

**DEPARTAMENTO DE**

**CIENCIAS DE LA COMPUTACIÓN**

**INGENIERÍA EN**

**SISTEMAS E INFORMÁTICA**

*SEXTO NIVEL*

*“GESTION DE PEDIDOS PARA EL RESTAURANTE JENGA”.*

**Integrantes:**

Maria Belen Ceron

Alex Chicaiza

Elian Llorente

**Tutor:**

Ing. Jenny Ruiz

Sangolquí, 2021

**Historial de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 08/09/2021 | 1.0 | Revisión documental  Capítulo 1 | Maria Belen Ceron  Alex Chicaiza  Elian Llorente |

Contenido

[RESUMEN EJECUTIVO 4](#_Toc81418045)

[INTRODUCCIÓN 5](#_Toc81418046)

[CAPÍTULO I 6](#_Toc81418047)

[1.1 Título del Proyecto 6](#_Toc81418048)

[1.2. Sistema de Objetivos 6](#_Toc81418049)

[1.2.1 Objetivo General 6](#_Toc81418050)

[1.2.2 Objetivos Específicos 6](#_Toc81418051)

[1.3 Alcance 6](#_Toc81418052)

[1.4 Definición y Justificación del Problema 7](#_Toc81418053)

[1.5 Presupuesto 8](#_Toc81418054)

[Hardware 8](#_Toc81418055)

[Software 8](#_Toc81418056)

[CAPÍTULO II 8](#_Toc81418057)

[2.1 Modelamiento del Negocio y sus Entregables 8](#_Toc81418058)

[2.1.1 Documentos de Caso de Uso Historias de Usuario) 8](#_Toc81418059)

[2.2 Definición de Requerimientos 10](#_Toc81418060)

[2.2.1 Especificación de Requerimientos de Software 10](#_Toc81418061)

[2.2.2 Especificación de Casos de Uso 11](#_Toc81418062)

[2.3 Análisis y Diseño 13](#_Toc81418063)

[2.3.1 Modelo Conceptual 13](#_Toc81418064)

[2.3.2 Modelo Lógico 14](#_Toc81418065)

[2.3.3 Modelo Físico 15](#_Toc81418066)

[2.3.4 Script de la Base de Datos 15](#_Toc81418067)

[2.4 Implementación 20](#_Toc81418068)

[2.4.1 Modelo de Arquitectura 20](#_Toc81418069)

[CAPÍTULO III 21](#_Toc81418070)

[3.1 Pruebas 21](#_Toc81418071)

[3.1.1 Prueba de Caja Negra 21](#_Toc81418072)

[3.1.2 Prueba de Caja Blanca 21](#_Toc81418073)

[3.1.3 Documentación de Informe de Errores 21](#_Toc81418074)

[3.1.4 Documentación Técnicas Caja Blanca y Caja Negra 21](#_Toc81418075)

[CAPÍTULO IV 4](#_Toc81418076)

[4.1 Conlcusiones 4](#_Toc81418077)

[4.2 Recomendaciones. 4](#_Toc81418078)

# RESUMEN EJECUTIVO

El presente proyecto tiene como finalidad presentar el análisis y diseño de un software que gestionara pedidos a domicilio para el restaurante Jenga, se ha tomado los requisitos necesarios en base a las necesidades del usuario y se presentara la documentación completa, junto al software con todo lo requerido, es decir, el usuario podrá comprender de manera sencilla el uso y funcionalidad del aplicativo porque se ha trabajado con una metodología ágil, la cual mantuvo al usuario al tanto del desarrollo del software y la documentación esta bien explicada para un fácil entendimiento.

# INTRODUCCIÓN

Las cadenas de restaurantes cuentan con alto número de consumidores que deben ir al establecimiento para conocer sus productos, precios y promociones disponibles y hacer largas colas para ser atendidos, lo que causa en los clientes molestias e inconformidad por la pérdida de tiempo en la espera e inclusive en algunos casos, conocer que los productos deseados no estén disponibles, provocando pérdidas de clientes en lugar de aumentar la comercialización de sus productos, y esto genera que el negocio no tenga una alta competitividad con las grandes franquicias de comida, que incluso manejan el servicio a domicilio, por esta razón se buscan estrategias tecnológicas como el uso de software para el control de ventas o pedidos.

En este sentido se pretende mejorar el servicio al cliente del restaurante Jenga que permitirá a los clientes ver el menú y realizar sus pedidos con toda seguridad desde la comodidad de su hogar o en el lugar que se encuentre mediante el uso de una aplicación web.

# CAPÍTULO I

## Título del Proyecto

Gestión de pedidos para el restaurante Jenga

## 1.2. Sistema de Objetivos

### 1.2.1 Objetivo General

Realizar un análisis y diseño de software para automatizar la promoción del restaurante Jenga mediante un aplicativo web.

### 1.2.2 Objetivos Específicos

* Realizar la matriz HU de historias de usuarios.
* Elaborar los entregables, basados en las métricas V3, ASI, DSI y OO, de la etapa de análisis y diseño, el establecimiento de requisitos, análisis de clases, de casos de uso, diseño de clases y diseño físico de datos.
* Realizar casos de prueba y reporte de errores.

## 1.3 Alcance

Las fases del proceso de Análisis y Diseño de Software que permitirá desarrollar el proyecto serán las siguientes:

* **4.1. Etapa de análisis:** Definir claramente el problema que se desea resolver o el sistema que se desea crear. Identificar los componentes principales que integrarán el sistema.
* **4.2. Etapa de Diseño:** En este proceso se utiliza la información recolectada en la etapa anterior y la principal tarea a desarrollar es un modelo o las especificaciones del sistema.
* **4.3. Etapa de Desarrollo:** En este caso se plantea utilizar los modelos creados durante la etapa de diseño de manera que sean solo objetos de prueba.

Debido a que el restaurante no cuenta con un apoyo de promoción y comercialización de productos, realizar una aplicación web, para modernizar el servicio que evitará a los clientes ir personalmente al restaurante o realizar llamadas al dueño del negocio para conocer promociones, costos y productos disponibles, dicha información actualizada y debidamente estructurada se brindará al acceder desde un dispositivo móvil a la aplicación web sin importar el sitio en que se encuentre.

✔ Gestionar promociones

o Gestionar promociones de pedidos mayor de 30 dólares se obsequiará 2 cervezas artesanales ($3.50), no es acumulable.

o Por cada 5 pizzas familiares, en el siguiente pedido se obsequiará 1 pizza familiar.

* **4.4. Etapa de Pruebas:** Consiste en asegurar que los componentes individuales que integran al sistema o producto cumplen con los requerimientos de las especificaciones (ERS)

## 1.4 Definición y Justificación del Problema

El proyecto se plantea desarrollar mediante la toma de requisitos de la gestión de un servicio de comidas a domicilio y en base a los requisitos funcionales, realizar una matriz del marco de trabajo de HU para identificación de requisitos funcionales, usuarios que se generan de manera automatizada en base a los datos ingresados, además generar un reporte errores de casos de prueba que permita ver la factibilidad del proyecto y si este satisface con nuestras necesidades.

Cuando se habla de análisis y diseño de software, se debe tomar en cuenta varios aspectos que son de gran importancia, esta propuesta busca orientar a otros investigadores sobre cómo se debe plantear un correcto análisis y diseño, utilizando e implementando varias herramientas que beneficien a este proceso

## 

## 1.5 Presupuesto

### Hardware

|  |  |  |  |
| --- | --- | --- | --- |
| **Cantidad** | **Descripción** | **Valor Unitario (USD)** | **Valor Total (USD)** |
| 1 | Dell Lenovo Hp Intel Core I3 10ma Gen | 579 | 579 |
| 3 | Computadores Personales | 0.00 | 0.00 |
| **Total** | | | 579 |

### Software

|  |  |  |  |
| --- | --- | --- | --- |
| **Cantidad** | **Descripción** | **Valor Unitario (USD)** | **Valor Total (USD)** |
| 3 | Power Design | 4.68 | 4.68 |
| 1 | SQL Server (Developer) | 0.00 | 0.00 |
| 1 | SQL Server (Standard: servidor) | 889.00 | 889.00 |
| 1 | Sistema Operativo Windows 10 | 289.00 | 289.00 |
| 1 | Visual Studio Code | 0.00 | 0.00 |
| 1 | Lucid | 0.00 | 0.00 |
| 3 | Suite de Office 365 | 0.00 | 0.00 |
| **Total** | | | 1,182.68 |

# CAPÍTULO II

## 2.1 Modelamiento del Negocio y sus Entregables

### 2.1.1 Documentos de Caso de Uso Historias de Usuario

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

## 2.2 Definición de Requerimientos

### 2.2.1 Especificación de Requerimientos de Software

#### Requisitos Funcionales

**Requisito funcional 1**

|  |  |
| --- | --- |
| Id. Requerimiento | REQ01 |
| Nombre | Agregar productos al carrito de compras |
| Actor | Usuario |
| Descripción | Añadir productos al carrito de compras y que se queden hasta que se decida realizar o no la compra. |
| Entradas | Cantidad de productos |
| Salidas | Interfaz del Sistema:.   * Aumentar la cantidad de productos   Añadir al carrito |
| Proceso | 1. Agregar la cantidad del producto a comprar.  2. Dar clic en el botón "Añadir al carrito" |
| Precondiciones | Verificar que en el carrito de compras se haya añadido de manera correcta el producto seleccionado (Producto, cantidad,precio unitario, precio total) y además el uso de pruebas unitarias |
| Post condiciones | Realizar la compra |
| Efectos Colaterales | Si no hay nada en el carrito de compras no se puede realizar ninguna compra |
| Prioridad | Alta |

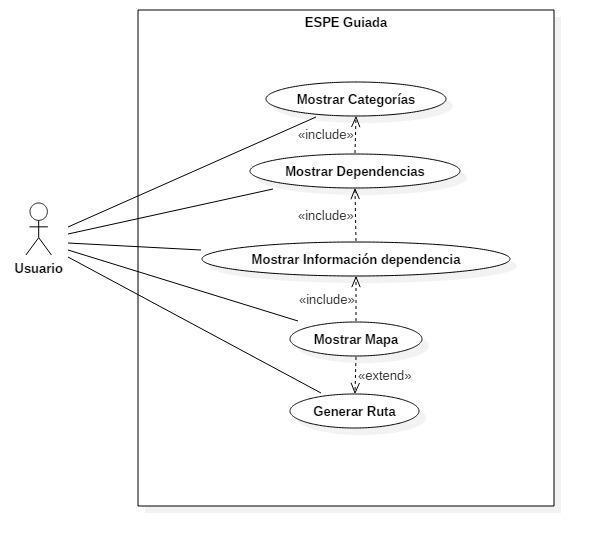
**Requisito funcional 2**

|  |  |
| --- | --- |
| Id. Requerimiento | REQ02 |
| Nombre | Eliminar productos del carrito |
| Autor | Usuario |
| Descripción | El aplicativo deberá permitir eliminar productos que no deseados |
| Entradas | Quitar productos del carrito |
| Salidas | Interfaz del Sistema:   * Disminuir la cantidad del producto   Quitar del carrito |
| Proceso | 1. Dar clic en el botón "Eliminar"  2. Dar clic en "Aceptar" del cuadro de diálogo que se despliegue para conocer que en realidad se desea eliminar el producto o fue equivocación |
| Precondiciones | Verificar que en el carrito de compras ya no esté el producto que se eliminó y usar pruebas unitarias |
| Post condiciones | Eliminar productos del carrito |
| Efectos Colaterales | Si no hay nada en el carrito no se puede quitar productos |
| Prioridad | Alta |

**Requisito funcional 3**

|  |  |
| --- | --- |
| Id. Requerimiento | REQ03 |
| Nombre | Registrar el total de pago productos |
| Actor | Usuario |
| Descripción | Pagar productos |
| Entradas | Los productos del carrito de compras |
| Salidas | Interfaz del Sistema:  Vista del total de pago |
| Proceso | 1. Dar clic en el visualizar pago.  2. Click en "Aceptar" (cuadro de información que muestra el total en dólares). |
| Precondiciones | Tener productos dentro del carrito de compras |
| Post condiciones | Compra realizada |
| Efectos Colaterales | Verificar que el total de pago esté calculado de forma correcta de acuerdo con los productos seleccionados con uso de pruebas unitarias. |
| Prioridad | Alta |

### 2.2.2 Especificación de Casos de Uso



## 2.3 Análisis y Diseño

### 2.3.1 Modelo Conceptual

Diagrama

Descripción generada automáticamente

### 2.3.2 Modelo Lógico

Diagrama

Descripción generada automáticamente

### Diagrama Descripción generada automáticamente2.3.3 Modelo Físico

### 2.3.4 Script de la Base de Datos

--*phpMyAdmin SQL Dump*

--*version 5.0.2*

--*https://www.phpmyadmin.net/*

--

--*Servidor: 127.0.0.1*

--*Tiempo de generación: 02-09-2021 a las 02:15:26*

--*Versión del servidor: 10.4.14-MariaDB*

--*Versión de PHP: 7.2.33*

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\**!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT*\*/;

/\**!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS*\*/;

/\**!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION*\*/;

/\**!40101 SET NAMES utf8mb4*\*/;

--

--*Base de datos: `restaurantejenga`*

--

--*--------------------------------------------------------*

--

--*Estructura de tabla para la tabla `cliente`*

--

CREATE TABLE `cliente` (

  `CodigoCliente` int(8) NOT NULL,

  `NombreCliente` varchar(32) NOT NULL,

  `TelefonoCliente` varchar(16) NOT NULL,

  `DireccionCliente` varchar(32) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

--*Volcado de datos para la tabla `cliente`*

--

INSERT INTO `cliente` (`CodigoCliente`, `NombreCliente`, `TelefonoCliente`, `DireccionCliente`) VALUES

(1, 'Maria Belen Ceron', '0997702037', 'Playa Chica'),

(2, 'Elian Llorente', '0998878765', 'Conocoto');

--*--------------------------------------------------------*

--

--*Estructura de tabla para la tabla `detallepedido`*

--

CREATE TABLE `detallepedido` (

  `CodigoDetallePedido` int(8) NOT NULL,

  `Codigo\_Pedido` int(8) NOT NULL,

  `Codigo\_Cliente` int(8) NOT NULL,

  `Codigo\_Producto` int(8) NOT NULL,

  `CantidadDetallePedido` varchar(8) NOT NULL,

  `TotalDetallePedido` varchar(16) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

--*Volcado de datos para la tabla `detallepedido`*

--

INSERT INTO `detallepedido` (`CodigoDetallePedido`, `Codigo\_Pedido`, `Codigo\_Cliente`, `Codigo\_Producto`, `CantidadDetallePedido`, `TotalDetallePedido`) VALUES

(1, 1, 1, 1, '1', '15'),

(2, 1, 1, 4, '2', '6');

--*--------------------------------------------------------*

--

--*Estructura de tabla para la tabla `pedido`*

--

CREATE TABLE `pedido` (

  `CodigoPedido` int(8) NOT NULL,

  `Codigo\_Cliente` int(8) NOT NULL,

  `FechaPedido` varchar(16) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

--*Volcado de datos para la tabla `pedido`*

--

INSERT INTO `pedido` (`CodigoPedido`, `Codigo\_Cliente`, `FechaPedido`) VALUES

(1, 1, '30/8/2021');

--*--------------------------------------------------------*

--

--*Estructura de tabla para la tabla `producto`*

--

CREATE TABLE `producto` (

  `CodigoProducto` int(8) NOT NULL,

  `NombreProducto` varchar(32) NOT NULL,

  `DetalleProducto` varchar(64) NOT NULL,

  `PrecioProducto` varchar(16) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

--*Volcado de datos para la tabla `producto`*

--

INSERT INTO `producto` (`CodigoProducto`, `NombreProducto`, `DetalleProducto`, `PrecioProducto`) VALUES

(1, 'Pizza Tradicional', 'Salsa de la casa, queso mozzarella, salami, pimientos y champiño', '15'),

(2, 'Pizza Hawaiana', 'Salsa de la casa, queso mozzarella, jamón y piña', '15'),

(3, 'Lasaña de res', 'Salsa de la casa, pasta fresca artesanal, salsa boloñesa, salsa ', '4'),

(4, 'Nachos', 'Nachos con carne molida y queso mozzarella', '3');

--

--*Índices para tablas volcadas*

--

--

--*Indices de la tabla `cliente`*

--

ALTER TABLE `cliente`

  ADD PRIMARY KEY (`CodigoCliente`);

--

--*Indices de la tabla `detallepedido`*

--

ALTER TABLE `detallepedido`

  ADD PRIMARY KEY (`CodigoDetallePedido`),

  ADD KEY `Codigo\_Pedido` (`Codigo\_Pedido`),

  ADD KEY `Codigo\_Cliente` (`Codigo\_Cliente`),

  ADD KEY `Codigo\_Producto` (`Codigo\_Producto`);

--

--*Indices de la tabla `pedido`*

--

ALTER TABLE `pedido`

  ADD PRIMARY KEY (`CodigoPedido`),

  ADD KEY `Codigo\_Cliente` (`Codigo\_Cliente`);

--

--*Indices de la tabla `producto`*

--

ALTER TABLE `producto`

  ADD PRIMARY KEY (`CodigoProducto`);

--

--*AUTO\_INCREMENT de las tablas volcadas*

--

--

--*AUTO\_INCREMENT de la tabla `cliente`*

--

ALTER TABLE `cliente`

  MODIFY `CodigoCliente` int(8) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

--*AUTO\_INCREMENT de la tabla `detallepedido`*

--

ALTER TABLE `detallepedido`

  MODIFY `CodigoDetallePedido` int(8) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

--*AUTO\_INCREMENT de la tabla `pedido`*

--

ALTER TABLE `pedido`

  MODIFY `CodigoPedido` int(8) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

--*AUTO\_INCREMENT de la tabla `producto`*

--

ALTER TABLE `producto`

  MODIFY `CodigoProducto` int(8) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

--*Restricciones para tablas volcadas*

--

--

--*Filtros para la tabla `detallepedido`*

--

ALTER TABLE `detallepedido`

  ADD CONSTRAINT `detallepedido\_ibfk\_1` FOREIGN KEY (`Codigo\_Pedido`) REFERENCES `pedido` (`CodigoPedido`) ON UPDATE CASCADE,

  ADD CONSTRAINT `detallepedido\_ibfk\_2` FOREIGN KEY (`Codigo\_Cliente`) REFERENCES `cliente` (`CodigoCliente`) ON UPDATE CASCADE,

  ADD CONSTRAINT `detallepedido\_ibfk\_3` FOREIGN KEY (`Codigo\_Producto`) REFERENCES `producto` (`CodigoProducto`) ON UPDATE CASCADE;

--

--*Filtros para la tabla `pedido`*

--

ALTER TABLE `pedido`

  ADD CONSTRAINT `pedido\_ibfk\_1` FOREIGN KEY (`Codigo\_Cliente`) REFERENCES `cliente` (`CodigoCliente`) ON UPDATE CASCADE;

COMMIT;

/\**!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT*\*/;

/\**!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS*\*/;

/\**!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION*\*/;

## 2.4 Implementación

### Diagrama Descripción generada automáticamente2.4.1 Modelo de Arquitectura

# CAPÍTULO III

## 3.1 Pruebas

### 3.1.1 Prueba de Caja Negra

El resultado de las pruebas de Caja Negra realizadas al sistema se refleja en el documento adjunto “G3\_Caja\_Blanca\_Prueba.docx”.

### 3.1.2 Prueba de Caja Blanca

El resultado de las pruebas de Caja Blanca realizadas al sistema se refleja en el documento adjunto “G3\_Caja\_Blanca\_Prueba.docx”.

### 3.1.3 Documentación de Informe de Errores

**Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos que conformarán un reporte de errores de las pruebas que se aplicaron al aplicativo web: “Gestión de pedidos del restaurante Jenga”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Reporte de Errores e Inconsistencias** | | | | |
| **Nombre del Proyecto:** | | GESTION DE PEDIDOS DEL RESTAURANTE JENGA | | |
| **Fecha de pruebas:** | | 30-08-2021 | | |
| **Módulos:** | | Página principal (HOME) | | |
| **Analista:** | | Ing. Jenny Ruiz | | |
| **Responsable:** | | Andrés Zambrano | | |
| **Fecha de revisión:** | | 30-08-2021 | | |
| **Identificación Caso Prueba** | **Descripción de prueba.** | | **Descripción del error.** | **Acciones de corrección** |
| CP-001/REQ001 | Agrega productos al carrito de compras | | Al dar clic en los productos de la 2da fila, estos no se registran correctamente | Solucionar que al momento de elegir un producto se agregue correctamente al carrito de compras |

### 3.1.4 Documentación Técnicas Caja Blanca y Caja Negra

**TÉCNICA DE CAJA BLANCA:**

**FUNCIÓN AGREGAR COMIDA**

**Código fuente**

|  |
| --- |
| function agregarComida(e) {  e.preventDefault();  if (e.target.classList.contains("agregar-carrito")) {  const comidaSeleccionado = e.target.parentElement.parentElement;  leerDatosComida(comidaSeleccionado);  }  } |

**DIAGRAMA DE FLUJO**

Diagrama

Descripción generada automáticamente

Imagen 1: Diagrama de Flujo Función Agregar comida.

**GRAFO**

Diagrama, Esquemático

Descripción generada automáticamente

**RUTAS**

**R1:** 1, 2, 3, 4,5

**R2:** 1, 2, 3,5

**COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

* V(G) = número de nodos predicados(decisiones)+1

V(G)=1+1=2

* V(G) = A – N + 2

V(G)= 5– 5 + 2 = 2

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

**TÉCNICA DE CAJA NEGRA:**

**Partición de clases equivalentes**

Agregar comida al carrito

|  |  |  |
| --- | --- | --- |
| **Condición de entrada** | **Clases validas** | **Clases Invalidas** |
| Agregar-comida | (1) clic sobre el botón “Agregar al carrito” | (2) clic sobre botón distinto al de “Agregar al carrito” |

|  |  |  |  |
| --- | --- | --- | --- |
| **No Caso** | **Clase de equivalencia** | **Agregar-comida** | **Respuestas** |
| 1 | 1 | Agregar al carrito | Se muestra en el carrito de compras |
| 2 | 2 | - | No se muestra nada en el carrito de compras |

**Casos de prueba**

**Caso de prueba 1**

Interfaz de usuario gráfica

Descripción generada automáticamente

**Caso de prueba 2**

Interfaz de usuario gráfica

Descripción generada automáticamente

# CAPÍTULO IV

## 4.1 Conclusiones

Hemos visto a lo largo del desarrollo del proyecto la importancia del análisis y el diseño del software, se pudo observar un gran cambio cuando el software tiene documentación y es llevado a cabo en un proceso el cual comprueba el estado del mismo en todo momento del proyecto, además, de que se pudo realizar las pruebas debidas y confirmar si los requisitos se cumplen en relación a la documentación, entonces es ahí cuando el proyecto ha logrado demostrar como un análisis y diseño detallado ayuda a desarrollar software de manera eficiente.

## 4.2 Recomendaciones.

La importancia que tiene el análisis y diseño de sistemas de proyectos de software es fundamental ya que el análisis nos sirve para tener una mejor planificación sobre el proyecto que deseemos presentar tanto como su análisis y diseño, con ella tenemos técnicas que podemos llevar a cabo también existen reglas que tenemos que tener en cuenta para nuestro proyecto,  el diseño de sistemas se define el proceso de aplicar ciertas técnicas con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles para permitir su interpretación y realización física.