**INSTITUTO TECNICO NACIONAL DE COMERCIO**

**“FEDERICO ALVAREZ PLATA”**

**NOCTURNO**



SISTEMA WEB DE VENTA DE ARTICULOS PARA MASCOTAS

**PROYECTO DE GRADO PARA OPTAR AL**

**TITULO DE TÉCNICO SUPERIOR EN**

**SISTEMAS INFORMÁTICOS**

**ESTUDIANTE:** DIEGO ARMANDO MENECES ORELLANA

**TUTOR:** ING. PAVEL CASCERES

**COCHABAMBA - BOLIVIA**

**DEDICATORIA**

Al creador de todas las cosas, el que me ha dado fortaleza para continuar cuando a punto de caer he estado; por ello, con toda la humildad que de mi corazón puede emanar, dedico primeramente mi trabajo a Dios.

De igual forma, dedico este proyecto a mi madre y mi padre que ha sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante en los momentos más difíciles.

A mi Novia Mary Cruz Laime que siempre ha estado esperándome después de mis clases en el instituto dándome su apoyo.

A mi familia en general, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momento.

Y a mis amigos que gracias a su apoyo, y conocimientos hicieron de esta experiencia una de las más especiales.

**AGRADECIMIENTOS**

Agradezco a Dios por protegerme durante todo mi camino y darme fuerzas para superar obstáculos y dificultades a lo largo de toda mi vida.

A mi madre, que con su demostración de una madre ejemplar me ha enseñado a no desfallecer ni rendirme ante nada y siempre perseverar a través de sus sabios consejos.

A mi padre, por su apoyo incondicional y por demostrarme la gran fe que tienen en mí.

A mis amigos, por acompañarme durante todo este arduo camino y compartir conmigo alegrías y fracasos.

Al Ing. Pavel Casceres, tutor de mi proyecto de grado , por su valiosa guía y asesoramiento a la realización de la misma.

Gracias a todas las personas que ayudaron directa e indirectamente en la realización de este proyecto.

**TABLA DE CONTENIDOS**

[DEDICATORIA 2](#_30j0zll)

[AGRADECIMIENTOS 3](#_1fob9te)

[TABLA DE CONTENIDOS 4](#_3znysh7)

[ÍNDICE DE TABLAS 6](#_2et92p0)

[ÍNDICE DE FIGURAS 7](#_tyjcwt)

[**RESUMEN 8**](#_3dy6vkm)

[**CAPÍTULO I 1**](#_1t3h5sf)

[**1.1.**](#_4d34og8) **Tema 1**

[**1.2.**](#_2s8eyo1) **Diagnóstico y justificación 1**

[**1.1.**](#_17dp8vu) **Planteamiento y formulación del problema técnico/tecnológico 2**

[**1.2.**](#_3rdcrjn) **Objetivos 3**

[1.2.1.](#_26in1rg) General 3

[1.2.2.](#_lnxbz9) Específicos 3

[**1.3.**](#_35nkun2) **Alcance 4**

[**1.4.**](#_1ksv4uv) **Enfoque metodológico 6**

[1.4.1.](#_44sinio) Métodos 6

[1.4.2.](#_2jxsxqh) Técnicas 7

[**CAPÍTULO 2 Marco Teórico Conceptual 8**](#_z337ya)

[**2.1.**](#_3j2qqm3)  SISTEMAS DE INFORMACION  **8**

[2.1.1.](#_1y810tw) Ingenieria de software 8

[2.1.2.](#_4i7ojhp) ASPECTOS DE PRODUCCION DE SOFTWARE 8

[**2.2.**](#_2xcytpi) **base de datos 9**

[2.2.1.](#_1ci93xb) Sistema gestor de base de datos 9

[2.2.2.](#_3whwml4) arquitectura de diseno mvc 9

2.2.3. Características de la programación orientada a objetos 9

2.2.3.1. Abstracción 9

2.2.3.2. Encapsulamiento 9

2.2.3.3. Modularidad 9

2.2.3.4. Principio de Ocultamiento 9

2.2.3.5. Polimorfismo 9

2.2.3.6. Herencia 9

2.2.4. Arquitectura de los sistemas web 9

2.2.5. Arquitectura 3 capas 9

[**CAPÍTULO 3 Propuesta de Innovación o Solución del Problema 10**](#_2bn6wsx)

[**3.1.**](#_qsh70q) **Análisis de requerimientos 10**

[**3.2.**](#_3as4poj) **Diagramas de casos de uso 10**

[**3.3.**](#_1pxezwc) **Diagrama de clases 11**

[**3.4.**](#_49x2ik5) **Diagrama relacional 11**

[**3.5.**](#_2p2csry) **Arquitectura del sistema 11**

[**3.6.**](#_147n2zr) **Diagrama de secuencias 11**

[**3.7.**](#_3o7alnk) **Diagrama de componentes 11**

[**3.8.**](#_23ckvvd) **Pruebas de calidad 11**

[**3.9.**](#_ihv636) **Documentación de la prueba del prototipo 11**

[RESULTADOS ESPERADOS 12](#_32hioqz)

[CONCLUSIONES 14](#_1hmsyys)

[RECOMENDACIONES 15](#_41mghml)

[FUENTES DE INFORMACIÓN BIBLIOGRÁFICA 16](#_2grqrue)

[ANEXOS 1](#_vx1227)

**ÍNDICE DE TABLAS**

[Tabla 1 Tabla de Requerimientos 1](#_3fwokq0)

[Tabla 2 Tabla de resultados 2](#_1v1yuxt)

**ÍNDICE DE FIGURAS**

[Figura 1 Diagrama de Casos de Uso 2](#_4f1mdlm)

# RESUMEN

El presente proyecto de grado es un sistema web de venta que trata sobre la comercialización de artículos para mascotas para tener una mejor gestión de ingresos y egresos.

Muchas empresas aun no usan los sistemas web administran sus ventas en cuadernos donde sus gestiones de ventas son muy tediosas.

El presente proyecto fue desarrollado para facilitar un mejor manejo de gestión de ventas donde la empresa que lo use saque provecho a las funciones con las que fue desarrollado.

INTRODUCCIÓN

Cada vez más lejana queda la época en que a los perros u otros animales domésticos se les daba sobras de comida. Y es que, en la actualidad, muchas personas tratan a sus mascotas como un miembro más de su familia, brindándole no solo alimento especial, sino otros cuidados. El negocio de los alimentos para mascotas crece a un ritmo anual del 6% en América Latina, según el reporte de Euromonitor. Una prueba de ello son las importaciones de alimento para mascotas, que aumentaron en un 71,1% en los últimos cinco años. Pasaron de $us 9,1 millones en 2013 a $us 15,5 millones el año pasado. A septiembre de 2018, las compras externas sumaron $us 12,6 millones, según datos del Instituto Boliviano de Comercio Exterior (IBCE). Otro indicador que confirma que el negocio de las mascotas está en alza, es el número de veterinarias y puntos de venta de alimentos balanceados que abrieron en Bolivia en los últimos cinco años. Las personas en la actualidad están cuidando más a sus mascotas, las alimentan con alimento balanceado.

El proyecto tiene como objetivo desarrollar un sistema que gestione los servicios de venta que brinda una empresa de venta de alimento balanceado para mascotas, ofreciendo una variedad de alimentos y artículos en general para tener un mejor control y comercialización donde el sistema brindara un reporte de que productos están disponibles y cuáles son sus características detalladas de la entrega del producto comprado a domicilio.

Para realizar el desarrollo del sistema se seguirá con el riguroso proceso de desarrollo de software RUP (Proceso Racional Unificado), que aplica un desarrollo iterativo e incremental, contemplando las etapas de diseño, elaboración y construcción del sistema, aplicando siempre las mejores prácticas de la ingeniería de software.

**CAPITULO 1**

**PLANTEAMIENTO DEL PROBLEMA**



# Introducción

## Tema

El tema a desarrollar será el siguiente:

“Sistema web de venta de artículos de mascotas”

## Diagnóstico y justificación

En la actualidad las ventas de cualquier producto, ya sean artículos (de Personas o Mascotas), materiales de construcción, artículos o accesorios, etc. En nuestro país la compra se hace en los mercados o Tiendas de lugar, arriesgándose de esta manera a diferentes peligros que existen.

Por ello este proyecto se enfocará en un sistema de venta de artículos para mascotas ya que en la actualidad la tecnología da mayores ventajas. Con esta pequeña explicación realizada se hizo un Estudio de mercado (La Pampa - Calatayud) para conocer el consumo promedio, preferencias de compra y conocimientos de marca de los accesorios para mascotas, la encuesta se realizó a 100 personas con los siguientes resultados:

**¿Compras productos para tu mascota?**

**Tabla 1: Representación gráfica de encuesta de venta productos alimenticios en los mercados de Cochabamba (La Pampa-Calatayud)**

**Fuente: Elaboración Propia Cochabamba 2020**

\*Sí, frecuentemente 51.5%

\*Sí, de vez en cuando 43.3%

\*No compro productos para mascota 5.2%

Según la encuesta realizada la mayor parte de las personas que circulan por el mercado (La pampa-Calatayud) acuden a comprar algún producto de alimento balanceado para su mascota.

**¿Qué productos compran más para sus mascotas?**

**Tabla 2: Representación gráfica en una encuesta general tanto a clientes como a comerciantes en el mercado La Pampa dándonos a conocer que es lo que mayormente los clientes compran y que productos son los que más venden los comerciantes.**

**Fuente: Elaboración propia Cochabamba 2020**

Según la encuesta realizada tanto a comerciantes y clientes En el mercado la Pampa se dio los siguientes resultados:

**Comerciante:** nos dieron a conocer un porcentaje de sus ventas y cuál es el tipo de producto que más venden con la siguiente calificación:

\*Alimento balanceado 70%

\*Accesorios (Pecheras,Collares,Ropa,) 20%

\*Cosméticos (shampoo anti pulgas,arenas,perfumes) 10%

**Cliente:** Aprovechando la compra del cliente logramos hacer una encuesta de que productos mayormente compran para sus mascotas y el resultado es el siguiente:

\*Alimento balanceado 80%

\*Accesorios (Pecheras,Collares,Ropa,) 10%

\*Cosméticos (shampoo anti pulgas,arenas,perfumes) 10%

## Planteamiento y formulación del problema técnico/tecnológico

Lamentablemente en Bolivia no utilizan los nuevos métodos para registrar sus productos son pocas las personas que usan sistemas webs para registrar los productos vendidos y esto se debe a que muchas personas recién se están informando sobre lo que son los sistemas webs.

Esto hace que las personas tengan que anotarse o simplemente realizar una venta sin entregar algún comprobante ni registrar la venta de los productos que se venden, en este caso sería bueno optar por los sistemas de ventas para poder tener un mejor registro de los productos que se vendieron en dichas gestiones.

¿Cómo aplicar la tecnología para gestionar los servicios de venta que ofrece un sistema web de venta de artículos de mascotas?

## Objetivos

### General

Desarrollar un sistema web de gestión de venta de artículos para mascotas.

### Específicos

El sistema de venta será diseñado y podrá ser utilizada por cualquier usuario el mismo sistema se desarrollará con los siguientes módulos:

\* Desarrollar el Módulo de gestión de usuarios.

\* Desarrollar el Módulo para registro de productos para mascotas.

\* Desarrollar el Módulo para venta de artículos para mascotas.

\* Desarrollar el Módulo para registro de atributos de artículos para mascotas.

\* Desarrollar el Modulo de productos vendidos.

\* Desarrollar el Modulo de reportes.

## Alcances

El sistema de venta será diseñado con los siguientes alcances y deberá ser utilizada por cualquier usuario, se tiene los siguientes:

**\* Se desarrollará el Módulo de gestión de usuarios: donde el modulo tendrá los siguientes roles:**

**Administrador:** Este rol podrá controlar el mercadeo de productos y actualización de los mismos productos que se encuentran a la venta también podrá solicitar más productos de stock a los proveedores para tener un sistema de venta completo sin que falte producto alguno también para administrar las cuentas de usuarios comunes.

**Usuario común:** El usuario común podrá registrarse, crearse una cuenta de perfil donde podrá añadir fondos para gastarlos en los productos que se encuentren en el sistema de venta.

**\* Se desarrollará el Módulo para Agregar, eliminar, modificar artículos para mascotas:** donde el modulo tendrá los siguientes roles

**Administrador:** Este rol podrá administrar (agregar eliminar modificar) los productos en un inventario para su dicho registro de precio y cantidad de productos que están disponibles también podrá registrar dichas ofertas dependiendo a la cantidad de productos que el usuario común quiera comprar.

**Usuario común:** el usuario común podrá agregar los productos que esté comprando también podrá eliminar de su carrito los productos que marcaron por equivocación y modificar los productos antes de la venta en caso de que se haya equivocado, también podrá buscar el producto que el usuario común necesite para su carrito de compras.

**\* Se desarrollará el Módulo para venta de artículos para mascotas:** el siguiente modulo Tendrá solo los artículos para mascotas, donde el usuario podrá ver el precio y qué marca o material está compuesto tiene dicho producto, Si el alimento es adecuado y correcto el usuario podrá añadir al carrito de compras y realizar dicho pedido.

**\* Se desarrollará el Módulo para recepción de consultas de clientes:** El siguiente módulo podrá ser utilizado por los usuarios comunes donde podrán hacer sus consultas de algún producto comprado o realizar dichos pedidos de algún producto que no se encuentre en el sistema de venta.

\* **Se desarrollará el Módulo de reportes de gestión y de los productos más vendidos:** En este módulo el administrador podrá ver todo lo vendido en las fechas que el ingrese para tener un mejor control y registro de sus ganancias en su propio negocio.

## Enfoque metodológico

### Métodos

El método que se hará será el Método Experimental. Este enfoque nos permitirá descubrir si hay un cambio positivo o negativo al momento de aplicar un sistema web y una aplicación móvil para la venta de alimentos y artículos de mascotas, tomando como variables la información almacenada de los diferentes productos en un periodo de tiempo determinado.

Nuestra Investigación será Aplicada Tecnológica. Se ha optado por la mencionada investigación debido a que está orientada al sector de servicios, se aplicaran todos los conocimientos adquiridos para mejorar la venta de productos vía internet, apoyado en el desarrollo de sistemas web y aplicación móvil utilizando todo el conocimiento aprendido en el instituto.

Las herramientas a utilizarse en el desarrollo web serán: Diseño de la base de Datos, Lenguaje de programación PHP, gestor de base de Datos MySQL. Además de herramientas de programación como son los frameworks (Boostrap 4, Composer.chart.js) para diseño web.

### Técnicas

El sistema web y aplicación móvil tendrá un enfoque cuantitativo mediante técnicas de recolección de datos con base de productos vendidos. Donde la técnica solo será de tipo analítico, registrando los procesos de venta, con un control de inventario llegando a controlar la cantidad de los productos vendidos y estableciendo diferentes fechas para tener un mejor funcionamiento y aumentar los ingresos de la empresa.

**CAPÍTULO 2**

**MARCO TEORICO CONCEPTUAL**



# Marco Teórico Conceptual

## SISTEMAS DE INFORMACION

Un sistema de información es un conjunto de datos que interactúan entre si con un fin común. En informática los sistemas de información ayudan a poder administrar recolectar recuperar procesar y distribuir información.

Un sistema de información se destaca por su diseño facilidad de uso mantenimiento automático de los registros y para mantener el anonimato en informaciones irrelevantes.

### Ingeniería de software

Con los actuales métodos y técnicas de ingeniería del software se logró que la construcción de sistemas grandes y complejos sean mejor desarrollados, pero a pesar de ello en algunos casos sigue siendo habitual encontrar proyectos que se retrasan.

En los últimos años, el desarrollo más significativo en ingeniería del software ha sido la aparición de UML como estándar para la descripción de sistemas orientados a objetos y el desarrollo de métodos agiles como la programación extrema. Los métodos agiles están permitiendo el desarrollo rápido de sistemas, explícitamente implican al usuario en el equipo de trabajo y reducen el papeleo y la burocracia en el proceso de software.

### Aspectos de producción de software

Varios sistemas tienen altos aspectos y requerimientos de disponibilidad y fiabilidad y se espera que nosotros tomemos la fiabilidad como un conductor básico en la ingeniería del software. La ingeniería del software no solo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y

el desarrollo de herramientas, método y teorías de apoyo a la producción de software.

En general, los ingenieros de software adoptan un enfoque sistemático y organizado en su trabajo, ya que es la forma más efectiva de producir software de alta calidad. Sin embargo, aunque la ingeniería consiste en seleccionar el método más apropiado para un conjunto de circunstancias, un enfoque más informal y creativo de desarrollo podrían ser efectivo en algunas circunstancias.” (SOMMERVILLE 2005: 6)

Según Roger Pressman, la ingeniería de software es “La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software: es decir, la aplicación de la ingeniería al software.

## BASE DE DATOS

En la actualidad, las bases de datos se usan tan ampliamente que se pueden encontrar en organizaciones de todos los tamaños, desde grandes corporaciones y agencias gubernamentales, hasta pequeños negocios e incluso en hogares. Las actividades diarias con frecuencia lo ponen en contacto con las bases de datos, ya sea directa o indirectamente.

Un entorno de base de datos integrada tiene un único gran repositorio de datos, llamado base de datos, que usan de manera simultánea muchos departamentos y usuarios en una organización. Todos los datos que la organización necesita para un grupo específico de aplicaciones, o incluso para todas sus aplicaciones, se almacenan juntos con tan poca repetición sea posible.

### Sistema gestor de base de datos

Con el advenimiento de internet el software libre se ha consolidado como alternativa técnicamente viable y económicamente sostenible al software comercial, contrariamente a lo que a menudo se piensa, convirtiéndose el software libre como otra alternativa para ofrecer los mismos servicios a un coste cada vez más reducido.

Estas alternativas se encuentran tanto para herramientas de ofimática como libreoffice o Microsoft office también disponemos de herramientas más avanzadas a un nivel de propósito general como MySQL, SQL Server.

Si, Por el contrario, mi base de datos requiere cierta cantidad de accesos de usuarios diversos, controles de integridad y otras funcionalidades, debería plantearme algo como MySQL,

En cuanto a si se debe ser libre o no la decisión dependerá de si disponemos de personal cualificado en cuyo caso un sistema libre es más barato y potente. En caso contrario el sistema de pago es la elección más adecuada. No obstante, todas las tecnologías disponen de servicios de soporte de gran calidad. (Ibañez, 2015)

### Arquitectura de diseño MVC

Es un patrón de arquitectura de software que separa los datos y principalmente a lo que es la lógica de negocio. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador. (Wikipedia, 2020)

El lenguaje de Internet para la creación de documentos electrónicos más conocido es el HTML, que es con el que se define la estructura y contenido de una página web. Consta de una serie de etiquetas (markups) predefinidas que permiten construir documentos que contienen títulos, párrafos y listas de texto, tablas, imágenes y otros elementos para la presentación de la información. Con el avance de la tecnología y la aparición por ejemplo del XML (Extensible Markup Language) que posibilita la creación de nuevos lenguajes o sublenguajes (XHTML, RSS, ATOM, MathML), surge la necesidad de que aquellos sistemas que simplemente ó solamente podían generar código HTML, sean modificados ó reemplazados por nuevos sistemas que si sean capaces de generar tantos formatos de salida como sean requeridos para poder orientar los negocios hacia las nuevas necesidades del mercado, de la forma más sencilla posible. Hoy en día muchos sistemas que manejan información a través de Internet, necesitan no solamente poder generar códigos en HTML para representarse en los navegadores de los usuarios, sino que también existe la necesidad de intercambiarla entre diferentes plataformas ó sistemas para lograr que la misma pueda ser interpretada según la tecnología involucrada (celulares, feeds de noticias, etc.) pero siempre partiendo de una misma base desde donde se genere dicha información. Para resolver la problemática descripta anteriormente, este proyecto trata sobre la implementación de un framework que permita desarrollar aplicaciones en las que pueda fácilmente separarse la lógica de negocios, el acceso de los datos y su forma de presentación de modo que la generación de información en diferentes formatos deje de ser un problema o una situación de difícil manejo que obligue a modificar los sistemas haciéndolos cada vez más complejos y repitiendo códigos y funcionalidades, agregándoles nuevas capas en su core cada vez que sea necesario generar una nueva salida. (Roberto, 2011)

### Características de la programación Orientada a Objetos

Las características principales de la programación orientadas a objetos son las siguientes:

Abstracción, Encapsulamiento, Modularidad, Principio de Ocultamiento, Polimorfismo, herencia.

### Abstracción

#### La abstracción expresa las principales características de un objeto, es decir donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar "cómo" se implementan estas características.

Los procesos, las funciones o los métodos pueden también ser abstraídos, y, cuando lo están, unas variedades de técnicas son requeridas para ampliar una abstracción. El proceso de abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real.

La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se requiere solucionar.

### Encapsulamiento

#### El encapsulamiento significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión (diseño estructurado) de los componentes del sistema.

### Modularidad

#### Se llama "modularidad" a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.

Estos módulos se pueden compilar por separado, pero tienen conexiones con otros módulos, al igual que la encapsulación, los lenguajes soportan el modularidad de diversas formas.

### Principio de Ocultamiento

#### Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una "interfaz" a otros objetos que específica cómo pueden interactuar con los objetos de la clase.

El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas; solamente los propios métodos internos del objeto pueden acceder a su estado.

Significa que otros objetos no puedan cambiar el estado interno de un objeto de manera inesperada, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.

### Polimorfismo

#### Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.

Las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado.

Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama

**asignación tardía** o **asignación dinámica**.

### Herencia

#### Las clases no se encuentran aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen.

La herencia organiza y facilita el polimorfismo y el encapsulamiento, permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo.

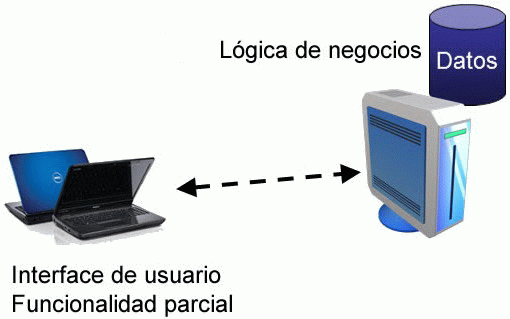
Esto suele hacerse habitualmente agrupando los objetos en **clases** y estas en **árboles** o **enrejados** que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay **herencia múltiple**; siendo de alta complejidad técnica por lo cual suele recurrirse a la herencia virtual para evitar la duplicación de datos.

### Arquitectura de los sistemas web

La arquitectura tradicional de cliente/servidor también es conocida como arquitectura de dos capas. Las aplicaciones web utilizan lo que se conocen como clientes livianos los cuales no ejecutan demasiadas labores de procesamiento para la ejecución de la aplicación misma.

Desde el punto de vista de la arquitectura se distinguen dos lados: el **cliente,** donde se encuentra el usuario final utilizando la aplicación por medio de un navegador (como, por ejemplo: Internet Explorer o Mozilla Firefox). Mediante este cliente web, el usuario interactúa con la aplicación localizada al otro lado, el **servidor,** es el lugar donde residen los datos, reglas y la lógica de la aplicación.

Por ejemplo, una estación de trabajo utilizada como cliente puede ejecutar una aplicación de interfaz de usuario que interroga a un servidor central de bases de datos.



*Modelo de Arquitectura 2 Capas*

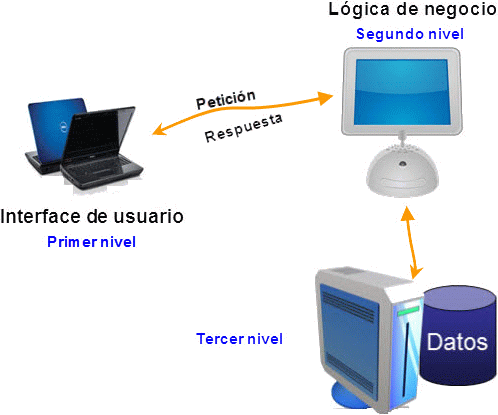
### Arquitectura 3 capas

La arquitectura de tres capas es un diseño que introduce una capa intermedia en el proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. En la arquitectura tradicional de tres capas se instala una interfaz de usuario en la computadora del usuario final (el cliente).

La arquitectura basada en Web transforma la interfaz de búsqueda existente (el explorador de Web), en la interfaz del usuario final.

#### El primer nivel consiste en la capa de presentación es la que se encarga de que el sistema o la aplicación interactúe con el usuario y viceversa.

* + - * + El segundo nivel consiste en la capa de negocio es donde residen las funciones que se ejecutan, se reciben las peticiones del usuario estos los procesa la información y envía los resultados tras él proceso.
        + Finalmente, el tercer nivel es la capa de datos es la que se encarga de almacenar los datos del sistema y de los usuarios.



*Modelo de Arquitectura 3 Capas*

#### Las diferentes capas suelen ser:

* + - * + **Capa de Presentación**: Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario. Se conoce como interfaz gráfica.
        + **Capa de Negocio**: Es donde residen los programas que se ejecutan, donde se reciben las solicitudes del usuario y envían las respuestas tras ejecutar todo el proceso.
        + **Capa de Datos:** Es donde residen los datos y es la que se encarga de acceder a los mismos. Puede estar formada por uno o más gestores de base de datos que realizan todo el almacenamiento de los datos.

(Guerrero)

**CAPITULO 3**

**PROPUESTA DE INNOVACION O SOLUCION DEL PROBLEMA**



# Propuesta de Innovación o Solución del Problema

## Análisis de requerimientos

**Actores**

Para dicho sistema de venta se necesitarán los siguientes datos:

Administrador: Posee acceso a todos los módulos Para poder controlar el inventario de los productos y para poder controlar el registro de ventas.

Clientes: En el inventario solo se registrarán los clientes que realicen una compra o recibieron su comprobante de pago para este catálogo se asignaran los siguientes datos: Nombre completo, CI, fecha, Cantidad del producto a pagar, nombre producto, material del producto, total a pagar.

Proveedor: dentro del sistema de venta el proveedor se ocupará de la siguiente información: nombre de proveedor, nombre del producto código del producto, la cantidad, y el nombre de la empresa.

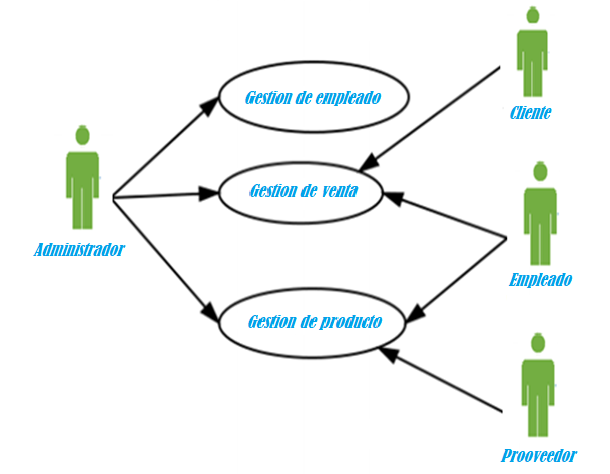
**Casos de uso**

Gestión de empleados: El caso de uso para la gestión de empleados contempla el control

de los empleados por parte del administrador.

Gestión de ventas: se llevará un registro de las ventas para facilitar el control de productos, en el apartado se producirá una nota de remisión para luego imprimirla después de haber terminado la venta para después entregársela al cliente.

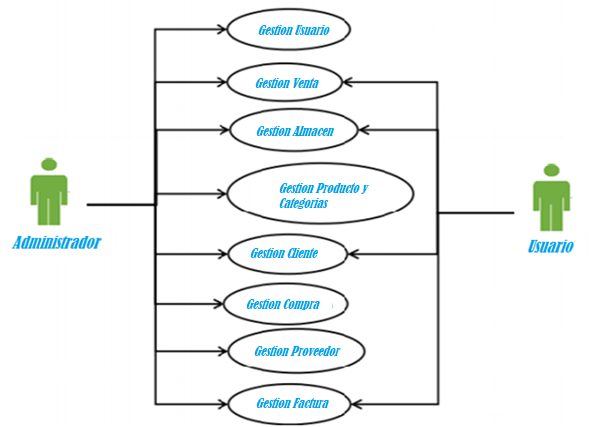
**Gestión de productos:** conjuntamente se realizan surtidos de productos para mascotas es por ello que se controlaran las adquisiciones hechas, registrando: nombre del proveedor, nombre del producto, cantidad, precio, la fecha y la hora.



*Figura 1 Caso de uso Gestión de actores*

Fuente: elaboración propia,2020

## Diagramas de casos de uso



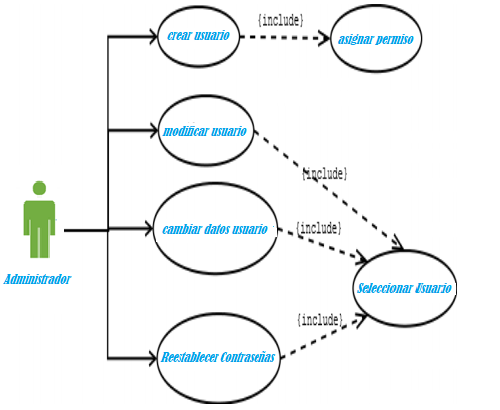
*Figura 2 Diagrama de Casos de Uso*

Fuente: elaboración propia,2020

|  |  |
| --- | --- |
| ACTOR | DESCRIPCION |
| Administrador | Tiene acceso a todos los módulos del sistema de venta |
| Usuario | Accede acorde a los permisos asignados por el administrador |

* **Gestión de usuarios**

El Administrador realizara el registro de empleado para el acceso al sistema de venta.



*Figura 5 Diagrama de Casos de Uso Gestión de usuario*

Fuente: elaboración propia,2020

Caso de uso crear usuario

|  |  |
| --- | --- |
| CASO DE USO | CREAR USUARIO |
| ACTORES | ADMINISTRADOR |
| TIPO | PRIMARIO |
| DESCRIPCION | Se realiza la creación de un usuario donde el usuario y la contraseña se genera por medio del administrador  Para que el administrador de asigne el permiso para el usuario para que pueda acceder al sistema |

Caso de uso cambiar permisos de usuarios

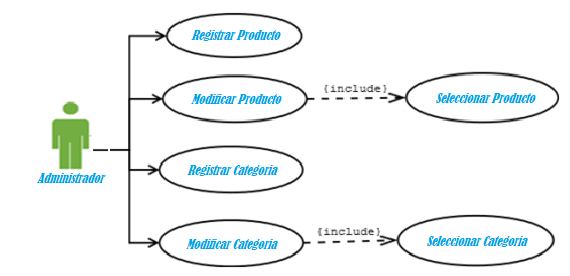
|  |  |
| --- | --- |
| CASO DE USO | CAMBIAR PERMISO DE USUARIO |
| ACTORES | ADMINISTRADOR |
| TIPO | PRIMARIO |
| DESCRIPCION | Se realiza el cambio de permisos de acceso al sistema de un usuario.  Solo el administrador puede realizar la modificación de permisos. |

Caso de uso reestablecer contraseña

|  |  |
| --- | --- |
| CASO DE USO | CAMBIAR PERMISO DE USUARIO |
| ACTORES | ADMINISTRADOR |
| TIPO | PRIMARIO |
| DESCRIPCION | Se realiza el reinicio de la contraseña del usuario en caso de que el mismo usuario no se acuerde.  El sistema realiza un reinicio de contraseña |

* **Gestión de Producto y Categorías**

La gestión de producto y categorías contempla el ingreso de artículos y productos.



*Figura 6 Diagrama de Casos de Uso Gestión de producto y Categoría*

Fuente: elaboración propia,2020

**Caso de uso registro del producto**

|  |  |
| --- | --- |
| CASO DE USO | REGISTRAR PRODUCTO |
| ACTORES | ADMINISTRADOR |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza el registro de un producto en el sistema ingresando la información.  Se ingresa el precio de compra y el precio de venta. |

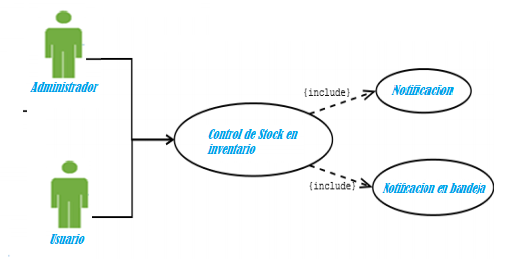
|  |  |
| --- | --- |
| CASO DE USO | MODIFICAR PRODUCTO |
| ACTORES | ADMINISTRADOR |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza la modificación de la información de un producto específico. |

|  |  |
| --- | --- |
| CASO DE USO | REGISTRAR CATEGORIA |
| ACTORES | ADMINISTRADOR |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza el registro de un producto en el  sistema ingresando la información.  Se ingresa el precio de compra y el precio  de venta. |

|  |  |
| --- | --- |
| CASO DE USO | MODIFICAR CATEGORIA |
| ACTORES | ADMINISTRADOR |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza la modificación de la información de la categoría específico. |

* GESTION DE INVENTARIO

Se realiza el control de los productos bajo stock en el almacenamiento.



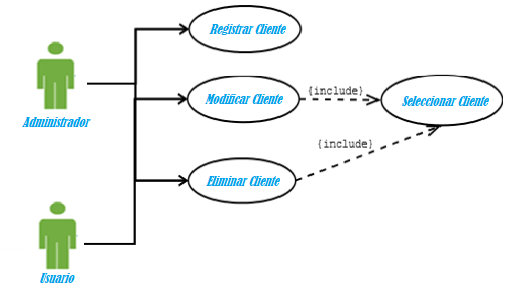
*Figura 7 Diagrama de Casos de Uso Gestión de inventario*

Fuente: elaboración propia,2020

|  |  |
| --- | --- |
| CASO DE USO | CONTROL DE STOCK INVENTARIO |
| ACTORES | ADMINISTRADOR,USUARIO |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza un control bajo stock de producto, mediante un sweetalert. |

* **GESTION DE CLIENTE**

La gestión del cliente contempla con el registro de los clientes



*Figura 8 Diagrama de Casos de Uso Gestión Cliente*

Fuente: elaboración propia,2020

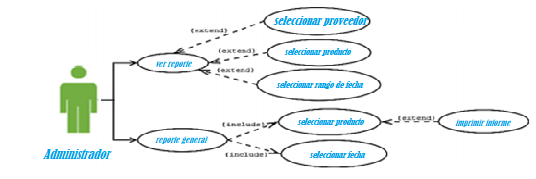
|  |  |
| --- | --- |
| CASO DE USO | REGISTRO DE USUARIO |
| ACTORES | ADMINISTRADOR,USUARIO |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza el registro de clientes en el  sistema. |

|  |  |
| --- | --- |
| CASO DE USO | MODIFICAR CLIENTE |
| ACTORES | ADMINISTRADOR,USUARIO |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza la modificación de los clientes en  el sistema. |

|  |  |
| --- | --- |
| CASO DE USO | ELIMINAR CLIENTE |
| ACTORES | ADMINISTRADOR,USUARIO |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza la eliminación del cliente del  sistema. |

* GESTION DE COMPRA

La gestión de compra y proveedor contempla la compra de un producto del proveedor.



*Figura 9 Diagrama de Casos de Uso Gestión Compra*

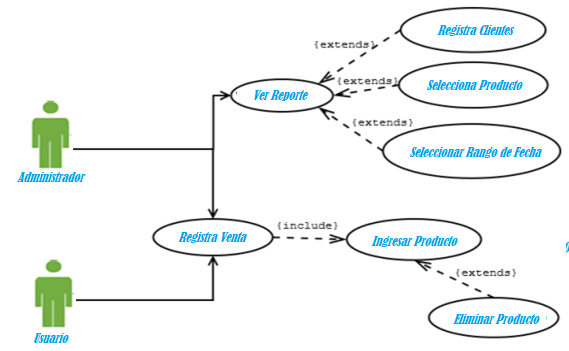
Fuente: elaboración propia,2020

|  |  |
| --- | --- |
| * CASO DE USO | REPORTE DE COMPRA |
| ACTORES | ADMINISTRADOR |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se genera un reporte de compra, para los cuales se puede tomar distintos parámetros de fecha. |

|  |  |
| --- | --- |
| CASO DE USO | REGISTRAR COMPRA |
| ACTORES | ADMINISTRADOR,USUARIO |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza el registro de una compra,  Asignando el producto a comprarse. |

* GESTION DE VENTAS

Se realizará la venta de artículos para mascotas a los clientes en general.



*Figura 10 Diagrama de Casos de Uso Gestión de Venta*

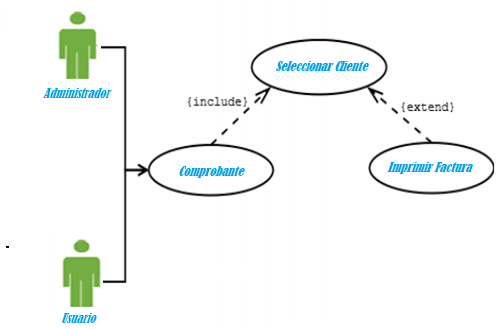
Fuente: elaboración propia,2020

|  |  |
| --- | --- |
| CASO DE USO | REGISTRAR VENTAS |
| ACTORES | ADMINISTRADOR |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se registra el producto vendido |

|  |  |
| --- | --- |
| CASO DE USO | REPORTE DE VENTAS |
| ACTORES | ADMINISTRADOR |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza la generación de reporte de ventas. De la misma forma se realiza el reporte de ventas por fecha. |

* GESTION DE COMPROBANTE DE PAGO

La gestión de factura contempla la entrega de una factura hacia el cliente.



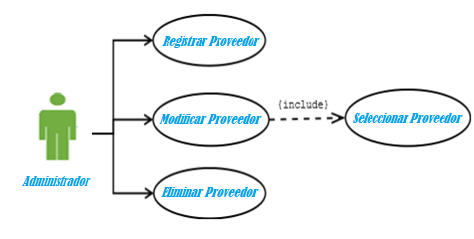
*Figura 11 Diagrama de Casos de Uso Gestión de Comprobante de Pago*

Fuente: elaboración propia,2020

|  |  |
| --- | --- |
| CASO DE USO | ENTREGAR FACTURA |
| ACTORES | ADMINISTRADOR,USUARIO |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza la entrega de la factura al cliente  de los productos que se está vendiendo. |

* GESTION DE PROVEEDOR

La gestión de proveedor contempla el registro de nuevos proveedores.



*Figura 12 Diagrama de Casos de Uso Gestión de Proveedor*

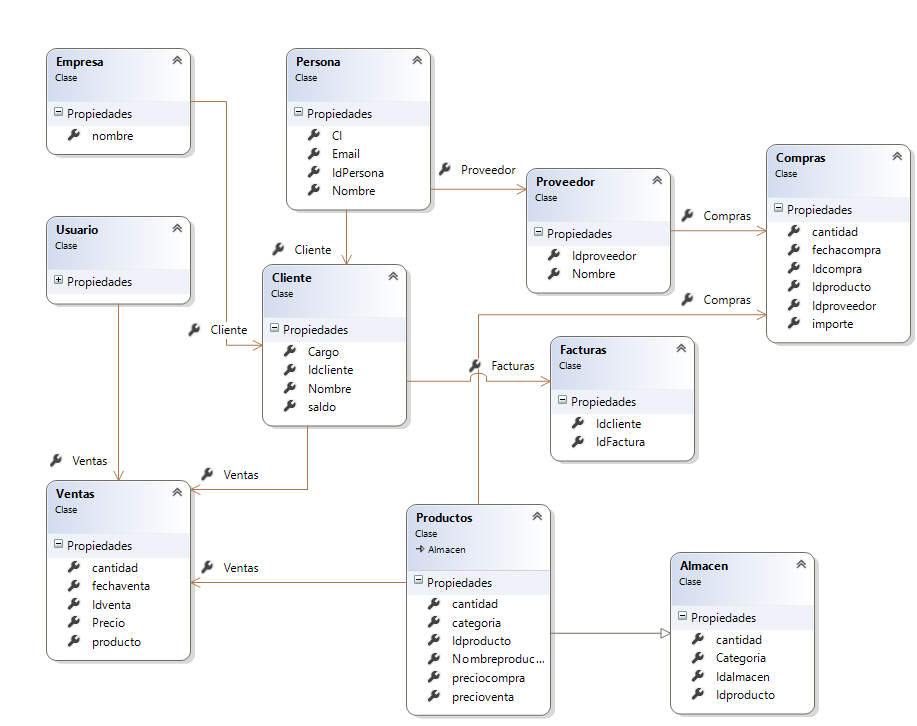
Fuente: elaboración propia,2020

|  |  |
| --- | --- |
| CASO DE USO | REGISTRAR PROVEEDOR |
| ACTORES | ADMINISTRADOR,USUARIO |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza el registro de proveedores en el  sistema. |

|  |  |
| --- | --- |
| CASO DE USO | MODIFICAR PROVEEDOR |
| ACTORES | ADMINISTRADOR,USUARIO |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza la modificación de los  proveedores en el sistema. |

|  |  |
| --- | --- |
| CASO DE USO | ELIMINAR PROVEEDOR |
| ACTORES | ADMINISTRADOR,USUARIO |
| TIPOS | PRIMARIO |
| DESCRIPCION | Se realiza la eliminación del proveedor del  sistema. |

## Diagrama de clases



## Diagrama relacional

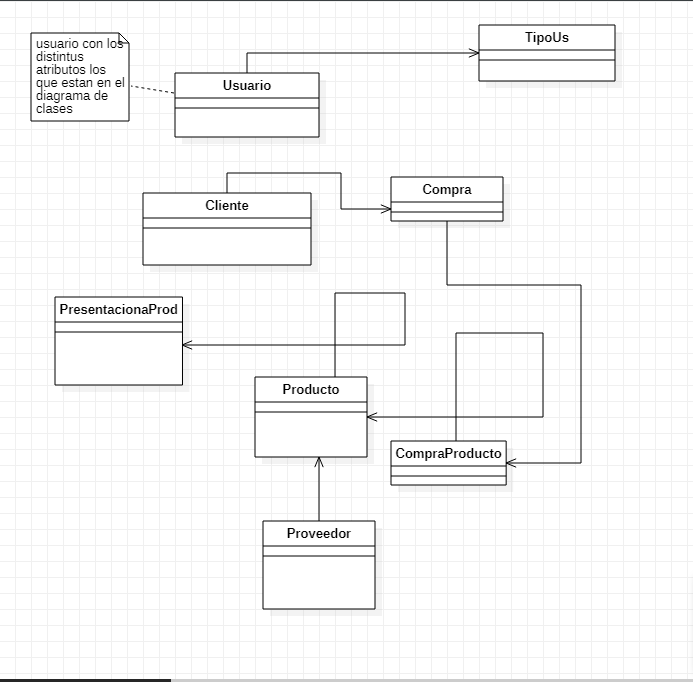
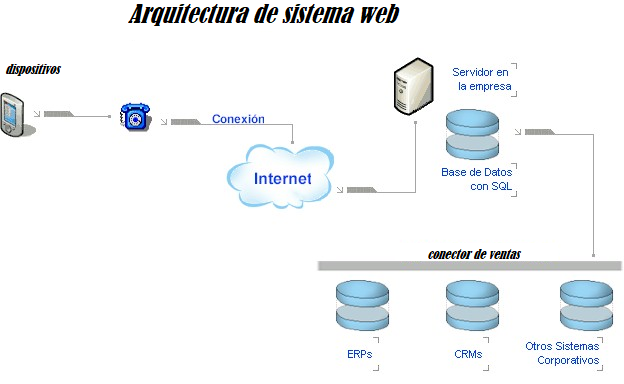
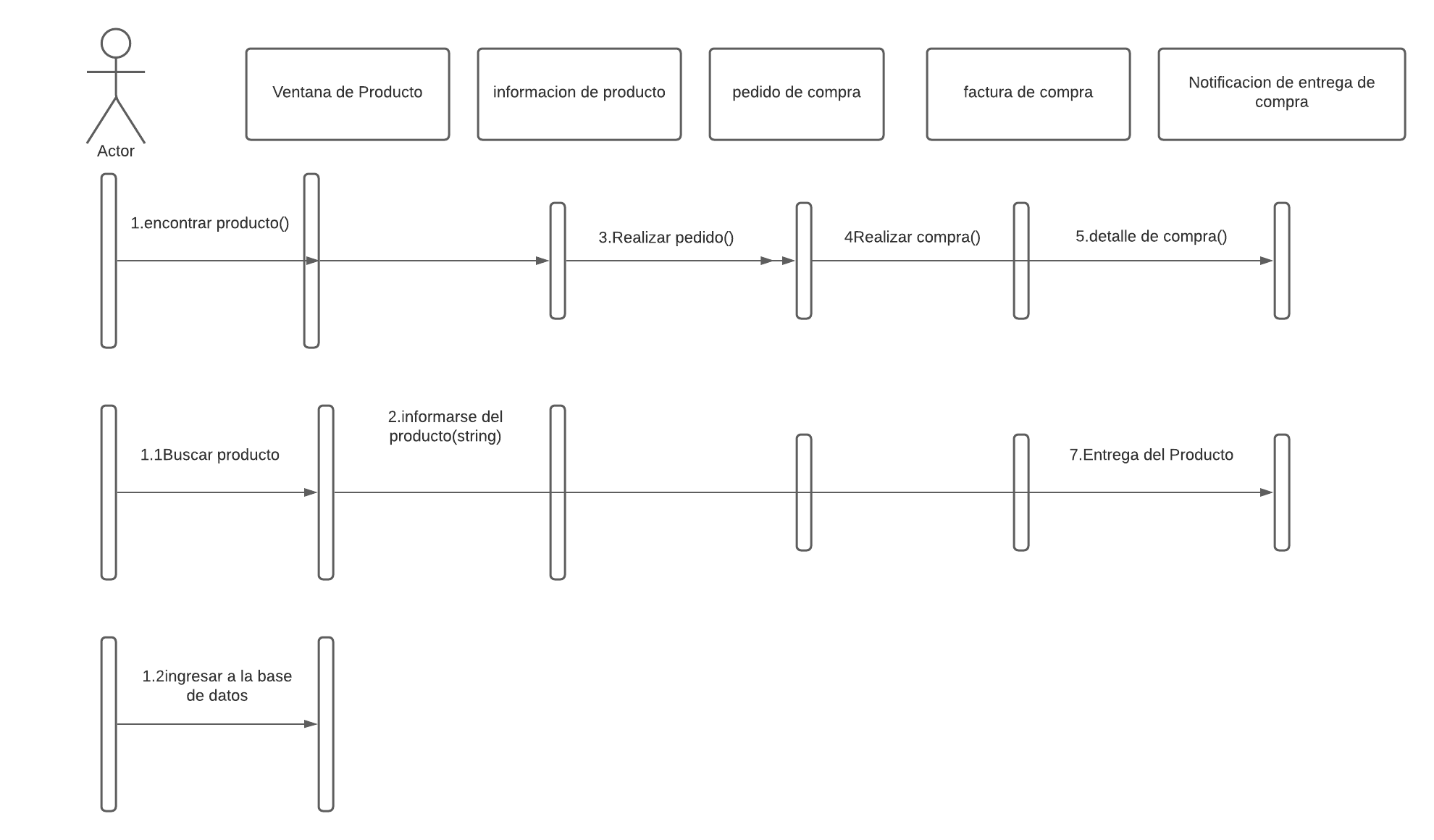


Imagen creación propia

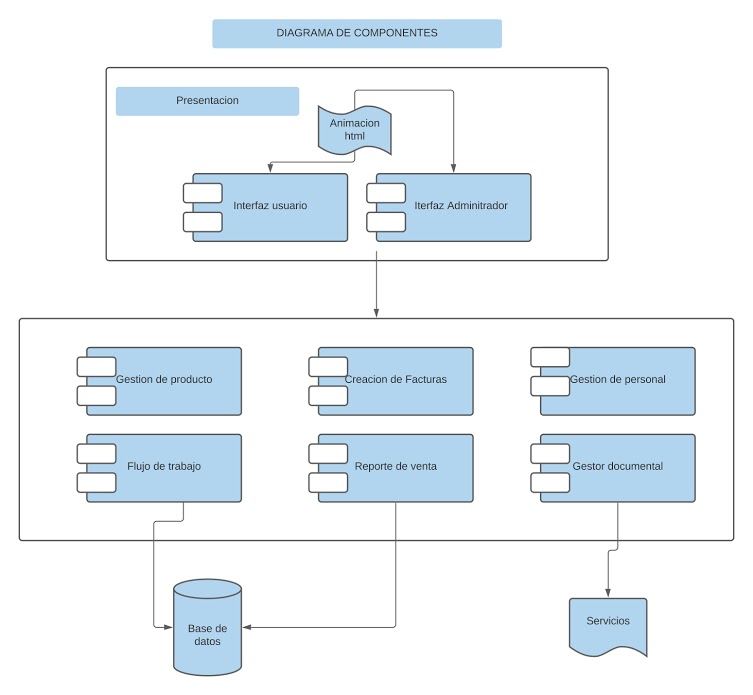
## Arquitectura del sistema



## Diagrama de secuencias



## Diagrama de componentes



## Pruebas de calidad

Debido a que las pruebas garantizan el correcto funcionamiento del mismo sistema puesto que verifican la correcta integración de todos sus componentes a partir de condiciones iniciales y resultados esperados en el desarrollo del presente proyecto se aplicaron los siguientes tipos de pruebas:

* + 1. **Pruebas de unidad**

Las pruebas de unidad han sido aplicadas por módulos, de manera independiente, antes de integrarse con otros módulos, evitando de esta manera errores posteriores de mayor dimensión. Con este tipo de pruebas, se espera tener un nivel óptimo de depuración.

Inicialmente se procedió a aplicar las siguientes pruebas a todos los formularios

* **DATOS DE PRUEBA**
* Conexión con la base de datos.
* Campos Vacíos.
* Ingresar números en campos que solo aceptan letras.
* Ingresar letras en campos que solo aceptan números.
* **RESULTADOS**
* Existe conexión con la base de datos.
* Mensaje de alerta “Es obligatorio llenar el campo”.
* No se permite ingresar números en campos donde solo admiten letras.
* No se permite ingresar letras en campos numéricos.

**Objetivo**

Evaluar el funcionamiento del sistema y su nivel de aceptación en el entorno social

## Documentación de la prueba del prototipo

Las pruebas de calidad en un [**Software**](https://www.microtech.es/software-de-gestion-erp)  son todos aquellos procesos cuya ejecución permiten conocer la calidad del mismo, así como los posibles fallos que puedan existir a corto, medio o largo plazo. Cuando se realizan pruebas en los **softwares de gestión empresarial** es posible predecir su comportamiento durante la implantación, su grado de manejabilidad y su interfaz gráfica.

**Existen diferentes maneras para realizar las pruebas de calidad**, las mismas van dadas por el contexto que representa cada uno de los clientes en particular, en otras palabras, no hay un plan de prueba que pueda servir para todos los escenarios, porque puede ser que una prueba para un [software ERP](https://www.microtech.es/software-de-gestion-erp) específico sea perfecta, pero en otro puede llegar a ser perjudicial.

**TIPOS DE PRUEBAS DE CALIDAD DE SOFTWARE**

Todas las pruebas de este tipo varían, no obstante, su finalidad sigue siendo la misma y es el hecho de comprobar y dar el visto bueno a todo el software de una empresa, sin importar que se trate de un [ERP](https://www.microtech.es/software-de-gestion-erp)**para empresas instaladoras** o para **empresas distribuidoras.** A continuación, se explican algunas pruebas de calidad:

**PRUEBAS DE FUNCIONAMIENTO:** Dentro de esta categoría es posible encontrar tres subgrupos, los cuales se clasifican, según sus objetivos, en:

**Testing de función:** Enfocada en conseguir informes sobre la aptitud de las funciones, los métodos y los servicios que componen el [ERP](https://www.microtech.es/software-de-gestion-erp).

**Testing de seguridad:** Ideada para comprobar la seguridad de los datos que maneja el [ERP](https://www.microtech.es/software-de-gestion-erp) para Pymes, es decir, los datos de la empresa.

**Testing de volumen:**Dirigida a medir la velocidad del [ERP](https://www.microtech.es/software-de-gestion-erp) para procesar grandes cantidades de información en la base de datos, bien sea en entrada o en salida.

**PRUEBAS DE USABILIDAD:** La finalidad de éstas es comprobar la relación del usuario con el software de gestión empresarial, prestando especial atención a la interfaz, la experiencia del usuario, asistentes de ayuda integrados, interactividad, entre otras cosas.

**PRUEBAS DE FIABILIDAD:** Al igual que las de funcionamiento, se dividen en tres grupos:

**Testing de integridad:** Trata de valorar y calificar la resistencia a fallos que presenta el software.

**Testing de estructura:** Consiste en verificar que cada una de las partes que componen el diseño del [software ERP](https://www.microtech.es/software-de-gestion-erp) se encuentren conectadas a las funciones correctas, en otras palabras, que no exista contenido huérfano en el [ERP](https://www.microtech.es/software-de-gestion-erp).

**Testing de estrés:** Se utiliza para evaluar el comportamiento del [software ERP](https://www.microtech.es/software-de-gestion-erp) en situaciones poco usuales, como por ejemplo, una sobre carga de funciones, la saturación de la memoria, la restricción de los recursos compartidos, etc…

**PRUEBAS DE RENDIMIENTO:** Miden la relación función resultado de un [ERP](https://www.microtech.es/software-de-gestion-erp), éstas se dividen en:

**Testing Benchamark:** Se encarga de cuantificar el rendimiento de un nuevo componente del [ERP](https://www.microtech.es/software-de-gestion-erp), comparándolo directamente con otro existente.

**Testing de contención**: Ésta valora la capacidad de respuesta de un elemento, en cuanto a sus habilidades, al momento de ser requerido por diversos actores, un ejemplo de dicho elemento podría ser el registro de recursos o la memoria.

**Testing de carga:** Su finalidad es probar y asegurar los límites operacionales del software de gestión empresarial, mediante la aplicación de cargas de trabajo promedio. Por ejemplo, si se quiere hacer la prueba sobre un [ERP](https://www.microtech.es/software-de-gestion-erp)**para una empresa de obras y servicios**, es necesario indicar cuales son las cargas habituales de trabajo de las mismas.

**PRUEBAS PARA LA CAPACIDAD DE SOPORTE:** Existen formas de medirlo, éstas son:

**Testing de configuración:** Su principal objetivo es asegurarse que **un**[ERP](https://www.microtech.es/software-de-gestion-erp)**funciona de forma correcta**, sin importar la configuración que la empresa decida aplicar, tanto a nivel de software como a nivel hardware. De igual forma es posible aplicar esta prueba como un medidor de rendimiento del sistema, o bien utilizarla en asociación con una prueba especializada en dicho tema.

(microtech, 2020)

**RESULTADOS ESPERADOS**

**los resultados esperados fueron exitosos se logró hacer un sistema donde se pueda REGISTRAR TODOS los registros de ventas por vendedor y el registro de cada producto para poder tener un sistema web confiable.**

**también se desarrolló un sistema de registro de productos al igual que un módulo de reportes imprimiendo un comprobante de venta.**

**CONCLUSIONES**

**las metodologías de análisis y diseño de sistemas puede ser mejor aprovechadas cuando son utilizadas en conjunto ya que cada una cuenta con herramientas que permiten identificar requerimientos analizar el PROBLEMA, Y diseñar la solución de acuerdo a las preferencias y comodidad de cada desarrollador.**

**la utilización de software que facilite la programación de sistemas web da la posibilidad de crear una INTERFAZ QUE permite al usuario el fácil uso y navegabilidad además facilita la tarea del desarrollador.**

**los agentes inteligentes tienen el potencial para automatizar una variedad muy grande de ACTIVIDADES DE la vida diaria además de otras más específicas que permitan a las personas usar más el tiempo en analizar los resultados y tomar decisiones basados en estos que en hacer la investigación y recopilación de datos.**

**el sistema ha logrado un mejor control de los productos vendidos y con un comprobante de venta por lo que se ha concluido satisfactoriamente con este objetivo.**

**se lleva un registro COMPLETO Y ordenado de cada venta lo cual ha disminuido la desorganización de la INFORMACIÓN Y los reportes generados son de gran ayuda al momento de tomar decisiones sobre las inversiones de la empresa.**

**RECOMENDACIONES**

**en cuanto a la empresa en general podemos recomendar que en el área de contabilidad se implemente un sistema contable puesto que con este módulo aún mantiene el proceso manual de la información teniendo dificultades de orden y efectividad.**

**un sistema de control de asistencia de personal dado que los empleados realizan su ingreso sin control causando ineficiencia a la empresa.**

**se recomienda la actualización y mantenimiento del sistema implantado esto para un correcto funcionamiento y evitar sorpresivas fallas en el futuro.**

**realizar un backup de toda la información almacenada por lo menos al final de cada mes para evitar problemas jurídicos puesto que se manejan facturas que son prueba de una transacción monetaria.**

**FUENTES DE INFORMACIÓN BIBLIOGRÁFICA**

# JSON: ¿Qué es y para qué sirve? https://www.nextu.com/blog/que-es-json/ (Barrera, 2020)

COMPOSER https://styde.net/composer-manejador-paquetes-php/

(Barrera, 2020)

Diagrama de casos de Usos https://es.wikipedia.org/wiki/Diagrama\_de\_casos\_de\_uso

(WIKIPEDIA, n.d.)

Diagrama de Secuencias https://www.lucidchart.com/pages/es/diagrama-de-secuencia

(lucidchart, n.d.)

Comenzando con chart.js introducción <https://code.tutsplus.com/es/tutorials/getting-started-with-chartjs-introduction--cms-28278> (envatotuts, n.d.)

***TIPOS DE PRUEBAS DE CALIDAD DE SOFTWARE*** [*https://www.ecured.cu/Pruebas\_de\_Calidad\_de\_Software*](https://www.ecured.cu/Pruebas_de_Calidad_de_Software)(ECURED, 2020)

***ARQUITECTURA 3 CAPAS*** [*https://www.ecured.cu/Arquitectura\_de\_tres\_niveles*](https://www.ecured.cu/Arquitectura_de_tres_niveles)

(ECURED, n.d.)

***Arquitectura Modelo vista controlador(MVC)*** [*https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador*](https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador)(Wikipedia, 2020)

***Programación orientada a objetos*** [*https://core.ac.uk/download/pdf/16357847.pdf*](https://core.ac.uk/download/pdf/16357847.pdf)(CCIA, 2006)

***ARQUITECTURA DE UN SISTEMA WEB*** [*https://es.wikipedia.org/wiki/Arquitectura\_de\_software*](https://es.wikipedia.org/wiki/Arquitectura_de_software)(WIKIPEDIA, WIKIPEDIA, n.d.)

**ANEXOS**