). *Oracle Database*. Recuperado el 10 de 09 de 2019, de Oracle Database:  
[https://es.wikipedia.org/w/index.php?title=Oracle\\_Database&oldid=118244706](https://es.wikipedia.org/w/index.php?title=Oracle_Database&oldid=118244706)
- Wikipedia, C. d. (09 de 09 de 2019). *Proceso Unificado de Rational*. Recuperado el 12 de 09 de 2019, de Proceso Unificado de Rational:  
[https://es.wikipedia.org/w/index.php?title=Proceso\\_Unificado\\_de\\_Rational&oldid=119113212](https://es.wikipedia.org/w/index.php?title=Proceso_Unificado_de_Rational&oldid=119113212)
- Wikipedia, C. d. (16 de 07 de 2019). *Redis*. Recuperado el 11 de 09 de 2019, de Redis:  
<https://es.wikipedia.org/w/index.php?title=Redis&oldid=117458064>
- Wikipedia, C. d. (13 de 07 de 2019). *Scala*. Recuperado el 09 de 09 de 2019, de Scala:  
[https://es.wikipedia.org/w/index.php?title=Scala\\_\(lenguaje\\_de\\_programaci%C3%B3n\)&oldid=117389656](https://es.wikipedia.org/w/index.php?title=Scala_(lenguaje_de_programaci%C3%B3n)&oldid=117389656)
- Wikipedia, C. d. (21 de 08 de 2019). *Scheme*. Recuperado el 09 de 09 de 2019, de Scheme: <https://es.wikipedia.org/w/index.php?title=Scheme&oldid=118416182>
- Wikipedia, C. d. (09 de 09 de 2019). *Scrum (desarrollo de software)*. Recuperado el 12 de 09 de 2019, de Scrum (desarrollo de software):



[https://es.wikipedia.org/w/index.php?title=Scrum\\_\(desarrollo\\_de\\_software\)&oldid=119110](https://es.wikipedia.org/w/index.php?title=Scrum_(desarrollo_de_software)&oldid=119110)  
432