**Diseño de Arquitectura para posterior implementación de una Aplicación Web para venta de productos.**

**Integrantes: Felipe Duve**

**Francisco Espinoza**

Contenido

[1. Introducción 3](#_Toc7556924)

[2. Problemática 4](#_Toc7556925)

[2.1 Definición del problema. 4](#_Toc7556926)

[2.2 Solución Propuesta. 5](#_Toc7556927)

[2.3 Propósitos. 6](#_Toc7556928)

[2.4 Objetivos. (Principal y Específicos) 6](#_Toc7556929)

[3. Definición de Requerimientos. 7](#_Toc7556930)

[4. Análisis de Requerimientos. 8](#_Toc7556931)

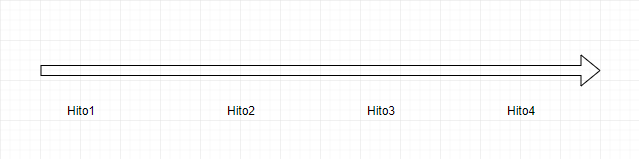
[5. Plan de trabajo. 10](#_Toc7556932)

[6. Componentes técnicos de la solución. 11](#_Toc7556933)

[7. Conclusiones. 11](#_Toc7556934)

# Introducción

Se acerca un vendedor de productos para encontrar una solución a su problema al momento de entregar sus productos, ya que al llegar y ofrecer el producto para realizar una venta, a menudo le rechazan la compra debido a que no están de acuerdo, debido a esto, cree que la solución es la implementación de una aplicación web que ofrezca productos y permita registro para comprar. Para mejorar el sistema de ventas del sujeto, se requiere hacer una aplicación web que muestre los productos/ítem, que permita a los usuarios registrarse para la aplicación, y que les permita usar una función de shop-cart, o carrito de compras para que, de ésta forma, los usuarios paguen, y sólo debe preocuparse de que el cliente obtenga su producto, ya sea por envío o por entrega.



Roadmap sobre los 4 hitos.

Se le explica al vendedor que la metodología de trabajo consta de cuatro entregas, siendo la primera una presentación sobre qué es lo que se realizará (hito1), la segunda la entrega consta de una página web que muestra los productos en la base de datos (hito2), la tercera consta de agregar la funcionalidad de creación de cuentas por los usuarios (hito3), y la cuarta consta de agrgar la funcionalidad de shop-cart, o “carrito de compras”, para que los usuarios puedan seleccionar productos y comprarlos.

# Problemática

# 2.1 Definición del problema.

El problema que presenta el sistema actual del vendedor, es que, a pesar de poseer páginas dedicas en R.R.S.S. para subir fotos de los productos, además de su descripción, precio y otros datos, es que al llegar a hacer el trato en físico con el cliente, éste puede retractarse y no comprar el producto, lo que ocasiona pérdida de tiempo, posibilidad de peligros para el vendedor, por ejemplo ser robado por el cliente, además de perder tiempo dando descripciones extra a los clientes vía R.R.S.S debido a que éste no entiende bien alguna característica del producto, ya sea precio, modo de pago o descripción general.

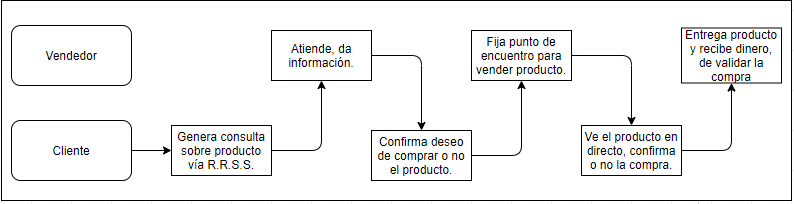
El proceso de venta actual es como continúa:

Diagrama de procesos de la tienda, previo a aplicación web.

# 2.2 Solución Propuesta.

La solución propuesta, automatiza la mayor parte de las acciones que debe realizar el vendedor, de modo que solo debe hacer entrega del producto, respecto a posibles amenazas, quedan reducidas, ya que el dinero será receptado de manera digital, y no física, esperando a que el cliente se lo facilite.

En el modelo que se ofrece, como se aprecia, el vendedor se desliga de todo trabajo, automatizando el 80% de las acciones referentes a concretar una venta, teniendo en cuenta las cinco acciones a las que él se encontraba relacionado anteriormente.

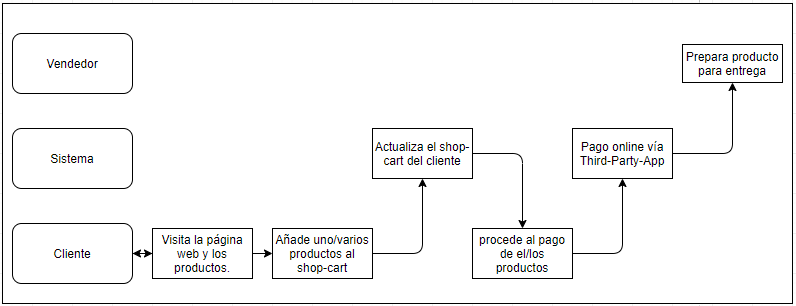
El proceso de venta, quedaría como sigue:

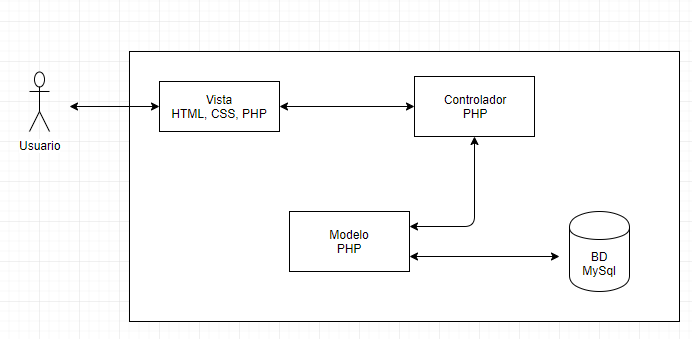
Diagrama de procesos de la tienda, después de la implementación de la aplicación web.

# 2.3 Propósitos.

La aplicación web cumple con el propósito de automatizar la mayor parte del proceso. Para ésto se trabajará en un diseño MVC, orientado completamente a la interacción del usuario (es decir, el cliente que desea comprar en la tienda) con el sistema.

# 2.4 Objetivos. (Principal y Específicos)

El objetivo principal está basado en la automatización de un proceso de compra, con la intención de que el usuario o cliente, en este caso, tenga una mejor experiencia al momento de realizar una compra. La idea de ésto es beneficiar al cliente con el ahorro de tiempo ya que el proceso será más eficiente.

Esquema de Arquitectura objetivo.

# Definición de Requerimientos.

Los requerimientos, se pueden dividir para los dos posibles usuarios, cliente y vendedor, como sigue:

* Vendedor: en sus palabras, solicita una página, donde se muestren sus productos, el cliente se pueda registrar, y pueda utilizar un carrito de compras para completar la compra y pago del producto.
* Cliente: en sus palabras, solicita que los productos se vean bien, con claridad, y ordenados, además de que el carrito de compras le muestre el total a pagar, y posibles descuentos.

Se muestra con la técnica de historias de usuario:

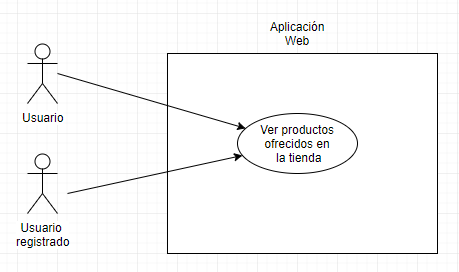
1. **Como** vendedor, **quiero** una página web que muestre los productos que ingreso, **para que** el cliente pueda observarlos.
2. **Como** vendedor, **quiero** que el cliente se pueda registrar, **para** obtener información y quizás ofrecerles descuentos, promociones u otros.
3. **Como** vendedor, **quiero** que el cliente utilice un carrito de compras, **para que**, con claridad verifique lo que desea comprar, antes de pagarlo.
4. **Como** cliente, **quiero** ver los productos que ofrece la tienda, **para así**, poder decidir sobre si me interesa o no.
5. **Como** cliente, **quiero** que el carrito de compras me muestra la información previa al pago, **para así** poder confirmar o no el pago para adquirir los productos.

# Análisis de Requerimientos.

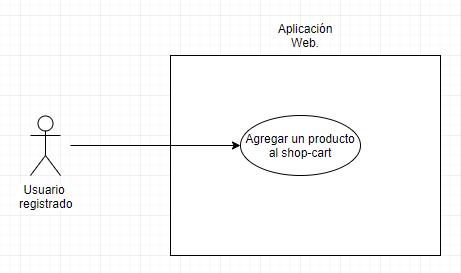
Los actores del sistema, son:

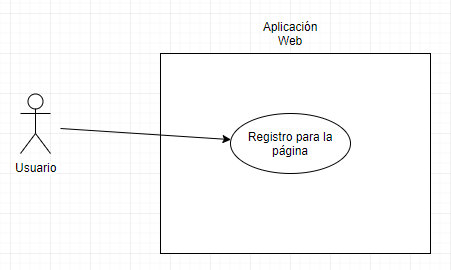
* Usuario: es el usuario que entra “por primera vez”, o sin registrarse, éste actor no podría realizar compras, debido a que al registrarse, añade su dirección, con la que, posiblemente se deba realizar un envío por parte del actor *vendedor*.
* Usuario registrado: es el usuario sobre el cual ya se tienen los datos, él puede realizar compras en la tienda, ya que sus datos están guardados por parte de la aplicación.

Diagramas de casos de uso:



1. Yo, como usuario (registrado o no) deseo ver los productos ofrecidos.



1. Yo, como usuario registrado, deseo agregar un producto al shop-cart.
2. Yo como usuario, deseo registrarme en la aplicación.
3. El usuario, esté registrado o no, quiere acceso a visualizar los productos que se ofrecen en la tienda.
4. El usuario registrado tiene acceso a agregar productos a un shop-cart, el punto a favor del shop-cart, es que el usuario puede decidir, al ver los datos del total, si comprarlo o no.
5. El usuario, quiere registrarse en la página, ésto le acceso al uso del shop-cart y poder comprar los productos ofrecidos.

# Plan de trabajo.

Equipo de Trabajo

El equipo se compone de dos desarrolladores, Francisco Espinoza, Felipe Duve, cuya disponibilidad de tiempo para la implementación del proyecto es de 6 horas semanales.

Programación

|  |  |  |  |
| --- | --- | --- | --- |
| Tarea a realizar | Tecnologías a usar | Horas estimadas | Semanas |
| F-E primitivo | Html, VSC, PHP | 8 | 2 |
| B-E parcial | PHP, VSC | 8 | 2 |
| Base de datos | Mysql | 2 | 2 |

Hito01, primera entrega.

|  |  |  |  |
| --- | --- | --- | --- |
| Tarea a realizar | Tecnologías a usar | Horas estimadas | Semanas |
| Feedback | Html, VSC, PHP, Mysql, Css | 6 | 2 |
| F-E mejorado | Html, VSC, PHP | 12 | 2 |
| B-E parcial | PHP, VSC | 4 | 2 |
| Base de datos | Mysql | 2 | 2 |

Hito02, segunda entrega.

|  |  |  |  |
| --- | --- | --- | --- |
| Tarea a realizar | Tecnologías a usar | Horas estimadas | Semanas |
| Feedback | Html, VSC, PHP, Mysql, Css | 6 | 3 |
| F-E primitivo | Html, VSC, PHP, Css | 12 | 3 |
| B-E parcial | PHP, VSC | 4 | 3 |
| Base de datos | Mysql | 2 | 3 |

Hito03, tercera entrega.

1. Hito01, para la primera entrega, vista para el día 14 mayo, se hace entrega de un sistema que muestra los productos que se han ingresado, funcional en lo mencionado, cumpliendo con el *caso de uso 01*.
2. Hito02, para la segunda entrega, vista para el día 11 junio, se hace entrega de un sistema que permite a un usuario registrarse para la aplicación, funcional en lo mencionado, cumpliendo con el *caso de uso 03*.
3. Hito03, para la tercera entrega, vista para el día 02 julio, se hace entrega del sistema que permite a los usuarios agregar productos a un shop-cart, funcional en lo mencionado, cumpliendo con el *caso de uso 02*.

# Componentes técnicos de la solución.

Para abordar la implementación, se planea utilizar la siguiente tecnología:

* CodeIgniter: Framework que facilita la orientación a objetos, se trabaja con PHP, además de tener la cualidad de ser muy ordenado, lo que facilita su uso y modificación, en caso de haber necesidad. La posible modificación toma peso al recibir feedback, y corrobar, por parte del “vendedor, o cliente que solicita la aplicación”, cuales puntos se deben modificar debido a que no cumple sus espectativas.
* PHP: lenguaje de lado del servidor, ya que realiza todas las acciones que, desde el lado del usuario, se cataloguen como interacciones, y las realiza a modo de consulta, es decir, PHP es el encargado de soportar los modelos y controladores dentro del Model-View-Controller. Tiene además el atractivo de poder ser incrustado en trozos de código de HTML (embebbed code).
* Mysql: Lenguaje utilizado para realizar consultas, mysql es un gestor de bases de datos, y en ésta tecnología es donde se almacenará la información de los clientes, productos, y shop-cart.
* Tecnologías de vista: son el conjunto de componentes que tienen que ver exclusivamente con lo que interactuará el usuario, se utiliza HTML para colocar texto plano, que tomará forma y diseño estéticamente agraciado con CSS, que en ésta oportunidad se trabajará con el framework Bootstrap.
* VSC: un extra, es que el equipo de trabajo utiliza de preferencia personal Visual Code Studio como editor de texto.

# Conclusiones.

Los riesgos posibles incluyen la falta de experiencia previa por parte del equipo para la implementación de tecnologías en el front-end, es decir, en la parte visual al usuario, de interacción, por lo mismo, para mitigar riesgos es que se ha decidido invertir más horas en éste apartado.