“Año de la lucha contra la corrupción e impunidad”



Escuela Académica Profesional de Ingeniería

SISTEMA DE PEDIDOS PARA EL RESTAURANTE DOÑA RUFINA

#### CURSO:

##### Proceso de software

**DOCENTE:**

Jorge Alfredo Guevara Jiménez

**INTEGRANTES:**

* Arroyo Cruzado, Stephanie
* Flores Guillen, Alexander
* Hinostroza Granados, Giancarlo Fernando
* Rodríguez Manuelo, Jhoelver
* Rojas Samaniego, Cecili

**ESCUELA ACADEMICA:**

* + Ingeniería de sistemas e informática

**SECCIÓN:** 3012

**FECHA:** 26/04/2019

Huancayo - 2019

# AGRADECIMIENTOS

A Dios todo poderoso por protegernos y cuidarnos cada día.

A nuestros padres por mostrarnos el camino hacia la superación.

A nuestros colegas por permitirnos aprender más de la vida a su lado. A cada uno de los participantes que hicieron posible el proyecto.

# DEDICATORIA

A nuestro DIOS todo poderoso.

A nuestros padres que nos han apoyado y hecho posible que este trabajo se realice.

A nuestro docente quien nos ha ofrecido sabios conocimientos para lograr nuestras metas en el presente proyecto.

A los participantes de este proyecto. Esto es posible gracias a ustedes.

# INDICE

[AGRADECIMIENTOS 2](#_Toc7711175)

[DEDICATORIA 3](#_Toc7711176)

[INDICE 4](#_Toc7711177)

[INDICE DE TABLAS 6](#_Toc7711178)

[INDICE DE ILUSTRACIONES 6](#_Toc7711179)

[CERTIFICADO DE REVISION DE REDACCION, ORTOGRAFIA Y ESTILO 7](#_Toc7711180)

[DECLARACIÓN DE AUTENTICIDAD 8](#_Toc7711181)

[RESUMEN EJECUTIVO 9](#_Toc7711182)

[ABSTRACT 10](#_Toc7711183)

[INTRODUCCIÓN 11](#_Toc7711184)

[CAPITULO I 11](#_Toc7711185)

[1.1. PLANTEAMIENTO Y JUSTIFICACION DEL TEMA 11](#_Toc7711186)

[1.1.1. Planteamiento del problema 11](#_Toc7711187)

[1.2. SITUACIÓN ACTUAL 11](#_Toc7711188)

[MISION: 12](#_Toc7711189)

[VISION: 12](#_Toc7711190)

[1.2.1. Datos adicionales de la empresa 12](#_Toc7711191)

[1.2.1.1. Perfil del proyecto: Negocio local 12](#_Toc7711192)

[1.2.1.2. Perfil del producto: Servicio de preparación de platillos 12](#_Toc7711193)

[1.3. PROBLEMAS 12](#_Toc7711194)

[Demora para tomar el pedido 12](#_Toc7711195)

[Demora en la entrega de comida 12](#_Toc7711196)

[Pedido incorrecto 12](#_Toc7711197)

[Falta de atención a los clientes 12](#_Toc7711198)

[1.4. JUSTIFICACION 13](#_Toc7711199)

[1.5. OBJETIVOS DE LA JUSTIFICACION 13](#_Toc7711200)

[1.6. OBJETIVOS GENERAL 13](#_Toc7711201)

[1.7. OBJETIVOS ESPECIFICO 13](#_Toc7711202)

[1.8. METODOLOGIA DE LA INVESTIGACION 14](#_Toc7711203)

[Método exploratorio 14](#_Toc7711204)

[Método análisis-síntesis 14](#_Toc7711205)

[Tipo de investigación: 14](#_Toc7711206)

[1.9. PASOS DE LA ELABORACION DEL PROYECTO 14](#_Toc7711207)

[1.10. LIMITACIONES 15](#_Toc7711208)

[1.11. ALCANCES DE LA INVESTIGACIÓN 15](#_Toc7711209)

[MARCO TEORICO Y CONCEPTUAL 16](#_Toc7711210)

[2.1. HERRAMIENTAS USADAS 16](#_Toc7711211)

[2.1.1. Herramientas del Modelado y Diseño de Sistemas 16](#_Toc7711212)

[2.1.2. Plataforma de desarrollo Web 17](#_Toc7711213)

[2.1.3 Motor de Base de datos 17](#_Toc7711214)

[2.1.4.Lenguaje de Programación Microsoft visual c sharp 19](#_Toc7711215)

[2.2. TÉCNICAS DE TEMÁTICAS 20](#_Toc7711216)

[2.2.1. Mapa de procesos 20](#_Toc7711217)

[2.2.1.1. Procesos estratégicos: 21](#_Toc7711218)

[2.2.1.2. Procesos operativos: 21](#_Toc7711219)

[2.2.1.3. Procesos de apoyo: 21](#_Toc7711220)

[2.2.2. Procedimientos almacenados 22](#_Toc7711221)

[2.2.3. Diagrama de flujo 22](#_Toc7711222)

[2.2.4. Software - ciclo de vida de desarrollo 23](#_Toc7711223)

[2.2.5. Hardware 24](#_Toc7711224)

[2.2.6. Recursos Humanos 24](#_Toc7711225)

[2.2.7. Patrón de Diseño Modelo – Vista – Controlador 24](#_Toc7711226)

[2.2.8. Modelo incremento 24](#_Toc7711227)

[2.2.8.1. FASES DEL MODELO INCREMENTAL 25](#_Toc7711228)

[CAPITULO III 26](#_Toc7711229)

[3.1. MAPA DE PROCESOS 26](#_Toc7711230)

[3.2. FLUJOGRAMA DE PROCESOS 28](#_Toc7711231)

[3.3. INVENTARIO DE REQUERIMIENTOS 30](#_Toc7711232)

[3.4. GESTION DEL PRODUCTO Y PROYECTO 32](#_Toc7711233)

[3.4.1. Fase inicial 33](#_Toc7711234)

[3.4.2. Fase de elaboración 33](#_Toc7711235)

[3.5. DESARROLLO DE CICLO DE VIDA DEL SOFTWARE 33](#_Toc7711236)

[3.5.1. Prototipo 33](#_Toc7711237)

[3.6. PLAN DE PRUEBAS 40](#_Toc7711238)

[3.6.1 INCREMENTO 40](#_Toc7711239)

[3.6.1.1. INCREMENTO 1 40](#_Toc7711240)

[3.6.1.2. INCREMENTO 2 42](#_Toc7711241)

[. INCREMENTO 3 42](#_Toc7711242)

[3.6.1.3. CASO DE USO SELECCIONADO 43](#_Toc7711243)

[3.7. PROCEDIMIENTOS ALMACENADOS 44](#_Toc7711244)

[3.8. GESTION DE LA CONFIGURACION 45](#_Toc7711245)

[3.8.1. Github 45](#_Toc7711246)

[3.9 PLAN DE PRUEBAS 45](#_Toc7711247)

[3.9.1 PLAN DE PRUEBAS DEL INCREMENTO 45](#_Toc7711248)

[3.9.1.1 PLAN DE PRUEBAS DEL INCREMENTO 1 45](#_Toc7711249)

[LOGIN ADMINISTRADOR 50](#_Toc7711250)

[REGISTRAR PROVEEDOR 54](#_Toc7711251)

[3.1. Evidencias de código almacenado en github 58](#_Toc7711252)

[3.2. META 1: MODELAMIENTO 55](#_Toc7711253)

[3.3. META 2: MODELAMIENTO 59](#_Toc7711254)

[PROCEDIMIENTOS ALMACENADOS 63](#_Toc7711255)

[CODIGO M-V-C DEL CASO DE USO SELECCIONADO 70](#_Toc7711256)

[PROCEDIMIENTO ALMACENADO DEL CASO DE USO SELECCIONADO 70](#_Toc7711257)

[3.4. EVIDENCIAS DE SOFTWARE 70](#_Toc7711258)

[CAPITULO IV 72](#_Toc7711259)

[CONCLUSIONES Y RECOMENDACIONES 72](#_Toc7711260)

[4.1. CONCLUSIONES 72](#_Toc7711261)

[4.2. RECOMENDACIONES 73](#_Toc7711262)

[4.3. BIBLIOGRAFIA 73](#_Toc7711263)

[4.4. INDICE DE ABREVIATURAS DE SIGLAS 74](#_Toc7711264)

[4.5. GLOSARIO DE TERMINOS 74](#_Toc7711265)

[4.6. APENDICE 76](#_Toc7711266)

# INDICE DE TABLAS

[Tabla 1- Requerimientos funcionales 30](#_Toc7718717)

[Tabla 2- Planificación de unidades de programación 31](#_Toc7718718)

[Tabla 3- Plan de pruebas del restaurant "Doña Rufina" 41](#_Toc7718719)

[Tabla 4-Cumplimiento de pruebas del restaurant "Doña Rufina" - Resultado 41](#_Toc7718720)

[Tabla 5- Plan de prueba Incremento 2 42](#_Toc7718721)

[Tabla 6- Plan de prueba Incremento 2 - Resultado 42](#_Toc7718722)

[Tabla 5- Plan de prueba Incremento 3 42](#_Toc7718723)

[Tabla 6- Plan de prueba Incremento 3 - Resultado 43](#_Toc7718724)

[Tabla 7- Plan de prueba del caso de uso seleccionado 43](#_Toc7718725)

[Tabla 8-Plan de prueba del caso seleccionado - resuelto 44](#_Toc7718726)

[Tabla 9-Procedimientos almacenados 44](#_Toc7718727)

# INDICE DE ILUSTRACIONES

##### [Ilustración 1 Balsamiq Mockup 17](#_bookmark28)

##### [Ilustración 2 Github 17](#_bookmark30)

##### [Ilustración 3 Microsoft sql server 18](#_bookmark32)

##### [Ilustración 4 Sql Server Management Studio 19](#_bookmark33)

##### [Ilustración 5 Ghostdoc 19](#_bookmark34)

##### [Ilustración 6 Microsoft Visual C Sharp 20](#_bookmark36)

##### [Ilustración 7 Actividades SDLC 23](#_bookmark45)

##### Ilustración 8 Mapa de Procesos AS-IS del restaurante "Doña Rufina" N° 1 26

##### Ilustración 9 Mapa de Procesos AS-IS del restaurante "Doña Rufina" N° 2 26

##### Ilustración 10 Flujograma de procesos del restaurant "Doña Rufina" N° 1 27

##### Ilustración 11 Flujograma de procesos del restaurant "Doña Rufina" N° 2 28

##### [Ilustración 12 Prototipos de Software 31](#_bookmark60)

##### Ilustración 13 Incremento 1 38

##### [Ilustración 14 Plan de pruebas del módulo Registrar cliente 41](#_bookmark68)

##### [Ilustración 15 Plan de pruebas del módulo Registrar administrador 44](#_bookmark69)

##### [Ilustración 16 Plan de pruebas del módulo Login administrador 46](#_bookmark70)

##### [Ilustración 17 Plan de pruebas del catalogo 47](#_bookmark71)

##### [Ilustración 18 Plan de pruebas del módulo Registrar platillos 48](#_bookmark72)

##### [Ilustración 19 Plan de pruebas del módulo Registrar proveedor 50](#_bookmark73)

##### [Ilustración 20 Plan de pruebas del módulo Registrar ingredientes 51](#_bookmark74)

## CERTIFICADO DE REVISION DE REDACCION, ORTOGRAFIA Y ESTILO

Nuestro Grupo del curso de Sistemas de Información Integrados con el código referencial SAI-5263 a cargo de la elaboración del proyecto titulado “Sistema de pedidos para el restaurante Doña Rufina”. Declara:

Que, el informe presentado ha sido íntegramente elaborado por el grupo y que la redacción, ortografía y estilo, además se encuentra revisado el proyecto sin encontrar dificultades presentado por el grupo; comprometiéndonos a poner a disposición de la UNIVERSIDAD CONTINENTAL, los documentos que acrediten la veracidad de la información presentada; si esto fuera solicitado por la entidad.

En tal sentido asumimos toda la responsabilidad que corresponda ante cualquier error ortográfico u omisión tanto en los documentos como en la información aportada.

Nos afirmamos y ratificamos en lo expresado, en señal de lo cual firmamos el presente documento.

Atentamente:



Arroyo Cruzado Stephanie Flores Guillen Alexander DNI: 75191823 DNI:



Hinostroza Granados Giancarlo F. Rodríguez Manuelo Jhoelver DNI: DNI:



Rojas Samaniego Cecili DNI: 72767995

Huancayo, 24 de Abril del 2018

## DECLARACIÓN DE AUTENTICIDAD

Nuestro Grupo del curso de Sistemas de Información Integrados con el código referencial SAI-5263 a cargo de la elaboración del proyecto titulado “Sistema de Control de Asistencia y Notas”.

Declara:

Que, el informe presentado ha sido íntegramente elaborado por el Grupo y que no existe plagio alguno, presentado por otra persona, grupo o institución, comprometiéndonos a poner a disposición de la UNIVERSIDAD CONTINENTAL, los documentos que acrediten la veracidad de la información presentada; si esto fuera solicitado por la entidad.

En tal sentido asumimos toda la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto en los documentos como en la información aportada.

Nos afirmamos y ratificamos en lo expresado, en señal de lo cual firmamos el presente documento. Atentamente:

Arroyo Cruzado Stephanie Flores Guillen Alexander DNI: 75191823 DNI:



Hinostroza Granados Giancarlo F. Rodríguez Manuelo Jhoelver DNI: DNI:



Rojas Samaniego Cecili DNI: 72767995

Huancayo, 24 de Abril del 2018.

# RESUMEN EJECUTIVO

Actualmente en la ciudad de Huancayo los restaurantes no cuentan con un sistema gestor de pedidos esta situación hace que en el proceso de tomar la orden de un cliente hasta la atención de la misma se tome más de lo debido. Además, el no contar con un sistema informático para tomar las ordenes obliga al personal del restaurante a tener que desplazarse un gran un número de veces de un lugar a otro para poder cumplir con el pedido del cliente, ocasionando deficiencias en el servicio, olvido de órdenes, retardos, y equivocaciones en los pedidos.

El proyecto "Sistema de pedidos para el restaurant Doña Rufina", consiste en la implementación de un sistema de escritorio para gestionar el proceso desde el pedido de orden hasta la entrega de comida a los comensales.

El presente trabajo está desarrollado para la implementación y ejecución en el restaurante “Doña Rufina” siendo la principal interesada la dueña del restaurant la señora Rufina Manuelo Quispe.

El presente proyecto tiene como objetivo mejorar la atención de los clientes, entrega de pedidos, eficiencia en la atención y calidad de servicio.

En cuanto a la funcionalidad se cuenta con los módulos de acceso de usuarios, registro de pedidos, registro de clientes para la reserva de pedidos, y un módulo completo para la gestión de platos del día.

El sistema está desarrollado en base a las necesidades y requerimientos identificados en el restaurante “Doña Rufina”, en donde se ha seguido etapas y una serie de incrementos en el diseño para facilitar el uso y adaptabilidad en los usuarios finales

Palabras clave: Pedidos, menú, restaurante, comida, sistema informático.

# ABSTRACT

Currently in the city of Huancayo restaurants do not have a management system of orders . This situation means that in the process of taking the order of a client until the attention of the same is taken more than it should. In addition, the lack of a computer system to take orders forces the restaurant staff to have to travel a great number of times from one place to another in order to fulfill the customer's order, causing deficiencies in the service, forgetting orders , delays, and mistakes in orders.

The project "Order system for the restaurant Doña Rufina", consists of the implementation of a desktop system to manage the process from the order to the delivery of the food to the diners.

The present work is developed for the implementation and execution in the restaurant "Doña Rufina" being the main interested the owner of the restaurant Mrs. Rufina Manuelo Quispe.

This project aims to improve customer service, order delivery, efficiency in service and quality of service.

In terms of functionality, there are user access modules, order registration, customer registration for the reservation of orders, and a complete module for the management of dishes of the day.

The system is developed based on the needs and requirements identified in the restaurant "Doña Rufina", where it has followed stages and a series of increases in design to facilitate the use and adaptability in end users

Keywords: Orders, menu, restaurant, food, computer system.

# INTRODUCCIÓN

# CAPITULO I

## PLANTEAMIENTO Y JUSTIFICACION DEL TEMA

En la ciudad de Huancayo los restaurantes no cuentan con un sistema gestor de pedidos esta situación hace que en el proceso de tomar la orden de un cliente hasta la atención de la misma se tome más de lo debido. Además, el no contar con un sistema informático para tomar las ordenes obliga al personal del restaurante a tener que desplazarse un gran un número de veces de un lugar a otro para poder cumplir con el pedido del cliente, ocasionando deficiencias en el servicio, olvido de órdenes, retardos, y equivocaciones en los pedidos.

### **Planteamiento del problema**

¿Cómo diseñar e implementar un sistema de pedidos de comida en el restaurante “Doña Rosita” de la ciudad de Huancayo?

## SITUACIÓN ACTUAL

En cuanto a la situación actual del restaurante Doña Rufina, se identificó que se tiene una gran concurrencia de clientes por día, la comida se encuentra a buen gusto de los comensales que van a este lugar.

Doña Rufina Manuelo Quispe abrió un pequeño puesto de comida en el mercado mayorista en aquellos de 1996 por una necesidad económica, años más tarde el negocio familiar prospero. Expandiendo así el negocio de ventas a las ferias distritales y regionales, introduciéndose a la preparación de platillos a pedido.

Actualmente siguen prosperando con la venta de este negocio familiar en distintos lugares de Huancayo y en diversas fiestas patronales.

### **MISION:**

Ofrecer a nuestros clientes alimentos y un servicio de calidad con una relación costo, beneficio, estableciéndonos como meta satisfacer y superar las expectativas del cliente.

### **VISION:**

Para el año 2021 ser una empresa reconocida a nivel distrital y expandir nuestras fronteras por toda la región Junín

### **Datos adicionales de la empresa**

**NOMBRE:** “Doña Rufina”

**RUC:** 10167796163

**DIRECCION:** Jirón Omar Yali 343, Huancayo

**NOMBRE DEL CONTACTO:** Rufina Manuelo Quispe (Director ejecutivo)

* + - 1. **Perfil del proyecto:** Negocio local
      2. **Perfil del producto:** Servicio de preparación de platillos

## PROBLEMAS

En el restaurante “Doña Rufina” se identificó una serie de dificultades tales como:

### **Demora para tomar el pedido**

Después de sentarse, generalmente los clientes quieren los menús en unos 2 a 5 minutos para decidir sobre lo que quieren comer, en el restaurant Dona Rufina existe una demora para tomar el pedido cuando hay mucha concurrencia de personas.

Demora en la entrega de comida

De la misma forma cuando el restaurante tiene muchos comensales al mismo tiempo la entrega del pedido en algunas ocasiones tarda demasiado, generando una especie de malestar al cliente.

### **Pedido incorrecto**

Este no es un problema recurrente en este restaurante, pero sí es un punto importante que se debe corregir, ya que la calidad de servicio depende de entregar el pedido correcto.

### **Falta de atención a los clientes**

Este problema ocurre cuando el comensal realiza pedidos adicionales a su orden, los meseros suelen descuidar la mesa cuando ya se ha entregado el pedido por atender a todas las demás.

## JUSTIFICACION

A partir de los problemas del restaurant “Doña Rufina” descritos anteriormente se identificó las principales falencias que se deben resolver.

Es por ello que la esencia principal de éste proyecto es que el restaurante “Doña Rufina” mejore la calidad de servicio que ofrece a los comensales de su restaurant, mejorando el tiempo de atención y la calidad de servicio por medio de un software de gestión que se puede utilizar en dispositivos Tablet en donde se pueda consultar el menú, tomar, entregar y finalizar el pedido, registrar clientes en el caso de reservas y registrar el menú del día todo con un control eficaz. De esta forma proporcionar a esta empresa una herramienta eficiente que genere mayores ingresos al negocio.

Generar también un ambiente e imagen agradable del restaurante ya que la tecnología siempre suma a vista de los clientes.

## OBJETIVOS DE LA JUSTIFICACION

* Mejorar el tiempo para tomar el pedido de cada cliente
* Reducir el tiempo de entrega de cada pedido de comida
* No entregar pedidos incorrectos
* Mejorar la atención de los clientes

## OBJETIVOS GENERAL

Diseñar e implementar un sistema de escritorio para la gestión de pedidos de comida en el restaurante “Doña Rosita”

## OBJETIVOS ESPECIFICO

* Acceder al sistema con diferentes privilegios de usuario
* Registrar platos y sus categorías
* Registra clientes
* Registrar pedidos
* Enviar pedido para preparación de plato
* Registrar carta de menú
* Mostrar carta de menú por día
* Visualizar los pedidos realizados por mesa

## METODOLOGIA DE LA INVESTIGACION

Las metodologías utilizadas para el presente trabajo fueron los siguientes:

### **Método exploratorio**

Se aplicó este método durante todo el proceso de desarrollo de la aplicación, ya que fue necesario saber cómo funcionaba el negocio y a partir de aquí ir aplicando las funciones en el sistema.

### **Método análisis-síntesis**

Que consiste en la separación de ideas o conceptos, de esta forma tener una comprensión plena del tema en estudio y el marco teórico.

### **Tipo de investigación:**

El tipo de investigación aplicada en el presente trabajo fue la investigación aplicada ya que como producto final se implementa una aplicación informática en la empresa en estudio.

## PASOS DE LA ELABORACION DEL PROYECTO

Para realizar este proyecto se utilizó un proceso que describe desde la fase inicial hasta la fase final del proyecto, de esta forma garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo.

###### Objetivo del Proyecto

Como fase inicial se realizó la definición de metas a lograr. Para llegar a este punto inicial anteriormente se identificaron los problemas y necesidades de la empresa en estudio y elaborar el presente proyecto.

###### Investigación

Referido al recojo de información que se realizó en la empresa en estudio, con la dueña del restaurant e información de libros e internet. Esta parte sirvió también para para aclarar los puntos claves en el desarrollo del trabajo.

###### Investigación

Esta parte sirvió para determinar el tipo de proyecto que se debía desarrollar, las funciones y procesos a implementar.

###### Planificación

Se asignó tareas y fechas de entrega a cada integrante de grupo, para determinar aproximadamente la fecha el inicio y fin del proyecto, se hizo uso de los cronogramas para gestionar los entregables más importantes.

###### Identificar recursos

Referido a las herramientas con las que contamos para el desarrollo del proyecto, tales como las laptops, y software que sirven de plataforma para el desarrollo de la aplicación.

###### Plan de comunicación

Reuniones constantes con el equipo de trabajo para coordinar y aclarar los requerimientos, necesidades, funciones y procesos del restaurant “Doña Rufina“.

###### Evaluación constante

Evaluación del funcionamiento de cada módulo del proyecto, de esta forma ir cumpliendo con los objetivos planteados en la realización del presente trabajo.

###### Reutilización

Referido a la utilización de código en módulos que tienen la misma funcionalidad o a documentos similares anteriores que sirvan de referencia para reforzar este proyecto.

## LIMITACIONES

##### Una de las grandes limitaciones durante el desarrollo de este proyecto fue el tiempo para estar físicamente presentes en el restaurante “Doña Rufina” ya que gran parte del equipo de trabajo vivía lejos de la ubicación de lugar, se llegó a la conclusión que para tener puntos más minuciosos del giro de negocio de esta empresa se tiene que aplicar observación del lugar por más tiempo.

**Limitaciones funcionales del sistema**

* Ningún usuario podrá eliminar o ningún otro.
* El usuario no registrará las mesas
* El cliente no podrá hacer pedidos online.
* El registro de clientes solo se hará para pedidos con reserva.
* El sistema funcionará solamente en dispositivos en donde se tenga instalada la aplicación

## ALCANCES DE LA INVESTIGACIÓN

Los alcances del presente proyecto se enfocan en el cumplimiento de los objetivos planteados. Donde en consecuencia de la implementación y puesta en marcha del software en el restaurant “Doña Rufina”, la gestión y calidad de servicio mejoraran en un 80% por ciento, como resultado, la demanda de clientes aumentará y mejorará los ingresos y ganancias de la empresa

**CAPITULO II**

# MARCO TEORICO Y CONCEPTUAL

## HERRAMIENTAS USADAS

Las herramientas que se usaron para el desarrollo de las soluciones de escritorio son en su mayoría de código abierto ello quiere decir que no poseen un ente privado que restrinja el uso del software o exija de algún pago, como una licencia para hacer uso del mismo es por ello que el equipo de desarrollo considero necesario hacer uso de las siguientes herramientas.

## Herramientas del Modelado y Diseño de Sistemas

#### BALSAMIQ MOCKUP

Herramienta que permite diseñar de forma rápida y sencilla maquetas de interfaz para webs y aplicaciones móviles. Programado en Flex y AIR por lo que crea muy rápido Wireframes. Interfaz fácil de usar. Como se ha creado con AIR es instalable tanto en Windows como Linux y Mac OS Balsamiq te permite escoger entre un montón de objetos prediseñados como: barras de estado, menús, barras de progreso, etc. Además, te permite exportar el diseño que realices en PNG, PDF e incluso al portapapeles. Permite incrustar tus diseños en tu página web o en informes de errores. Puedes compartir tus diseños a través de un sistema muy parecido a Dropbox. Balsamiq funciona como cualquier aplicación normal, te permite arrastrar, soltar, guardar mockups en un archivo, copiar y pegar, deshacer, etc.

*Ilustración 1 Balsamiq Mockup*



*Balsamiq mockup es una interfaz que nos permite diseñar de forma rápida nuestros diseños para la aplicación que desarrollamos.*

## Plataforma de desarrollo Web

#### GITHUB

GitHub, un servicio para mantener tu código a salvo de peligro y es una plataforma de desarrollo inspirada en la forma en que trabajas. Desde el código abierto hasta el negocio, puede alojar y revisar códigos, administrar proyectos y crear software junto a millones de otros desarrolladores. (Github, 2008)

*Ilustración 2 Github*



*Github es una plataforma que nos permitirá guardar nuestro código para mantenerlo a salvo.*

# Motor de Base de datos

###### MICROSOFT SQL SERVER

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).

*Ilustración 3 Microsoft sql server*



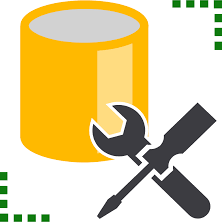
*Microsoft sql server es un sistema de gestión que nos permite usar la base de datos con uno de sus complementos*

**SQL SERVER MANAGEMENT STUDIO**

SQL Server Management Studio (SSMS) es una aplicación de software lanzada por primera vez con Microsoft SQL Server 2005 que se utiliza para configurar, administrar y administrar todos los componentes dentro de Microsoft SQL Server. La herramienta incluye editores de script y herramientas gráficas que trabajan con objetos y características del servidor.

Una característica central de SSMS es el Explorador de objetos, que permite al usuario navegar, seleccionar y actuar sobre cualquiera de los objetos dentro del servidor. También se distribuyó una edición Express separada que se podía descargar libremente, sin embargo, las versiones recientes de SSMS son completamente capaces de conectarse y administrar cualquier instancia de SQL Server Express. Microsoft también incorporó compatibilidad con versiones anteriores para versiones anteriores de SQL Server, permitiendo así que una versión más reciente de SSMS se conecte a versiones anteriores de instancias de SQL Server.

*Ilustración 4 Sql Server Management Studio*



*SSMS es una base de datos que nos permite operar de manera directa con los servidores de manera fácil.*

#### GHOSTDOC

GhostDoc es una extensión de Visual Studio para desarrolladores que necesitan generar comentarios XML a partir del código fuente utilizando plantillas personalizables, mantener documentación limpia y actualizada, producir documentación de ayuda en múltiples formatos, usar el código fuente inteligente Spell Checker en Visual Studio y más.

GhostDoc también ayuda a automatizar la generación de documentación limpia y útil de su código limpio, a compilarlo y a empaquetarlo todo sin intervención humana. (GhostDoc, 2019)

*Ilustración 5 Ghostdoc*



*Ghostdoc en una extensión de Visual Studio que nos permite generar documentación.*

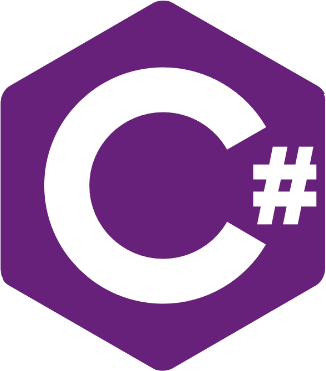
# Lenguaje de Programación Microsoft visual c sharp

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. 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. Además, es muy sencillo al usar, tiene un lenguaje de última generación, incorpora elementos que se ha demostrado a lo largo del tiempo que son muy útiles para el programador, C# incluye mecanismos de control de acceso a tipos de datos, lo que garantiza que no se produzca errores difíciles de detectar como un acceso a memoria de ningún objeto.

*.*

*Ilustración 6 Microsoft Visual C Sharp*



*Microsoft Visual C Sharp es un lenguaje de programación que nos permite muchas funcionalidades.*

## TÉCNICAS DE TEMÁTICAS

La técnica se refiere a los procedimientos y recursos que se emplean para lograr un resultado específico para ayudar a determinar los requerimientos funcionales y no funcionales de una empresa.

## Mapa de procesos

El mapa de procesos es una representación gráfica de los procesos de una organización. Este diagrama condenso de forma visual todos los procesos y subprocesos de la empresa y la interrelación existente entre los mismos. Podría ejemplificarse como una especie de telaraña que une y conecta las diferentes actividades que realizan los departamentos de la compañía.

Ahora bien, debe reflejar las entradas y salidas de cada proceso, teniendo en cuenta que la entrada tiene unos objetivos marcados en relación al cliente y la salida de éste tiene que satisfacer las necesidades del cliente en relación a los objetivos marcados. (Captio, 2017) La manera de generar estos gráficos es hasta cierto punto independientemente de hacerlo, siempre y cuando sean:

Auto explicativos: El gráfico es sencillo de interpretarse por sí mismo, debido al flujo y sentido con se genera entre las relaciones. Contengan los procesos necesarios: El mapa de procesos dibujado no ha dejado diagramas sueltos o vagamente definidos en relación con las actividades que se quieren esclarecer.

Admite despliegues: El mapa puede ser desplegado (descompuesto) en varios niveles de despliegue. Para dejar la ambigüedad de los mapas elaborados para los recién entendidos del tema, existen tipos de mapas de procesos para poder elaborarlo de mejor forma. (Álvarez, 2012)

Mapa de procesos convencional: En este tipo de mapa de procesos se clasifican a los procesos de en tres tipos:

## Procesos estratégicos:

También denominados procesos de dirección, pues en ellos la dirección tiene un papel relevante. Suelen estar relacionados con la estrategia y su evolución, y con el control global de la organización. Es curioso apreciar cómo en muchas

Organizaciones estos procesos no se encuentran configurados como procesos tipo. Estos son las actividades de la dirección.

## Procesos operativos:

Mediante estos procesos la organización genera los productos y servicios que entrega a sus clientes. Su conjunto constituye la cadena de valor de la organización, lo que significa que un desajuste en los mismos puede tener repercusiones importantes en los resultados de esta. Conforman el núcleo central del negocio y por ello también reciben otros nombres como procesos nucleares, procesos clave, procesos específicos, procesos de negocio. Son los procesos operativos o de desarrollo del negocio.

## Procesos de apoyo:

Se los designa también como procesos de soporte, de ayuda o de apoyo. Estos procesos dan apoyo a los estratégicos, a los específicos o a otros de soporte. En numerosas

ocasiones están relacionados con el suministro o mantenimiento de los recursos necesarios para el funcionamiento o mantenimiento de los recursos necesarios para el funcionamiento de la organización. Son los procesos de provisión de recursos necesarios para el funcionamiento de la organización.

En el momento de distribuir los procesos espacialmente, los procesos estratégicos se colocan en la parte superior del mapa, los auxiliares o de soporte en la inferior, y los operativos en la parte media, donde habitualmente se representa la cadena de valor (procesos operativos).

## Procedimientos almacenados

Un procedimiento almacenado es un programa (o procedimiento) almacenado físicamente en una base de datos. Su implementación varía de un gestor de bases de datos a otro. La ventaja de un procedimiento almacenado es que, al ser ejecutado, en respuesta a una petición de usuario, es ejecutado directamente en el motor de bases de datos, el cual usualmente corre en un servidor separado. Como tal, posee acceso directo a los datos que necesita manipular y sólo necesita enviar sus resultados de regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes.

Los procedimientos pueden ser ventajosos: cuando una base de datos es manipulada desde muchos programas externos. Al incluir la lógica de la aplicación en la base de datos utilizando procedimientos almacenados, la necesidad de embeber la misma lógica en todos los programas que acceden a los datos es reducida. Esto puede simplificar la creación y, particularmente, el mantenimiento de los programas involucrados.

Podemos ver un claro ejemplo de estos procedimientos cuando requerimos realizar una misma operación en un servidor dentro de algunas o todas las bases de datos y a la vez dentro de todas o algunas de las tablas de las bases de datos del mismo. Para ello podemos utilizar a los Procedimientos almacenados auto creable que es una forma de generar ciclos redundantes a través de los procedimientos almacenados. (Microsoft, 2017)

## Diagrama de flujo

Un diagrama de flujo es la representación gráfica del flujo o secuencia de rutinas simples. Tiene la ventaja de indicar la secuencia del proceso en cuestión, las unidades involucradas y los responsables de su ejecución, es decir, viene a ser la representación simbólica o pictórica de un procedimiento administrativo.

Luego, un diagrama de flujo es una representación gráfica que desglosa un proceso en cualquier tipo de actividad a desarrollarse tanto en empresas industriales o de servicios y en sus departamentos, secciones u áreas de su estructura organizativa. (Luis, 2011)

## Software - ciclo de vida de desarrollo

El ciclo de vida del desarrollo Software (SDLC en sus siglas inglesas), es una secuencia estructurada y bien definida de las etapas en Ingeniería de software para desarrollar el producto software deseado. (tutorialPoint, 2016)

Como ya conocemos hoy en día el software no es estático, más bien se ha convertido en un ente dinámico que constantemente esta cambios y sujetos a nuevos requerimientos para su acoplamiento a las reglas de negocio, en este caso la empresa del restaurante” Doña Rufina”. Es por ello que en el requerimiento de análisis se procedió a realizar una entrevista al dueño del negocio, con el fin de recabar la información necesaria para el desarrollo del software y proveer de las herramientas necesarias para la resolución de los problemas de la empresa, luego de ello, el diseño es necesario puesto que es en el que se maneja la interfaz de usuario la cual permitirá interactuar con el usuario final, quien mediante ella podrá retribuir de información para realizar pedidos, los casos siguientes como son la evolución e implementación aún no se realizan debido a que aún no se concluye el desarrollo de los incrementos, en lo que concierne al manejo de pruebas solo se ha considerado el desarrollo de manera local y en un ambiente externo.

*Ilustración 7 Actividades SDLC*



*El ciclo de vida presenta los pasos por los que atraviesa un software en desarrollo, para que esto sea utilizado en un entorno real.*

## Hardware

El sistema que se está desarrollando debe tener la capacidad técnica de manejar una gran cantidad de datos, por ello se requiere una computadora servidor que pueda soportar el almacenamiento de datos de la empresa del restaurante “Doña Ruffina”. Por ello deberá de obtener una computadora que permita a desarrollar el sistema.

## Recursos Humanos

Es necesario utilizar solo el escritorio y algunos documentos como Word, Excel, para diferentes reportes que brindara el sistema. Estos de bajan de manera gratuita en sus sitios correspondientes o si no, ya cuentan con el sistema instalado en la misma máquina.

## Patrón de Diseño Modelo – Vista – Controlador

El patrón de diseño de modelo-vista-controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control. El patrón requiere que cada uno de estos elementos esté separado en distintos objetos. El modelo (por ejemplo, la información de datos) contiene únicamente los datos puros de aplicación; no contiene lógica que describe cómo pueden presentarse los datos a un usuario. La vista (por ejemplo, la información de presentación) presenta al usuario los datos del modelo. La vista sabe cómo acceder a los datos del modelo, pero no sabe el significado de estos datos ni lo que el usuario puede hacer para manipularlos. Por último, el controlador (por ejemplo, la información de control) está entre la vista y el modelo. Escucha los sucesos desencadenados por la vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la reacción es llamar a un método del modelo. Puesto que la vista y el modelo están conectados a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la vista. El diagrama siguiente muestra cómo se aplica el patrón de diseño MVC, este patrón se utiliza para las aplicaciones web y para los Rich Client y se puede utilizar la infraestructura de servicios web. (IBM, 2017)

## Modelo incremento

El modelo incremental de gestión de proyectos tiene como objetivo un crecimiento progresivo de la funcionalidad. Es decir, el producto va evolucionando con cada una de las entregas previstas hasta que se amolda a lo requerido por el cliente o destinatario.

Este enfoque, que se usó inicialmente para proyectos de software, aunque más tarde se aplicó a otros sectores, establece entregas parciales mediante un calendario de plazos. En cada una de ellas, el producto debe mostrar una evolución con respecto a la fecha anterior; nunca puede ser igual.

Una de las claves para que esto se haga efectivo es la evaluación de las etapas. Los responsables del proyecto deben analizar si los resultados parciales son los esperados y si, sobre todo, apuntan al objetivo principal. De no ser así, deberán intervenir en él e implementar las soluciones que la situación requiera.

* + - 1. **FASES DEL MODELO INCREMENTAL**

El modelo de gestión incremental no es un modelo necesariamente rígido, es decir, que puede adaptarse a las características de cualquier tipo de proyecto, existen al menos 7 fases que debemos tener en cuenta a la hora de implementarlo:

1. Requerimientos: son los objetivos centrales y específicos que persigue el proyecto.
2. Definición de las tareas y las iteraciones: teniendo en cuenta lo que se busca, el siguiente paso es hacer una lista de tareas y agruparlas en las iteraciones que tendrá el proyecto. Esta agrupación no puede ser aleatoria. Cada una debe perseguir objetivos específicos que la definan como tal.
3. Diseño de los incrementos: establecidas las iteraciones, es preciso definir cuál será la evolución del producto en cada una de ellas. Cada iteración debe superar a la que le ha precedido. Esto es lo que se denomina incremento.
4. Desarrollo del incremento: posteriormente se realizan las tareas previstas y se desarrollan los incrementos establecidos en la etapa anterior.
5. Validación de incrementos: al término de cada iteración, los responsables de la gestión del proyecto deben dar por buenos los incrementos que cada una de ellas ha arrojado. Si no son los esperados o si ha habido algún retroceso, es necesario volver la vista atrás y buscar las causas de ello.
6. Integración de incrementos: una vez son validados, los incrementos dan forma a lo que se denomina línea incremental o evolución del proyecto en su conjunto. Cada incremento ha contribuido al resultado final.
7. Entrega del producto: cuando el producto en su conjunto ha sido validado y se confirma su correspondencia con los objetivos iniciales, se procede a su entrega final.

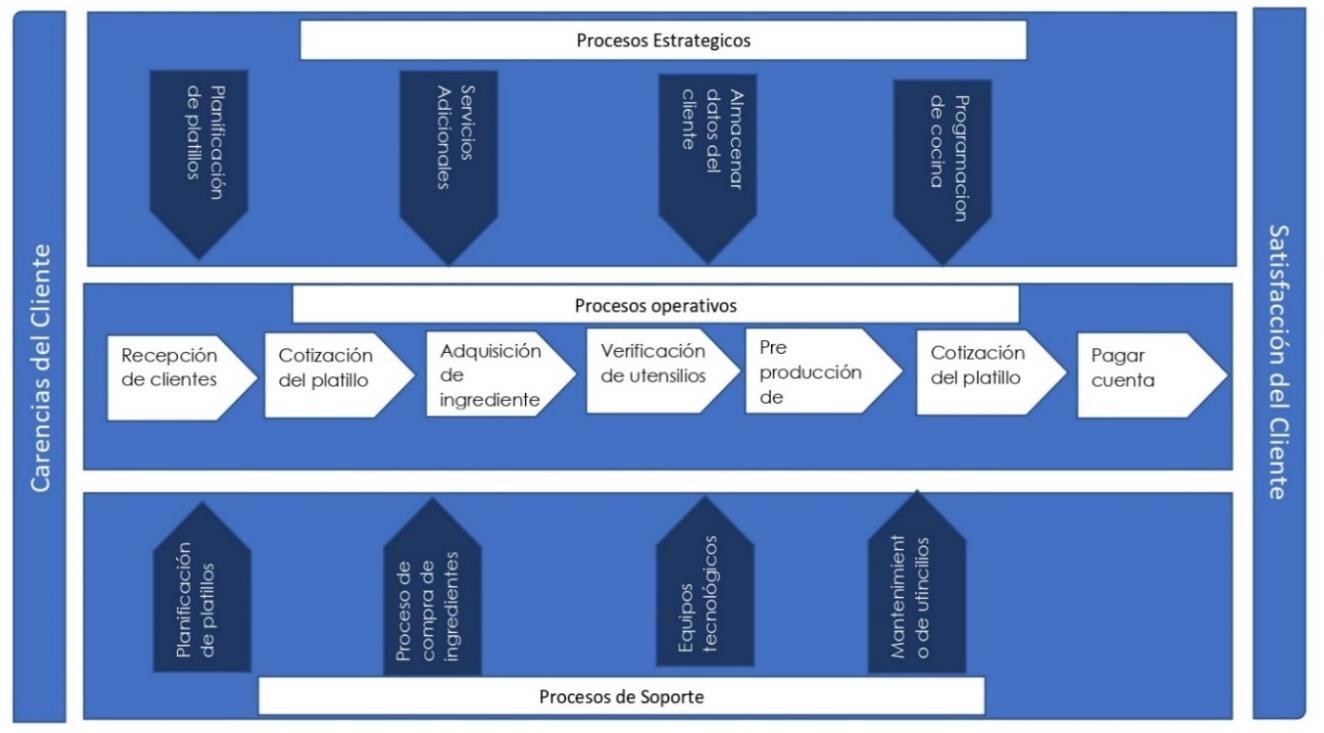
# CAPITULO III

**INGENIERIA DEL PRODUCTO O DE DESARROLLO DE LA INVESTIGACIÓN**

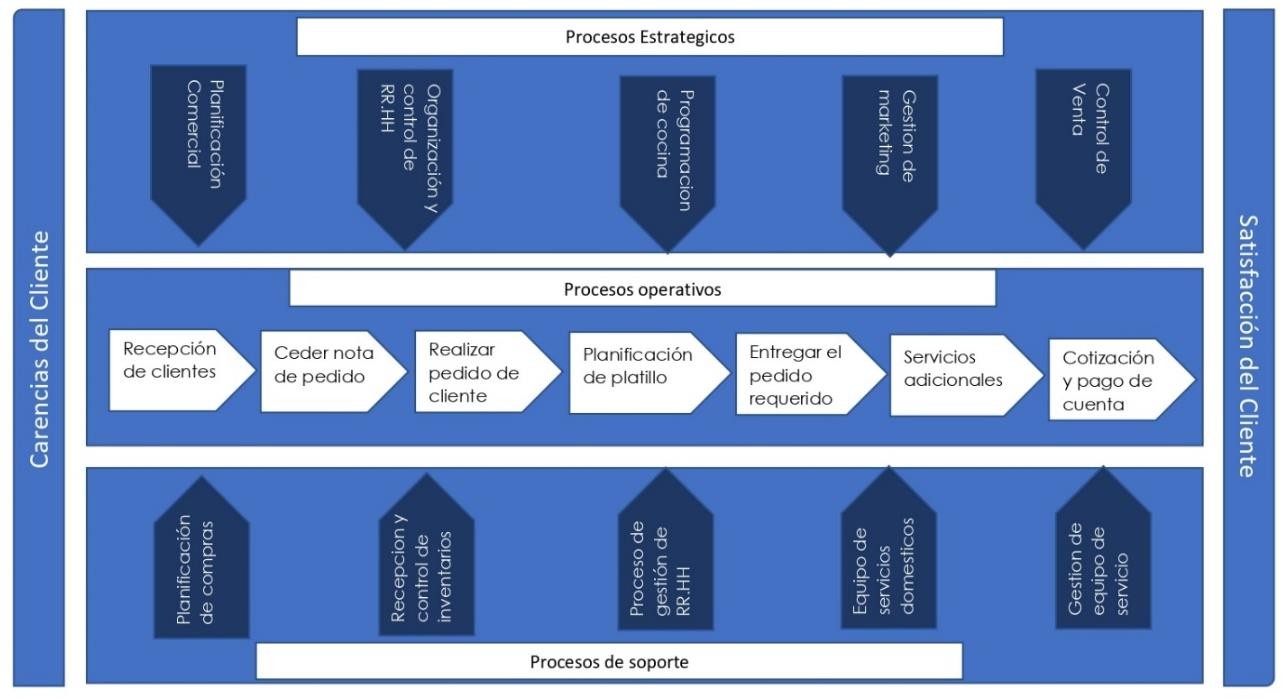
## MAPA DE PROCESOS

El mapa de procesos nos sirve para entender y [mejorar](https://blog.pro-optim.com/2016/10/06/como-definir-la-estructura-de-mi-empresa/) el proceso de control y venta de restaurante “Doña Rufifa”, El proceso nos brindó una oportunidad de observar su funcionamiento desde cada uno de nuestras perspectivas, lo cual nos permite ver la secuencia de dicho proceso desde otras perspectivas facilitando así el trabajo de corrección y mejora en la organización. Y ver todas las actividades que participan en cada una de las tareas. En este caso logramos optimizar el proceso de control de venta del restaurante “Doña Rufina”

* + - **Procesos Estratégicos:** Nos brinda las metas de la empresa como serían las ventas de servicio al cliente en tiempo real y de manera efectiva.
    - **Procesos Operativos:** Son los servicios que se genera a los clientes y públicos interesados para tener un servicio adecuado.
    - **Proceso de Soporte:** Generan un servicio de apoyo a los procesos para las atenciones de clientes en horarios estandarizados.

*Ilustración 8 Mapa de Procesos AS-IS del restaurante "Doña Rufina" N° 1*

*El mapa de procesos enfocado a mejorar la gestión de pedidos*

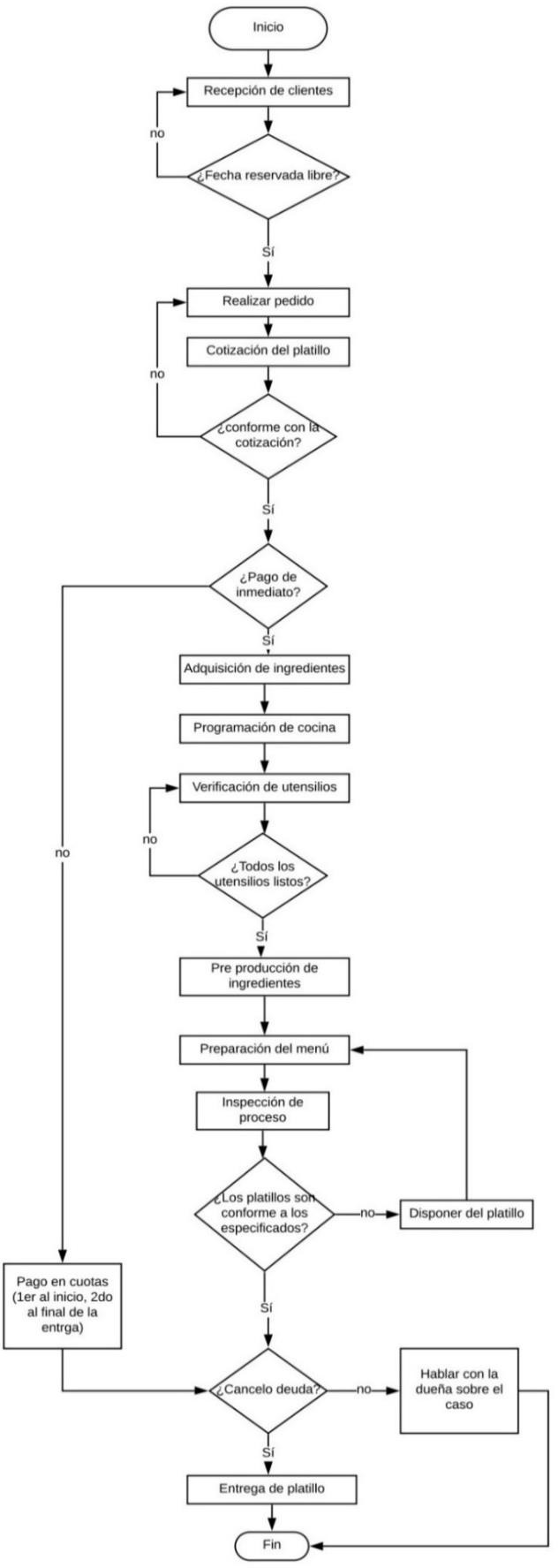
*Ilustración 9 Mapa de Procesos AS-IS del restaurante "Doña Rufina" N° 2*

*El mapa de procesos enfocado a mejorar la gestión de pedidos*

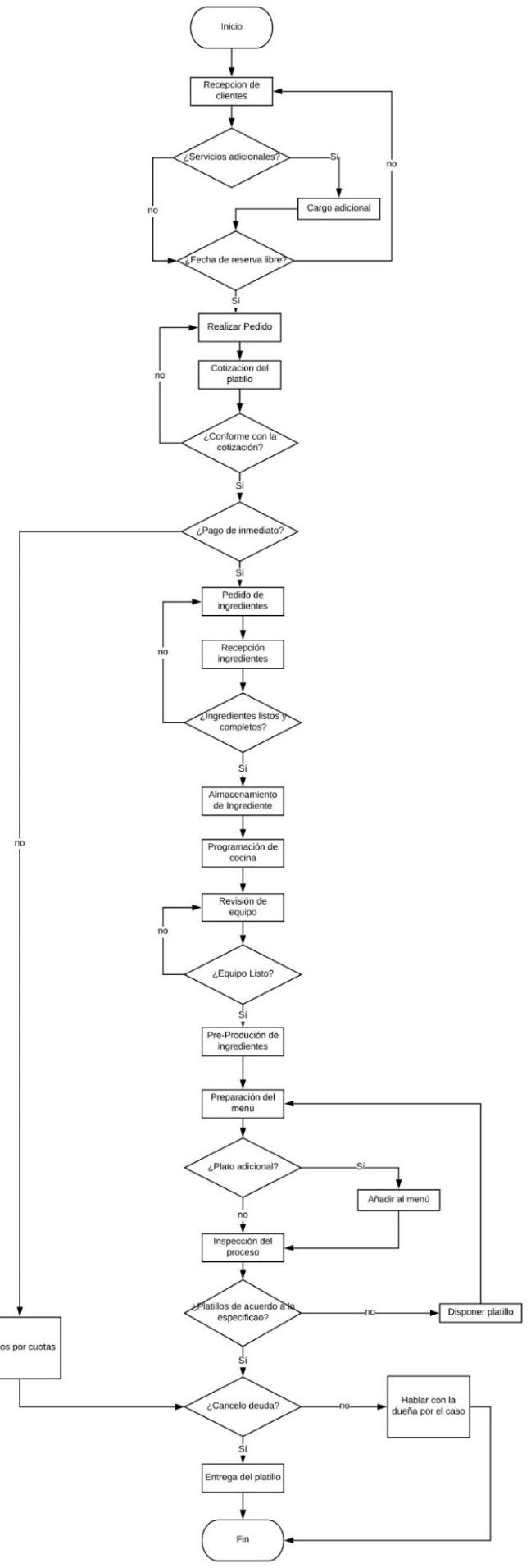
## FLUJOGRAMA DE PROCESOS

El flujograma de procesos es una representación gráfica de una secuencia de actividades o acciones del restaurante “Doña Rufina”, implica un proceso determinado. Es decir, el flujograma consiste en representar gráficamente, situaciones, hechos, movimientos y relaciones de control de venta de dicho restaurante mediante el uso de símbolos.

*Ilustración 10 Flujograma de procesos del restaurant "Doña Rufina" N° 1*



*Ilustración 11 Flujograma de procesos del restaurant "Doña Rufina" N° 2*



*Representación gráfica del proceso para tomar un pedido*

## INVENTARIO DE REQUERIMIENTOS

El inventario de requerimientos nos facilitó construir la aplicación del restaurante “Doña Rufina” que resuelve las necesidades del servicio al cliente de manera óptima los procesos del negocio, se estableció con detalle las funciones, servicios y restricciones operativas del sistema. De esta manera los requerimientos nos facilitaron para elaborar el caso de uso general, la planificación de unidades de programación, el modelo lógico de la base de datos, los prototipos, los procedimientos almacenados en la base de datos y el plan de pruebas en los diferentes Software.

Tabla 1- Requerimientos funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| Requerimientos funcionales | Incremento | Fecha Inicio | Fecha Fin |
| Registrar Cliente | 1 | 05/04/2019 | 12/04/2019 |
| Mostrar Promoción | 4 | 28/05/2019 | 17/06/2019 |
| Mostrar carta | 3 | 26/04/2019 | 27/05/2019 |
| agregar platillo | 2 | 12/04/2019 | 26/04/2019 |
| Registrar Platillo | 1 | 11/04/2019 | 08/05/2019 |
| Servicio Adicional | 3 | 26/04/2019 | 27/05/2019 |
| Cotizar Platillo | 3 | 26/04/2019 | 27/05/2019 |
| Pagar | 5 | 06/06/19 | 03/07/19 |
| Comprobante de pago | 5 | 18/06/2019 | 04/07/2019 |
| Enviar reporte de pedido de venta | 4 | 28/05/2019 | 17/06/2019 |
| Stock de platillos | 4 | 28/05/2019 | 17/06/2019 |
| Realizar pedido | 2 | 12/04/2019 | 26/04/2019 |
| Entregar pedido | 3 | 26/04/2019 | 27/05/2019 |
| Eliminar platillo | 2 | 12/04/2019 | 26/04/2019 |
| Actualizar datos | 3 | 26/04/2019 | 30/05/2019 |
| Generar Reportes Cada  tres semanas | 5 | 18/06/2019 | 04/07/2019 |
| Registrar usuarios | 1 | 11/04/2019 | 08/05/2019 |
| Registrar proveedor | 3 | 26/04/2019 | 27/05/2019 |
| Registrar ingredientes | 2 | 12/04/2019 | 26/04/2019 |
| Eliminar cliente | 2 | 12/04/2019 | 26/04/2019 |
| Eliminar ingrediente | 3 | 26/04/2019 | 27/05/2019 |
| Registrar dirección cliente | 1 | 11/04/2019 | 08/05/2019 |

Tabla 2- Planificación de unidades de programación

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requerimiento | Incremento | Tipo Clase | Nombre Clase | Responsable | Fecha | Fecha Fin |
| Inicio |
| Registrar Platillo |  | Modelo | M\_Registrar\_Platillo | RSC | 05/04/2019 | 12/04/2019 |
| 1 | Vista | V\_ Registrar \_Platillo | FGA | 05/04/2019 | 12/04/2019 |
|  | Controlador | C\_ Registrar\_Platillo | RSC | 05/04/2019 | 12/04/2019 |
| Registrar usuario |  | Modelo | M\_Registrar\_usuario | RMJ | 05/04/2019 | 12/04/2019 |
| 1 | Vista | V\_Registrar usuario | RSC | 05/04/2019 | 12/04/2019 |
|  | Controlador | C\_Registrar \_usuario | ACS | 05/04/2019 | 12/04/2019 |
| Registrar Cliente |  | Modelo | M\_Registrar\_Cliente | ACS | 05/04/2019 | 12/04/2019 |
| 1 | Vista | V\_Registrar\_ Cliente | RMJ | 05/04/2019 | 12/04/2019 |
|  | Controlador | C\_Registrar\_ Cliente | RSC | 05/04/2019 | 12/04/2019 |
| Registrar Dirección | 1 | Modelo | M\_Registrar\_direccion | ACS | 05/04/2019 | 12/04/2019 |
| Vista | V\_Registrar\_Cliente | RSC | 05/04/2019 | 12/04/2019 |
| Controlador | C\_Registrar\_ direccion | RSC | 05/04/2019 | 12/04/2019 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requerimiento | Incremento | Tipo Clase | Nombre Clase | Responsable | Fecha | Fecha Fin |
| Inicio |
| Registrar Ingredientes | 2 | Modelo | M\_Registrar\_Ingre | FGA | 12/04/2019 | 26/04/2019 |
|  | Vista | V\_Registrar\_ Ingre | RSC | 12/04/2019 | 26/04/2019 |
|  | Controlador | C\_Registrar\_ Ingre | RMJ | 12/04/2019 | 26/04/2019 |
| Eliminar platillo | 2 | *Modelo* | M\_eliminar\_platillo | FGA | 12/04/2019 | 26/04/2019 |
|  | *Vista* | M\_Registrar\_Platillo | RSC | 12/04/2019 | 26/04/2019 |
|  | Controlador | C\_Registrar\_Platillo | FGA | 12/04/2019 | 26/04/2019 |
| Eliminar cliente | 2 | Modelo | M\_Registrar\_Cliente | FGA | 12/04/2019 | 26/04/2019 |
| Vista | V\_eliminar\_cliente | ACE | 12/04/2019 | 26/04/2019 |
| Controlador | C\_Registrar\_Cliente | ACE | 12/04/2019 | 26/04/2019 |
| Realizar pedido | 2 | Modelo | M\_realizar\_pedido | RSC | 12/04/2019 | 26/04/2019 |
| Vista | V\_realizar\_pedido | RSC | 12/04/2019 | 26/04/2019 |
| Controlador | C\_realizar\_pedido | RMJ | 12/04/2019 | 26/04/2019 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requerimiento | Incremento | Tipo Clase | Nombre Clase | Responsable | Fecha | Fecha Fin |
| Inicio |
| Registrar Proveedor |  | Modelo | M\_Registrar\_Prove | RSC | 26/04/2019 | 27/05/2019 |
| 3 | Vista | V\_Registrar\_Prove | ACS | 26/04/2019 | 27/05/2019 |
|  | Controlador | C\_Registrar\_Prove | RMJ | 26/04/2019 | 27/05/2019 |
| Mostrar carta |  | Modelo | M\_Mostrar\_ Carta | RSC | 26/04/2019 | 27/05/2019 |
| 3 | Vista | V\_Mostrar\_ Carta | RMJ | 26/04/2019 | 27/05/2019 |
|  | Controlador | C\_Mostrar\_ Carta | FGA | 26/04/2019 | 27/05/2019 |
| Mostrar categoría cliente |  | Modelo | M\_Mostrar\_ categoriaCli | RSC | 26/04/2019 | 27/05/2019 |
| 3 | Vista | V\_Registrar\_Cliente | RMJ | 26/04/2019 | 27/05/2019 |
|  | Controlador | C\_Mostrar\_ categoriaCli | FGA | 26/04/2019 | 27/05/2019 |
| Mostrar categoría platilllos |  | Modelo | M\_Mostrar\_ categoriaPlat | ACS | 26/04/2019 | 27/05/2019 |
| 3 | Vista | V\_Registrar\_Platillo | ACS | 26/04/2019 | 27/05/2019 |
|  | Controlador | C\_Mostrar\_ categoriaPlat | RSC | 26/04/2019 | 27/05/2019 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requerimiento | Incremento | Tipo Clase | Nombre Clase | Responsable | Fecha | Fecha Fin |
| Inicio |
| Mostrar Promoción |  | Modelo | M\_mostrar\_promocion | RSC | 28/05/2019 | 17/06/2019 |
| 4 | Vista | V\_mostrar\_promocion | RMJ | 28/05/2019 | 17/06/2019 |
|  | Controlador | C\_mostrar\_promocion | RSC | 28/05/2019 | 17/06/2019 |
| Stock de platillos |  | Modelo | M\_mostrarStock\_ingredientes | RSC | 28/05/2019 | 17/06/2019 |
| 4 | Vista | V\_mostrarStock\_ingredientes | RMJ | 28/05/2019 | 17/06/2019 |
|  | Controlador | C\_mostrarStock\_ingredientes | RSC | 28/05/2019 | 17/06/2019 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requerimiento | Incremento | Tipo Clase | Nombre Clase | Responsable | Fecha | Fecha Fin |
| Inicio |
| Pagar |  | Modelo | M\_registrar\_pago | RSC | 18/06/2019 | 04/07/2019 |
| 5 | Vista | V\_registrar\_pago | RMJ | 18/06/2019 | 04/07/2019 |
|  | Controlador | C\_registrar\_pago | RSC | 18/06/2019 | 04/07/2019 |
| Comprobante de pago |  | Modelo | M\_mostrar\_comprobante | RSC | 18/06/2019 | 04/07/2019 |
| 5 | Vista | V\_mostrar\_comprobante | RMJ | 18/06/2019 | 04/07/2019 |
|  | Controlador | C\_mostrar\_comprobante | RSC | 18/06/2019 | 04/07/2019 |

## GESTION DEL PRODUCTO Y PROYECTO

Proyecto: Esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único.

Producto: Es un artículo producido, cuantificable y que puede ser un elemento terminado o un componente.

A diferencia del proyecto, el ciclo de vida del producto, está centrado en el entregable, el producto en sí mismo, siendo sus fases diferentes: Introducción, Crecimiento, Madurez y Retiro.

Es decir, el proyecto puede haber ya finalizado, pero no así el ciclo de vida del producto que vivirá aún hasta que deje de producirse.

* + - El ciclo de vida del producto es más largo que el ciclo de vida del proyecto
    - El ciclo de vida del proyecto tiene una fecha fin definida, el ciclo de vida del producto puede tenerla o no.

## Fase inicial

Durante primera parte de la elaboración del software se tenía la necesidad de automatizar el proceso de pedido del restaurante “Dona Rufina” teniendo que para el desarrollo de este proyecto tenemos que iniciar de 0 sin previos avanzas o antecedentes de este como principal necesidad tenemos que.

* + - * El software debe operar a tiempo real para un exacto funcionamiento
      * Debe tener un login de administrador.

## Fase de elaboración

Con los siguientes requerimientos elaboramos nuestro plan de desarrollo del software

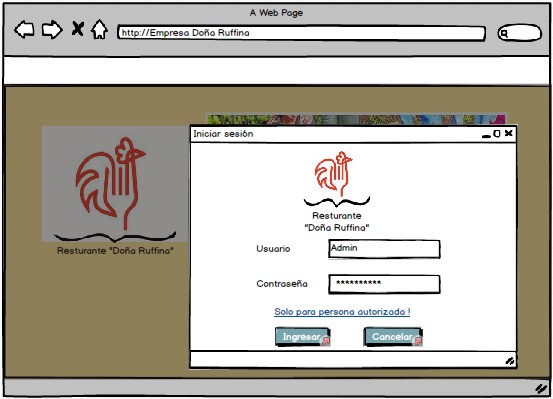
* + - * Mostrar platillos
      * Dar de alta a un usuario
      * Gestionar los diferentes platillos e ingredientes que se tiene en el almacén.

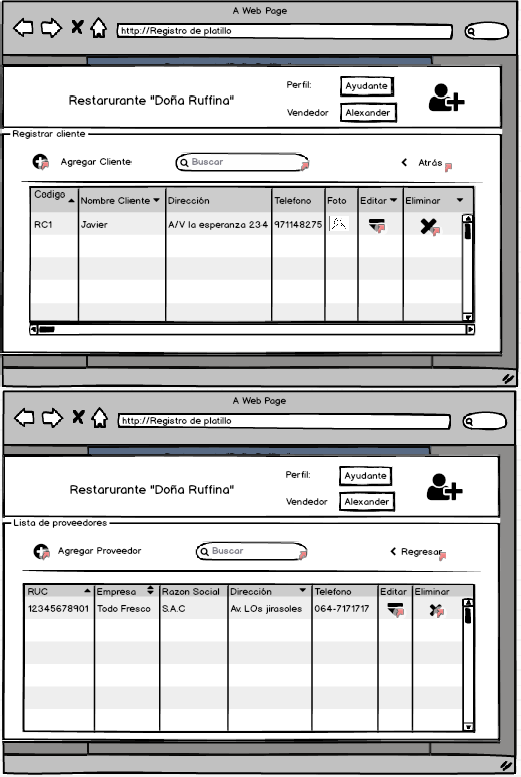
## DESARROLLO DE CICLO DE VIDA DEL SOFTWARE

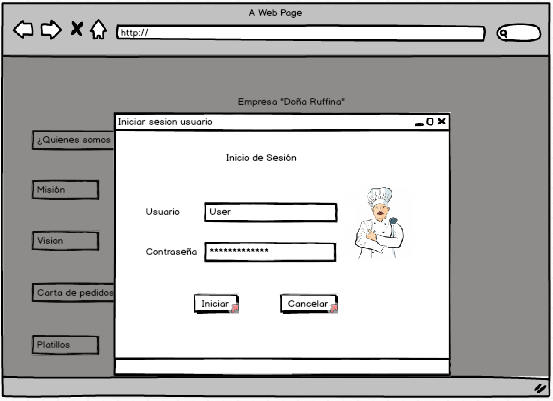
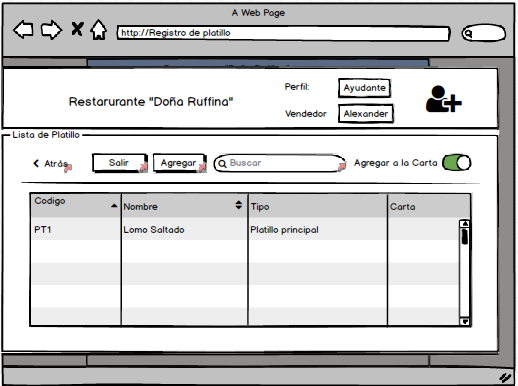
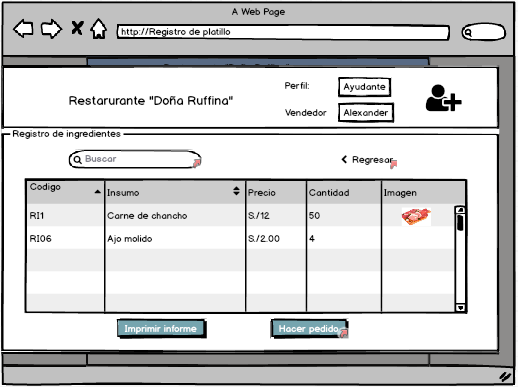
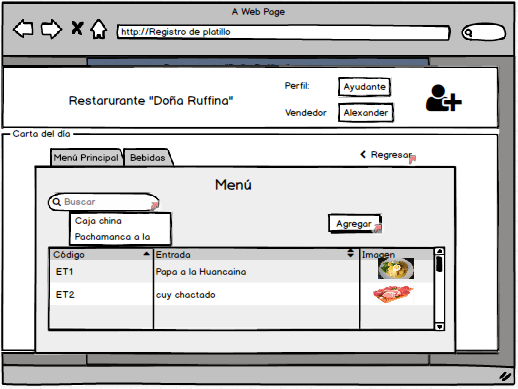
## Prototipo

*Ilustración 12 Prototipos de Software*













## PLAN DE PRUEBAS

## INCREMENTO

*Ilustración 13 Incremento*

**

Es necesario el desarrollo de un plan de pruebas, para poder describir el ámbito del refuerzo de prueba general y proporciona un registro del proceso de planificación de pruebas para que se ajuste a las necesidades del equipo. Ello normalmente se emplea para evaluar los requisitos, riesgos, casos de pruebas y entornos de pruebas que hay que probar, los objetivos de negocio y calidad las planificaciones de pruebas y otros elementos, por ello presentamos lo siguiente.

### **3.6.1.1. INCREMENTO 1**

Tabla 3- Plan de pruebas del restaurant "Doña Rufina"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | **DATOS DE ENTRADA**  **NORMAL** | **RESULTADO ESPERADO**  **NORMAL** | **DATOS DE ENTRADA**  **ANÓMALO** | **RESULTADO ESPERADO**  **ANÓMALO** | **INTENTOS** | | |
| **1** | **2** | **3** |
| Registrar cliente | * Nombre * Apellidos * Teléfono * Email * Categoría * Dirección | Cliente registrado correctamente | Conjunto de caracteres incongruentes | Cliente no registrado |  |  |  |
| Registrar usuario | * Nombre * Apellido * Rol * Contraseña | Administrador registrado correctamente | Conjunto de caracteres incongruentes en apellidos | usuario no registrado |  |  |  |
| Login in Administrador | Nombre de usuario y contraseña. | Usuario conforme | * Usuario incorrecto * Contraseña   incorrecta | Usuario no válido.  Contraseña no válida. |  |  |  |
| Registrar Platillo | Ingresar ingredientes | Ingrediente registrado  correctamente | No se selecciona ningún  ingrediente | No se registra el platillo |  |  |  |
| Registrar Ingredientes | * Nombre * Stock | Ingredientes registrados  correctamente | Ingresar otra vez el mismo  ingrediente | No se actualiza ingrediente y  stock |  |  |  |

Tabla 4-Cumplimiento de pruebas del restaurant "Doña Rufina" - Resultado

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | **DATOS DE ENTRADA**  **NORMAL** | **RESULTADO ESPERADO NORMAL** | **DATOS DE ENTRADA ANÓMALO** | **RESULTADO ESPERADO ANÓMALO** | **INTENTOS** | | |
| **1** | **2** | **3** |
| **CU1** | **✔** | **✔** | **✔** | **✖** | **✖** | **✔** | **✔** |
| **CU2** | **✔** | **✔** | **✔** | **✖** | **✖** | **✔** | **✔** |
| **CU3** | **✔** | **✔** | **✔** | **✔** | **✔** | **✖** | **✔** |
| **CU4** | **✔** | **✔** |  |  | **✔** | **✔** | **✔** |
| **CU5** | **✔** | **✔** | **✖** | **✖** | **✖** | **✖** | **✔** |
| **%CUMPLIMIENTO DE PRUEBAS** | | | | | **42.86%** | **71.43%** | **100%** |

### **3.6.1.2. INCREMENTO 2**

Tabla 5- Plan de prueba Incremento 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | **DATOS ENTRADA NORMAL** | **RESULTADO ESPERADO NORMAL** | **DATOS DE ENTRADA ANÓMALO** | **RESULTADO ESPERADO ANÓMALO** | **INTENTOS** | | |
| **1** | **2** | **3** |
| Realizar Pedido | * Cantidad de comida * Selección de comida * Fecha de creación * Fecha de entrega * Dirección de entrega | Pedido registrado correctamente | No ingresar la comida ni el nombre del cliente | Pedido no registrado |  |  |  |
| Eliminar cliente | * Ingresar el nombre | Cliente eliminado | Ingresar datos incongruentes | Cliente no eliminado |  |  |  |
| Eliminar platillo | * Nombre * Descripción * Precio * Categoría | Platillo eliminado correctamente | Seleccionar el mismo nombre | Platillo no se elimina |  |  |  |
| Registrar Ingredientes | * Nombre * Stock | Ingredientes registrados correctamente | Ingresar otra vez el mismo ingrediente | No se actualiza ingrediente y stock |  |  |  |

Tabla 6- Plan de prueba Incremento 2 - Resultado

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | **DATOS DE ENTRADA**  **NORMAL** | **RESULTADO ESPERADO NORMAL** | **DATOS DE ENTRADA ANÓMALO** | **RESULTADO ESPERADO ANÓMALO** | **INTENTOS** | | |
| **1** | **2** | **3** |
| **CU1** | **✔** | **✔** |  |  | **✔** |  |  |
| **CU2** | **✔** | **✔** |  |  | **✔** |  |  |
| **CU3** | **✖** | **✖** | **✔** | **✔** | **✖** |  |  |
| **CU4** | **✔** | **✔** | **✖** | **✖** | **✖** |  |  |
| **CU5** | **✔** | **✔** | **✔** | **✖** | **✖** |  |  |
| **%CUMPLIMIENTO DE PRUEBAS** | | | | | **45.45%** |  |  |

### **. INCREMENTO 3**

Tabla 5- Plan de prueba Incremento 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | **DATOS ENTRADA NORMAL** | **RESULTADO ESPERADO NORMAL** | **DATOS DE ENTRADA ANÓMALO** | **RESULTADO ESPERADO ANÓMALO** | **INTENTOS** | | |
| **1** | **2** | **3** |
| Mostrar Carta | * Tipo de platillos * Precio de cada platillo | Búsqueda realizada |  |  |  |  |  |
| Registrar Proveedor | RUC o DNI  Razón social  Representante  Email  Teléfono  Celular del representante  Dirección | Proveedor registrado correctamente | Conjunto de  caracteres  incongruentes | Proveedor no registrado |  |  |  |
| Mostrar categoría de clientes | * Nombre de categorías de clientes | Búsqueda realizada |  |  |  |  |  |
| Mostrar categoría de platillos | * Nombre del categorías de platillos | Búsqueda realizada |  |  |  |  |  |

Tabla 6- Plan de prueba Incremento 3 - Resultado

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | **DATOS DE ENTRADA**  **NORMAL** | **RESULTADO ESPERADO NORMAL** | **DATOS DE ENTRADA ANÓMALO** | **RESULTADO ESPERADO ANÓMALO** | **INTENTOS** | | |
| **1** | **2** | **3** |
| **CU1** | **✔** | **✔** |  |  | **✔** |  |  |
| **CU2** | **✔** | **✔** |  |  | **✔** |  |  |
| **CU3** | **✖** | **✖** | **✔** | **✔** | **✖** |  |  |
| **CU4** | **✔** | **✔** | **✖** | **✖** | **✖** |  |  |
| **%CUMPLIMIENTO DE PRUEBAS** | | | | | **50.00%** |  |  |

### **3.6.1.3. CASO DE USO SELECCIONADO**

Tabla 7- Plan de prueba del caso de uso seleccionado

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | **DATOS DE ENTRADA**  **NORMAL** | **RESULTADO ESPERADO NORMAL** | **DATOS DE ENTRADA ANÓMALO** | **RESULTADO ESPERADO ANÓMALO** | **INTENTOS** | | |
| **1** | **2** | **3** |
| Registrar cliente | * Nombre * Apellidos * Teléfono * Email * Categoría * Dirección | Cliente registrado correctamente | Conjunto de caracteres incongruentes | Cliente no registrado |  |  |  |
| Eliminar cliente | * Ingresar el nombre | Cliente eliminado | Ingresar datos incongruentes | Cliente no eliminado |  |  |  |
| Modificar clientes | * Nombre * Apellidos * Teléfono * Email * Categoría * Dirección | Cliente modificado correctamente | Ingresar datos incongruentes | No se pudo modificar los datos del cliente |  |  |  |

Tabla 8-Plan de prueba del caso seleccionado - resuelto

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | **DATOS DE ENTRADA**  **NORMAL** | **RESULTADO ESPERADO NORMAL** | **DATOS DE ENTRADA ANÓMALO** | **RESULTADO ESPERADO ANÓMALO** | **INTENTOS** | | |
| **1** | **2** | **3** |
| **CU1** | **✔** | **✔** | **✔** | **✖** | **✖** | **✔** | **✔** |
| **CU2** | **✔** | **✔** | **✔** | **✖** | **✖** |  |  |
| **CU3** | **✔** | **✔** | **✔** | **✔** | **✔** |  |  |
| **%CUMPLIMIENTO DE PRUEBAS** | | | | | **100%** |  |  |

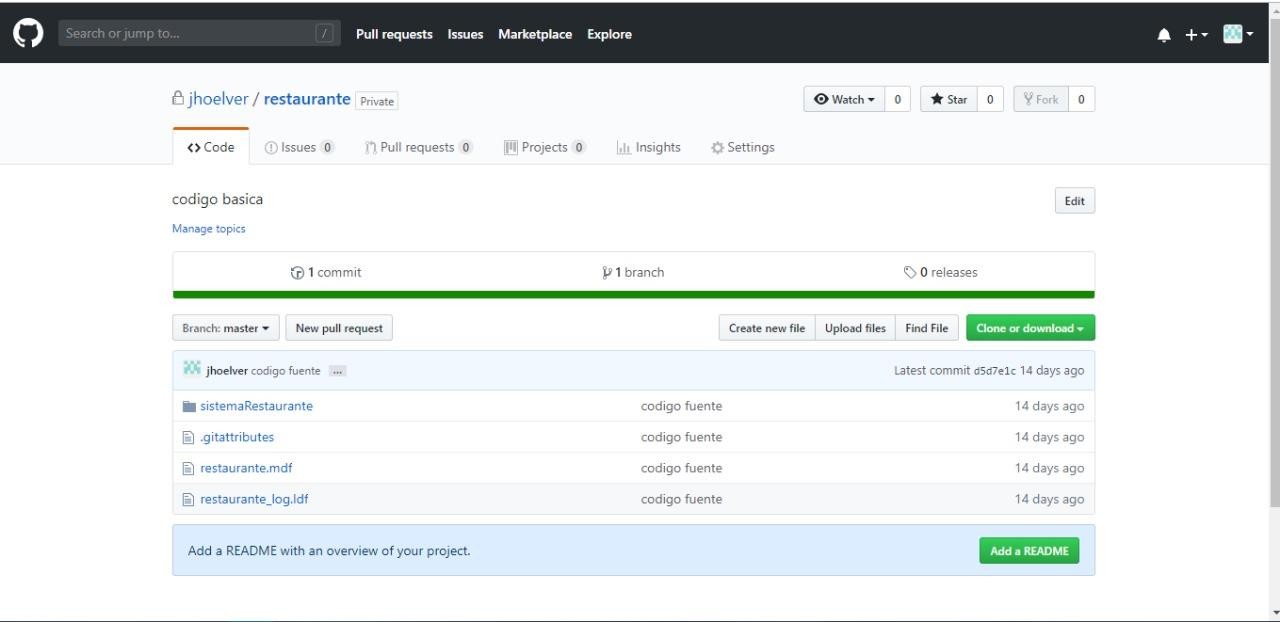
## PROCEDIMIENTOS ALMACENADOS

Tabla 9-Procedimientos almacenados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Tabla | Increment | Litsa de Procesos Almacenados | Responsable |
| 1 | Usuarios | 1 | SP\_A\_Tabla\_ Administrador | RSC |
| 2 | 1 | SP\_C\_Tabla\_ Administrador | ACS |
| 3 | 1 | SP\_M\_Tabla\_ Administrador | RMJ |
| 4 | 1 | SP\_E\_Tabla\_ Administrador | HGJ |
| 5 | Categoria (ingredientes) | 1 | SP\_A\_Tabla\_ Administrador | RSC |
| 6 | 1 | SP\_C\_Tabla\_ Categoria | ACS |
| 7 | 2 | SP\_M\_Tabla\_ Categoria | RMJ |
| 8 | 2 | SP\_E\_Tabla\_ Categoria | HGJ |
| 9 | Cliente | 1 | SP\_A\_Tabla\_Cliente | FGA |
| 10 | 1 | SP\_C\_Tabla\_Cliente | ACS |
| 11 | 2 | SP\_M\_Tabla\_Cliente | RMJ |
| 12 | 2 | SP\_E\_Tabla\_ Cliente | HGJ |
| 13 | Insumos | 1 | SP\_A\_Tabla\_ Insumos | RSC |
| 14 | 1 | SP\_C\_Tabla\_ Insumos | ACS |
| 15 | 1 | SP\_M\_Tabla\_ Insumos | RMJ |
| 16 | 1 | SP\_E\_Tabla\_ Insumos | RSC |
| 17 | Proveedor | 3 | SP\_A\_Tabla\_ Proveedor | FGA |
| 18 | 3 | SP\_C\_Tabla\_ Proveedor | ACS |
| 19 | 3 | SP\_M\_Tabla\_ Proveedor | RSC |
| 20 | 3 | SP\_E\_Tabla\_ Proveedor | HGJ |
| 21 | Platillos | 1 | SP\_A\_Tabla\_ Platillos | FGA |
| 22 | 1 | SP\_C\_Tabla\_ Platillos | ACS |
| 23 | 2 | SP\_M\_Tabla\_ Platillos | RSC |
| 24 | 2 | SP\_E\_Tabla\_ Platillos | RSC |

## GESTION DE LA CONFIGURACION

## 3.8.1. Github



## 

## PLAN DE PRUEBAS

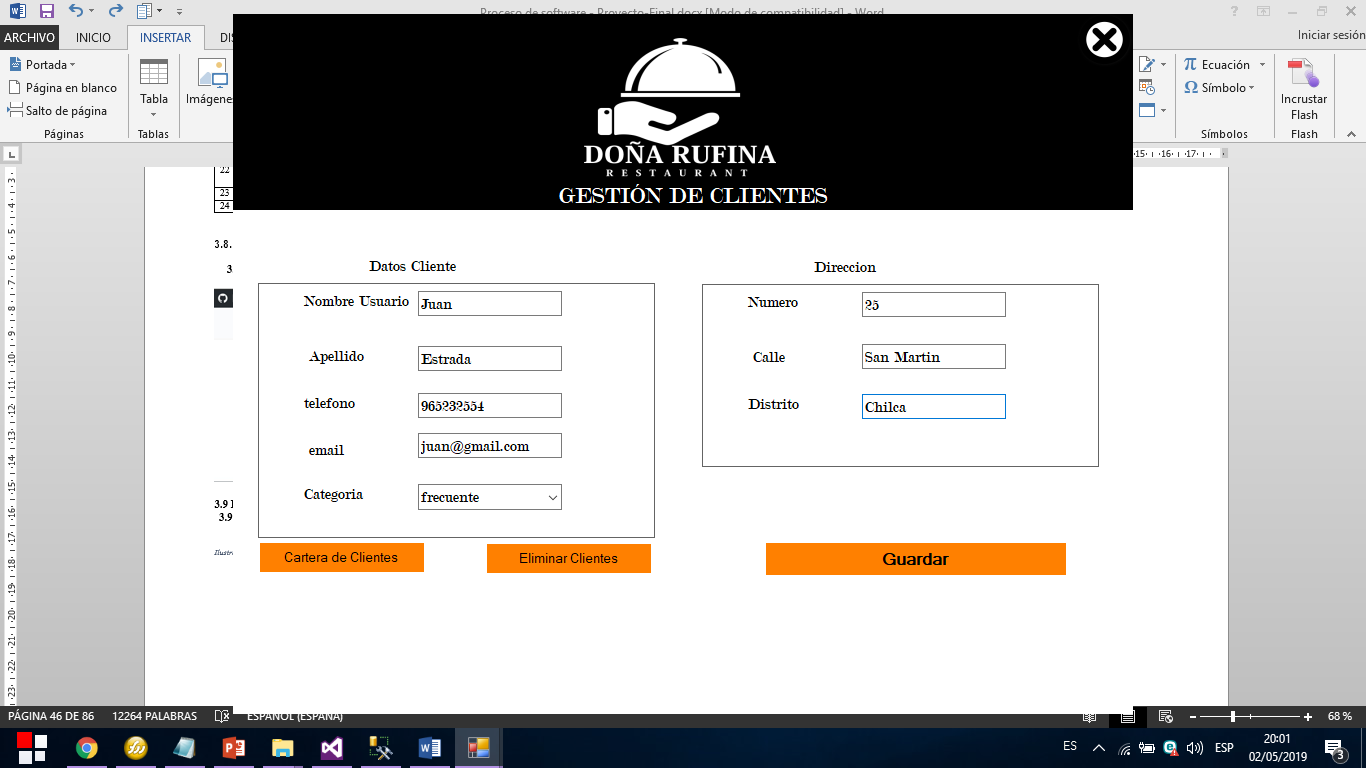
### **PLAN DE PRUEBAS DEL INCREMENTO**

### **PLAN DE PRUEBAS DEL INCREMENTO 1**

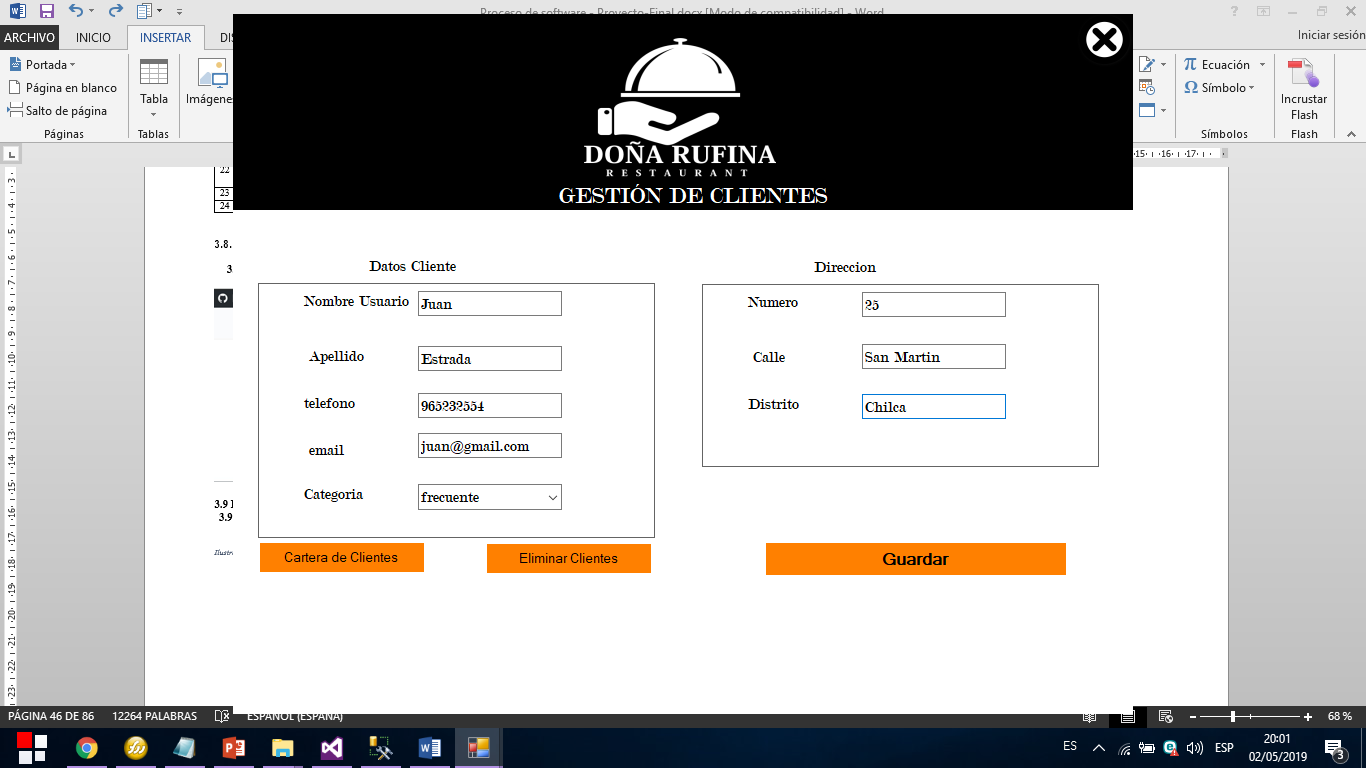
*Ilustración 14 Plan de pruebas del módulo Registrar cliente*

**REGISTRAR CLIENTE**

Dato de entrada normal



Resultado esperado normal



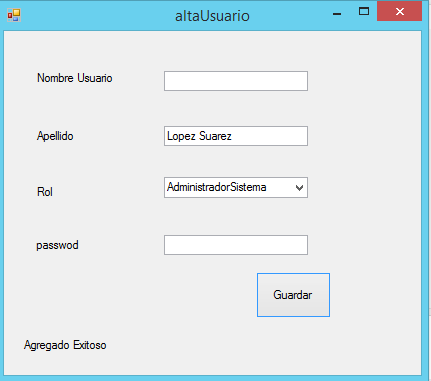
Dato de entrada anómalo



Resultado esperado anómalo

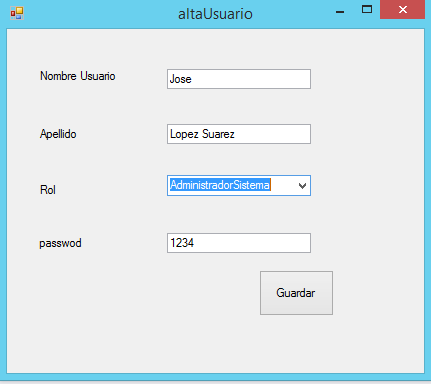


*Ilustración 15 Plan de pruebas del módulo Registrar administrador*



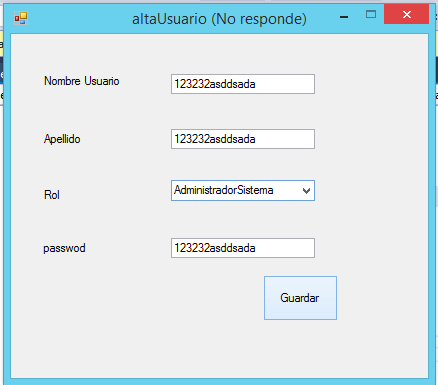
**REGISTRAR ADMINISTRADOR**

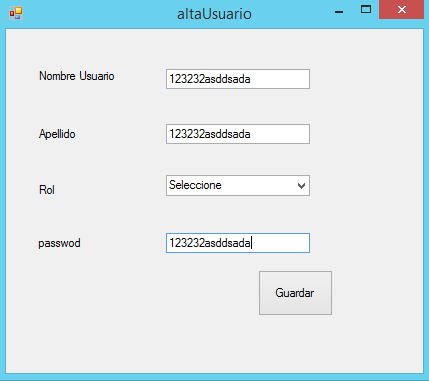
Dato de entrada normal



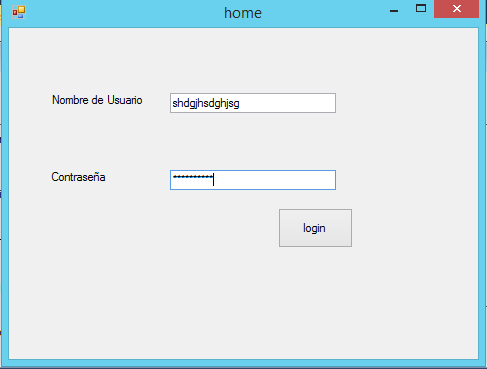
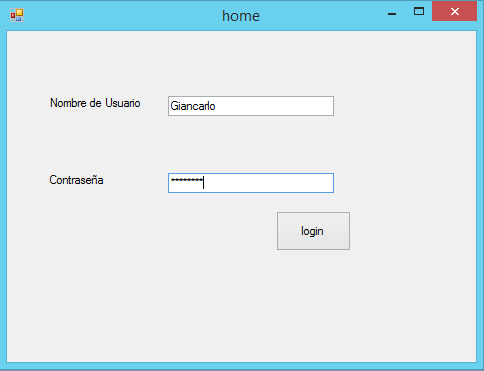
Resultado esperado normal

Dato de entrada anómalo





Resultado esperado anómalo



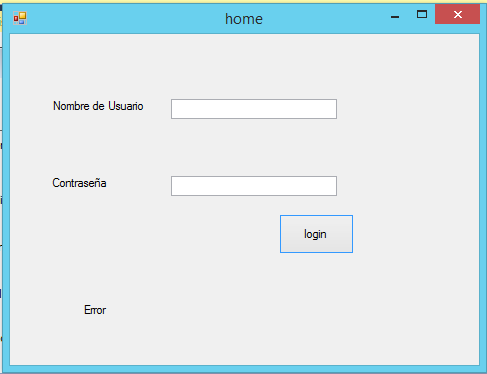
*Ilustración 16 Plan de pruebas del módulo Login administrador*

# LOGIN ADMINISTRADOR

##### Dato de entrada normal

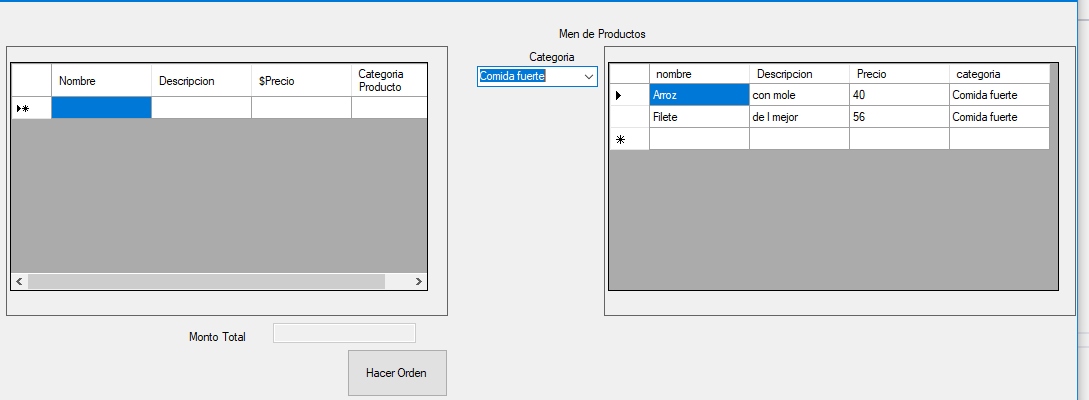
Resultado esperado normal

Dato de entrada anómalo



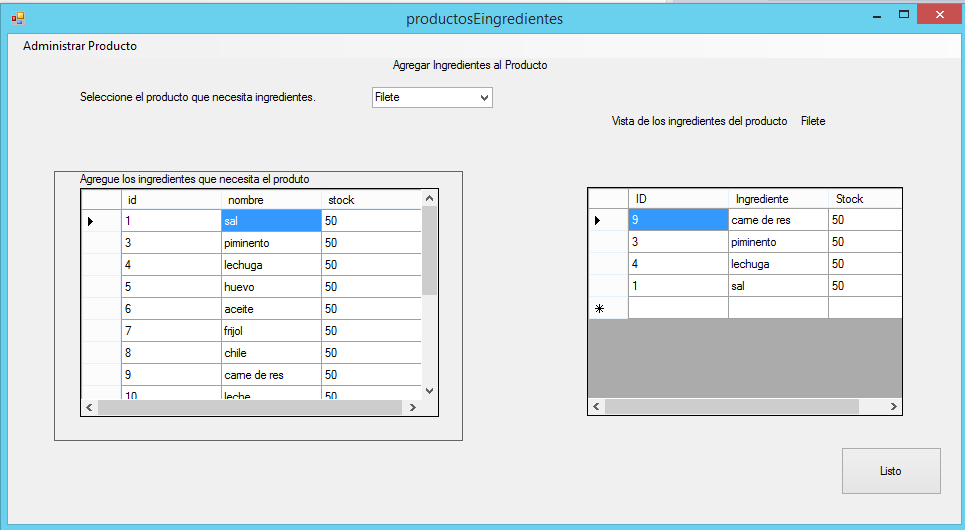
Resultado esperado anómalo

*Ilustración 17 Plan de pruebas del catalogo*



**MOSTRAR CATALOGO**

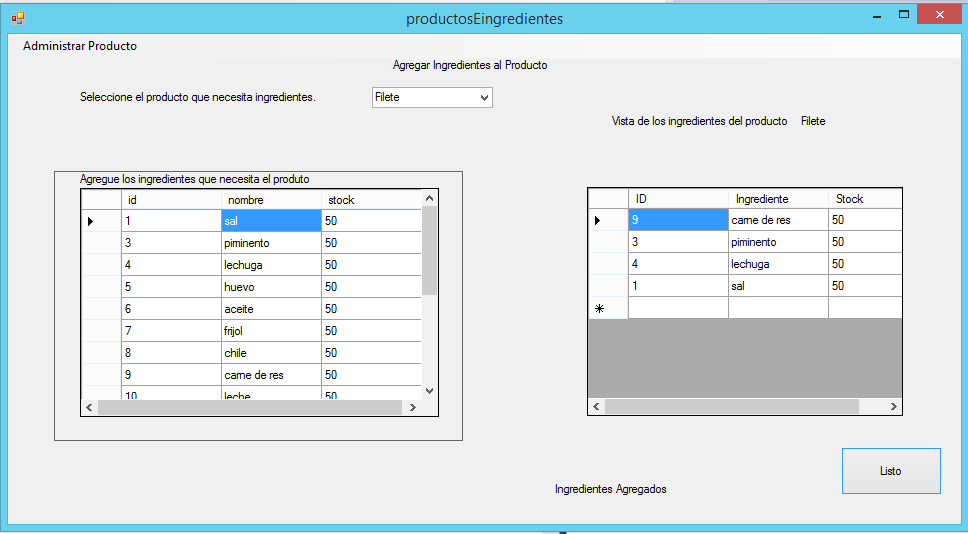
*Ilustración 18 Plan de pruebas del módulo Registrar platillos*



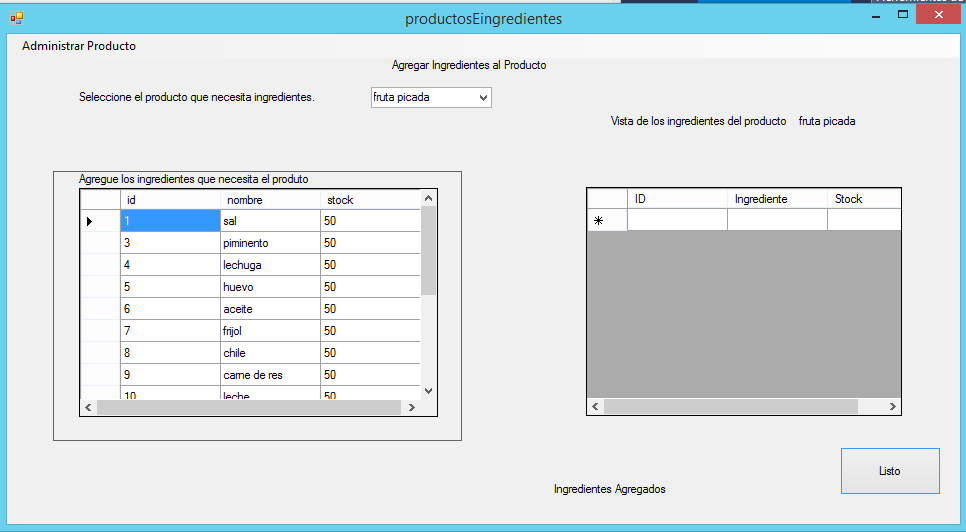
**REGISTRAR PLATILLOS**

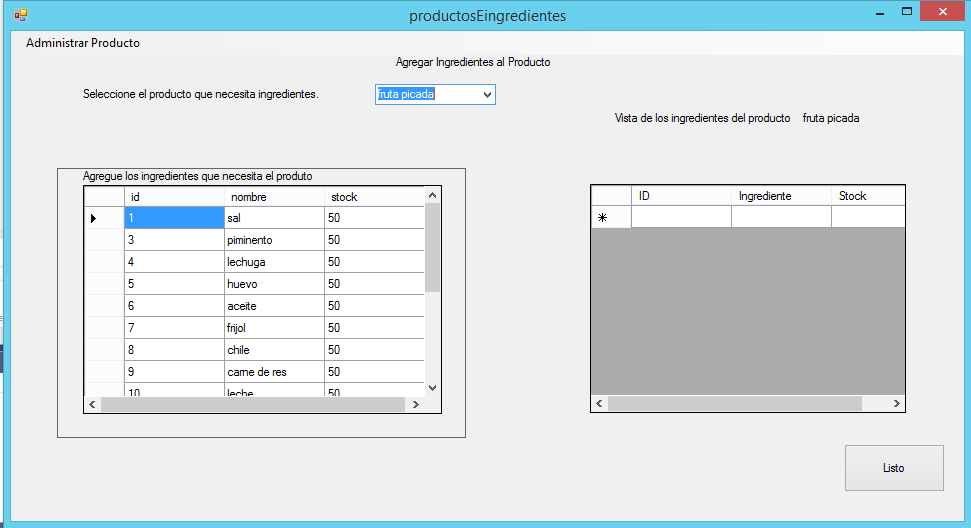
Dato de entrada normal

Resultado esperado normal

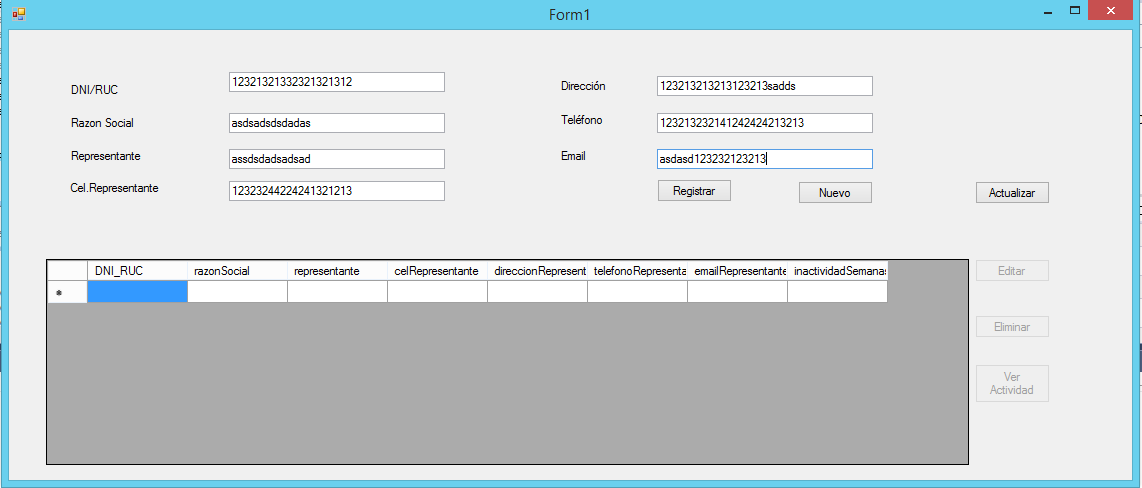
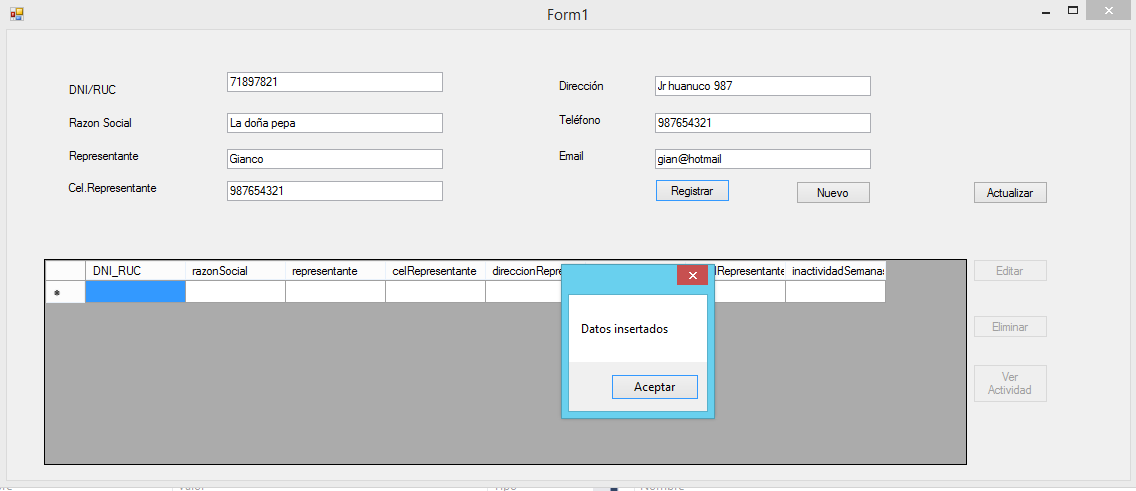
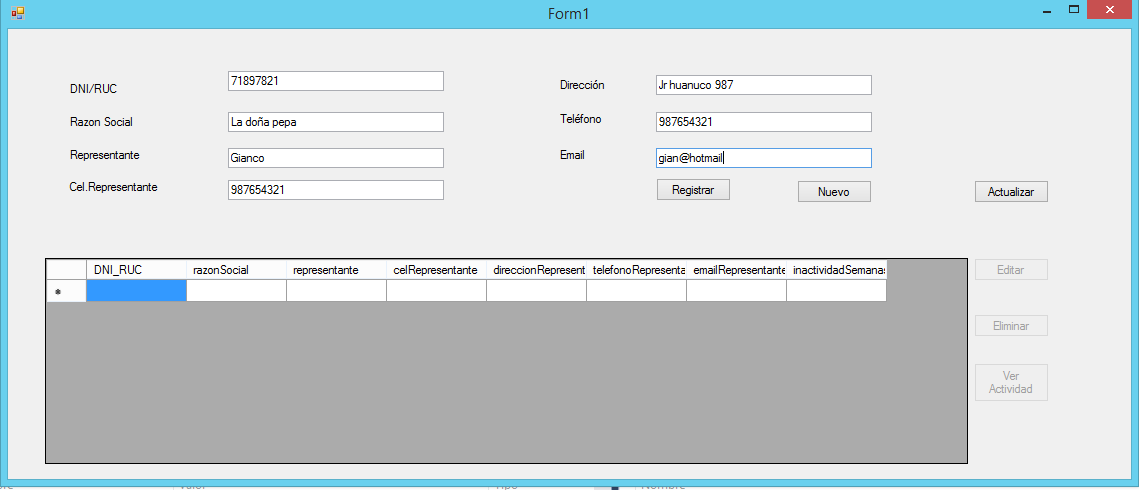


Dato de entrada anómalo





Resultado esperado anómalo



*Ilustración 19 Plan de pruebas del módulo Registrar proveedor*

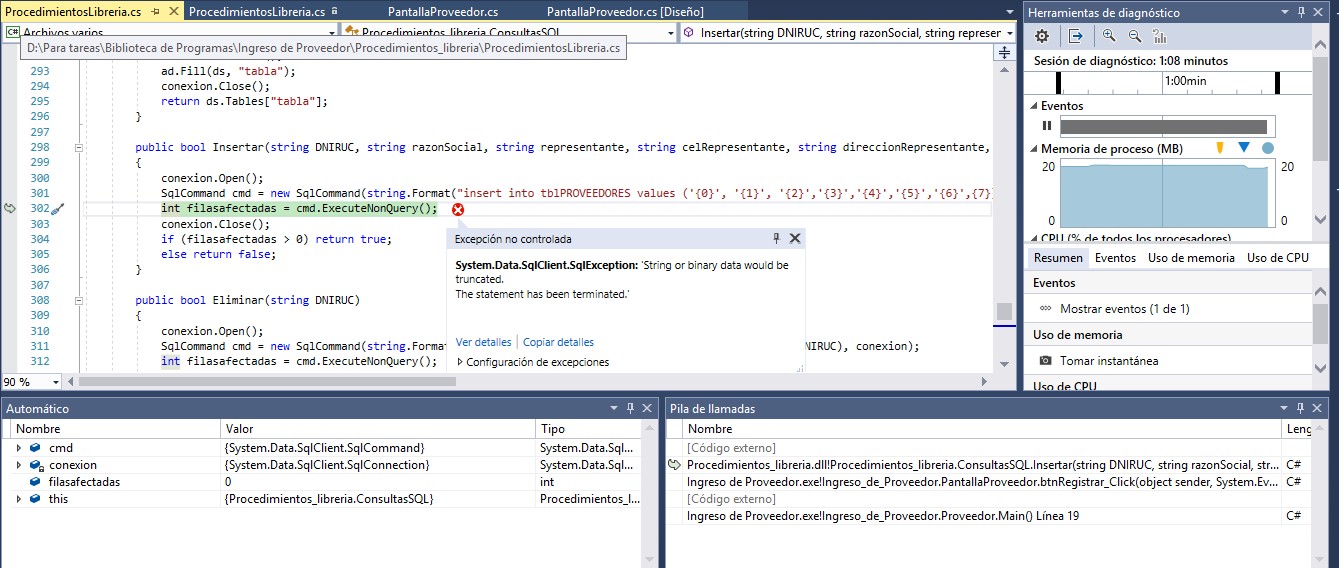
# REGISTRAR PROVEEDOR

##### Dato de entrada normal

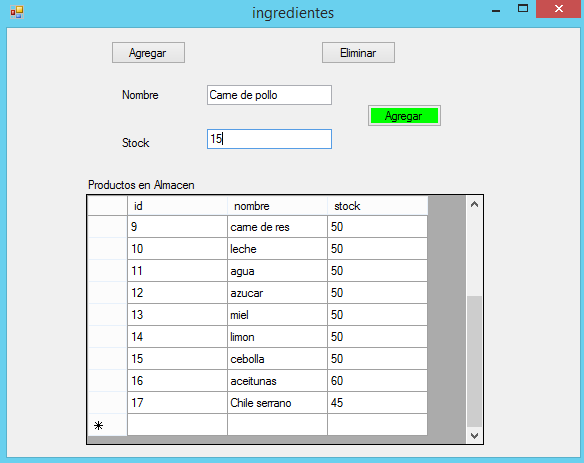
Resultado esperado normal

Dato de entrada anómalo

Resultado esperado anómalo



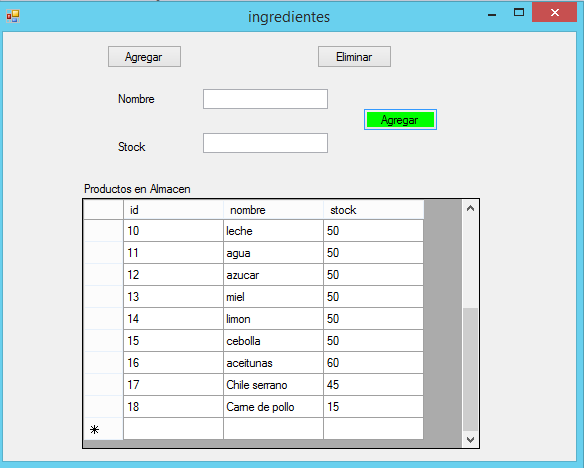
*Ilustración 20 Plan de pruebas del módulo Registrar ingredientes*



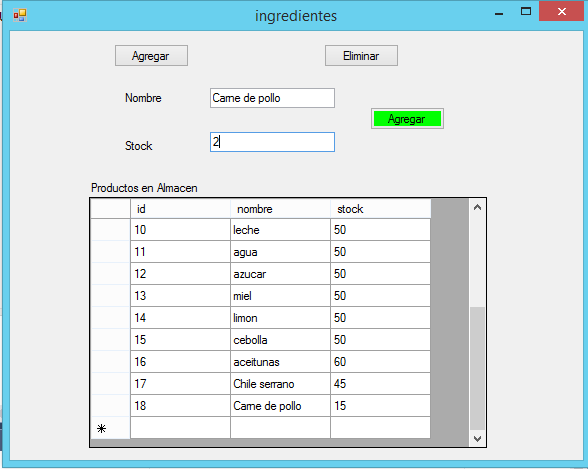
**REGISTRAR INGREDIENTES**

Dato de entrada normal

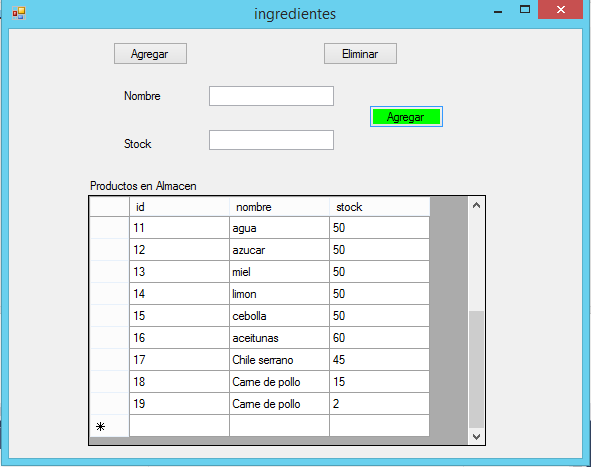
Resultado esperado normal



Dato de entrada anómalo



Resultado esperado anómalo



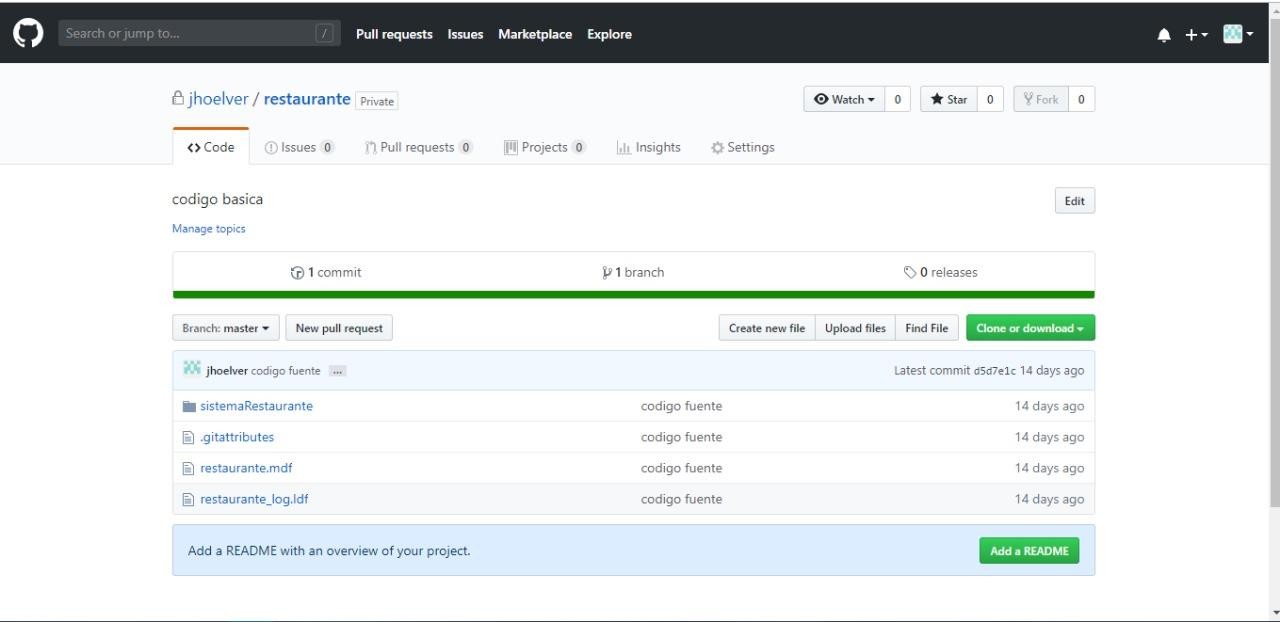
* + - 1. **PLAN DE PRUEBAS DEL INCREMENTO 2**

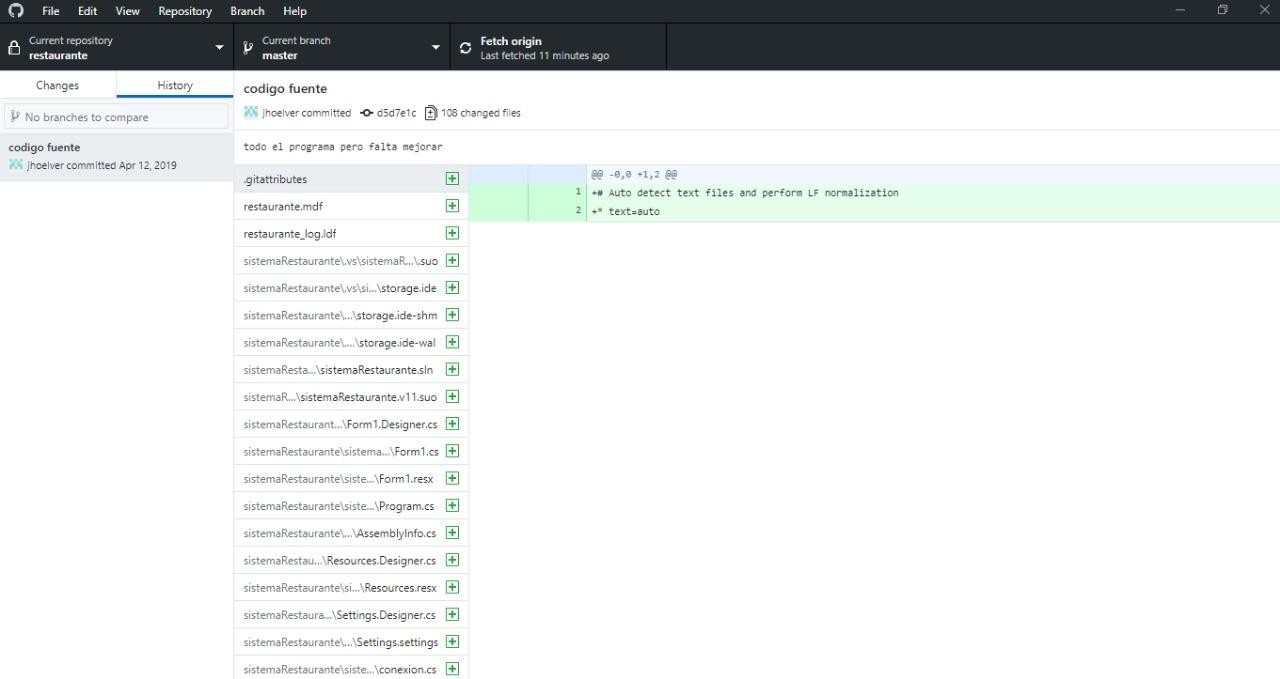
|  |
| --- |
| **Mostrar Carta**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Mostrar Promoción**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Realizar Pedido**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Eliminar cliente**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Eliminar Ingredientes**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Eliminar Pedido**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Modificar Producto**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Modificar Ingredientes**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Modificar Proveedor**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Modificar clientes**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Modificar Pedido**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |

* + - 1. **PLAN DE PRUEBAS DEL CASO DE USO SELECCIONADO**

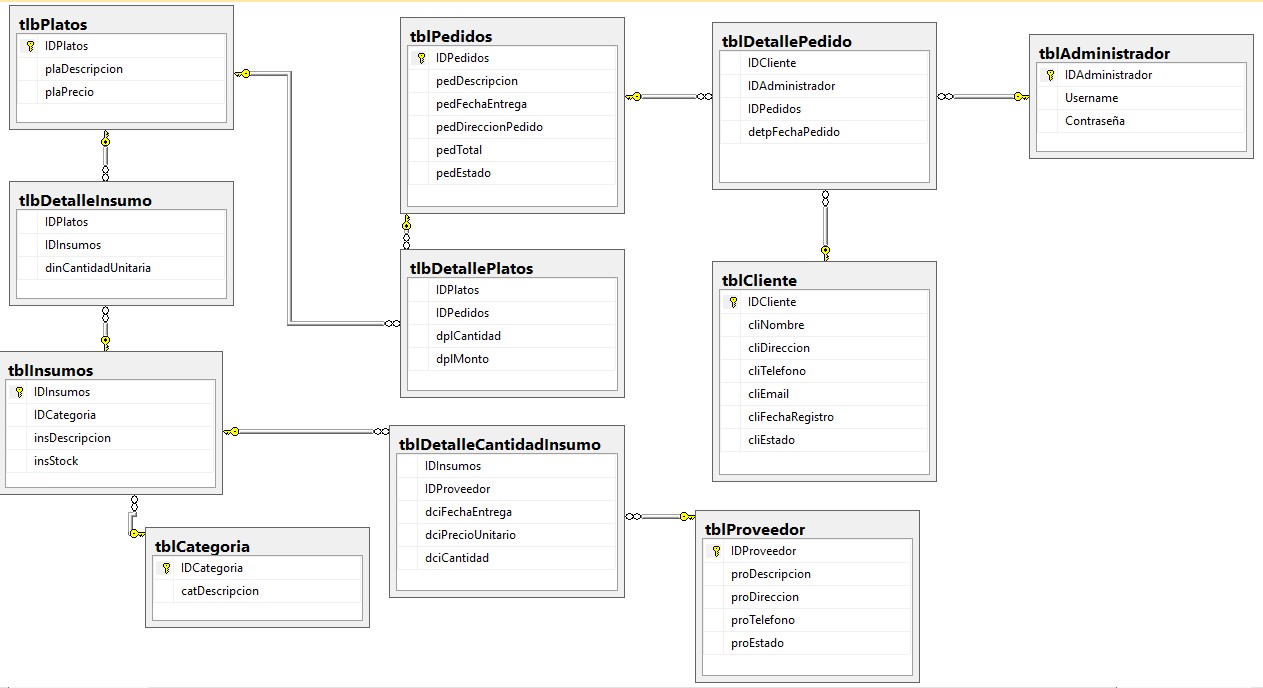
|  |
| --- |
| **Registrar platillo**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |
| **Eliminar platillo**  Datos de entrada normal  Respuesta esperado normal  Datos de entrada anómalo  Respuesta esperado anómalo |

## Evidencias de código almacenado en github





## META 1: MODELAMIENTO



#### SCRIPT DE PROCEDIMIENTOS:

- **CREACIÓN DE BASE DE DATOS:**

USE master

CREATE DATABASE BD\_Doña\_Rufina

CREATE TABLE tblAdministrador(

IDAdministrador varchar(2) primary key , Username varchar(10) NULL,

Contraseña varchar(10) NULL

)

CREATE TABLE tblCategoria(

IDCategoria varchar(6) primary key, catDescripcion varchar(50) NULL

)

CREATE TABLE tblCliente(

IDCliente int primary key, cliNombre varchar(50) NULL, cliDireccion varchar(50) NULL, cliTelefono varchar(9) NULL, cliEmail varchar(50) NULL, cliFechaRegistro datetime null, cliEstado int

CREATE TABLE tblInsumos(

IDInsumos varchar(50) primary key, IDCategoria varchar(6) NULL, insDescripcion varchar(50) NULL, insStock int NULL,

FOREIGN KEY (IDCategoria) REFERENCES tblCategoria (IDCategoria)

)

CREATE TABLE tblPedidos(

IDPedidos varchar(6) primary key, pedDescripcion varchar(50) NULL, pedFechaEntrega datetime NULL, pedDireccionPedido varchar(100) NULL, pedTotal decimal(10, 2) NULL, pedEstado int

)

CREATE TABLE tblDetallePedido( IDCliente int,

IDAdministrador varchar(2), IDPedidos varchar(6), detpFechaPedido date,

FOREIGN KEY (IDCliente) REFERENCES tblCliente (IDCliente), FOREIGN KEY (IDPedidos) REFERENCES tblPedidos (IDPedidos),

FOREIGN KEY (IDAdministrador) REFERENCES tblAdministrador (IDAdministrador),

)

CREATE TABLE tblProveedor(

IDProveedor varchar(4) primary key, proDescripcion varchar(10) NULL, proDireccion varchar(50) NULL, proTelefono varchar(9) NULL, proEstado int

)

CREATE TABLE tblDetalleCantidadInsumo( IDInsumos varchar(50),

IDProveedor varchar(4) , dciFechaEntrega date, dciPrecioUnitario decimal(3,2) null, dciCantidad int null,

FOREIGN KEY (IDInsumos) REFERENCES tblInsumos (IDInsumos), FOREIGN KEY (IDProveedor) REFERENCES tblProveedor (IDProveedor)

)

CREATE TABLE tlbPlatos(

IDPlatos varchar(7) primary key, plaDescripcion varchar(50) NULL, plaPrecio decimal(5, 2) NULL

)

CREATE TABLE tlbDetallePlatos( IDPlatos varchar(7), IDPedidos varchar(6), dplCantidad int, dplMonto decimal (5,2),

FOREIGN KEY (IDPlatos) REFERENCES tlbPlatos (IDPlatos), FOREIGN KEY (IDPedidos) REFERENCES tblPedidos (IDPedidos)

CREATE TABLE tlbDetalleInsumo( IDPlatos varchar(7), IDInsumos varchar(50), dinCantidadUnitaria int,

FOREIGN KEY (IDPlatos) REFERENCES tlbPlatos (IDPlatos), FOREIGN KEY (IDInsumos) REFERENCES tblInsumos (IDInsumos),

)

#### - Procedimiento almacenado Cliente

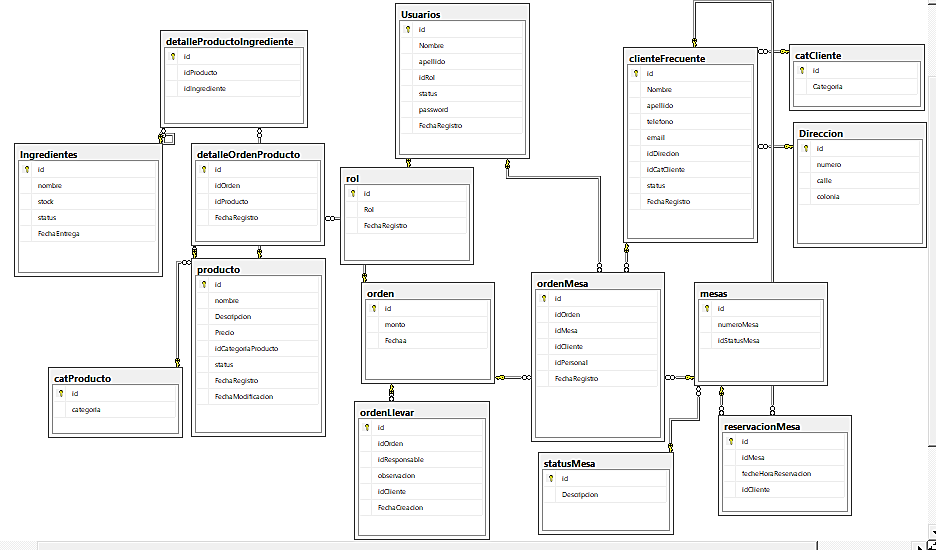
|  |  |
| --- | --- |
| CREATE PROCEDURE  SP\_A\_TABLA\_tblCliente  ( @IDCliente int, @cliNombre  varchar(50),  @cliDireccion varchar(50),  @cliTelefono varchar(9),  @cliEmail varchar(50), @cliFechaRegistro  datetime,  @cliEstado int)  AS BEGIN  INSERT INTO  tblCliente(IDCliente,  cliNombre  ,cliDireccion  ,cliTelefono  ,cliEmail  ,cliFechaRegistro  ,cliEstado)  VALUES  (@IDCliente, @cliNombre  ,@cliDireccion  ,@cliTelefono  ,@cliEmail  ,@cliFechaRegistro  ,@cliEstado);  END | CREATE PROCEDURE  SP\_C\_TABLA\_tblCliente AS  BEGIN  select \* from tblCliente  END |
| CREATE PROCEDURE  SP\_M\_TABLA\_tblCliente (@IDCliente int, @cliDireccion varchar(50),  @cliTelefono varchar(9))  AS BEGIN  UPDATE tblCliente set cliDireccion =  @cliDireccion,  cliTelefono = @cliTelefono Where IDCliente = @IDCliente  END | CREATE PROCEDURE  SP\_E\_TABLA\_tblCliente (@clieEstado int)  AS BEGIN  UPDATE tblCliente set cliEstado = 0  Where cliEstado = @clieEstado  END |

## META 2: MODELAMIENTO

#### SCRIPT DE PROCEDIMIENTOS:

- **CREACIÓN DE BSE DE DATOS**

USE m

****

Master

CREATE DATABASE BD\_Doña\_Rufina

CREATE TABLE [dbo].[catProducto](

[id] [int] IDENTITY(1,1) NOT NULL,

[categoria] [varchar](50) NULL,

CONSTRAINT [PK\_catProducto] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[clienteFrecuente](

[id] [int] IDENTITY(1,1) NOT NULL,

[Nombre] [varchar](50) NULL,

[apellido] [varchar](50) NULL,

[telefono] [varchar](50) NULL,

[email] [varchar](50) NULL,

[idDirecion] [int] NULL,

[idCatCliente] [int] NULL,

[status] [bit] NULL,

[FechaRegistro] [date] NULL,

CONSTRAINT [PK\_clienteFrecuente] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[detalleOrdenProducto](

[id] [int] IDENTITY(1,1) NOT NULL,

[idOrden] [int] NOT NULL,

[idProducto] [int] NOT NULL,

[FechaRegistro] [date] NULL,

CONSTRAINT [PK\_detalleOrdenProducto] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[detalleProductoIngrediente](

[id] [int] IDENTITY(1,1) NOT NULL,

[idProducto] [int] NOT NULL,

[idIngrediente] [int] NOT NULL,

CONSTRAINT [PK\_detalleProductoIngrediente] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Direccion](

[id] [int] IDENTITY(1,1) NOT NULL,

[numero] [varchar](50) NULL,

[calle] [varchar](50) NULL,

[colonia] [varchar](50) NULL,

CONSTRAINT [PK\_Direccion] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Ingredientes](

[id] [int] IDENTITY(1,1) NOT NULL,

[nombre] [varchar](50) NOT NULL,

[stock] [int] NULL,

[status] [bit] NULL,

[FechaeEntrada] [date] NULL,

CONSTRAINT [PK\_Ingredientes] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[mesas](

[id] [int] IDENTITY(1,1) NOT NULL,

[numeroMesa] [int] NULL,

[idStatusMesa] [int] NULL,

CONSTRAINT [PK\_mesas] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[orden](

[id] [int] IDENTITY(1,1) NOT NULL,

[monto] [int] NULL,

[Fechaa] [datetime] NULL,

CONSTRAINT [PK\_orden] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[ordenLlevar](

[id] [int] IDENTITY(1,1) NOT NULL,

[idOrden] [int] NULL,

[idResponsable] [int] NULL,

[observacion] [varchar](50) NULL,

[idCliente] [int] NULL,

[FechaRegistro] [date] NULL,

CONSTRAINT [PK\_ordenLlevar] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[ordenMesa](

[id] [int] IDENTITY(1,1) NOT NULL,

[idOrden] [int] NOT NULL,

[idMesa] [int] NOT NULL,

[idCliente] [int] NOT NULL,

[idPersonal] [int] NOT NULL,

[FechaRegistro] [date] NULL,

CONSTRAINT [PK\_ordenMesa] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[producto](

[id] [int] IDENTITY(1,1) NOT NULL,

[nombre] [varchar](50) NULL,

[Descripcion] [varchar](50) NULL,

[Precio] [int] NULL,

[idCategoriaProducto] [int] NOT NULL,

[status] [bit] NULL,

[FechaRegistro] [date] NULL,

CONSTRAINT [PK\_producto] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[reservacionMesa](

[id] [int] IDENTITY(1,1) NOT NULL,

[idMesa] [int] NULL,

[fecheHoraReservacion] [datetime] NULL,

[idCliente] [int] NULL,

[FechaModificacion] [date] NULL,

CONSTRAINT [PK\_reservacionMesa] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[rol](

[id] [int] IDENTITY(1,1) NOT NULL,

[Rol] [varchar](50) NOT NULL,

CONSTRAINT [PK\_rol] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[statusMesa](

[id] [int] IDENTITY(1,1) NOT NULL,

[Descripcion] [varchar](50) NULL,

[FechaRegistro] [date] NULL,

CONSTRAINT [PK\_statusMesa] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Usuarios](

[id] [int] IDENTITY(1,1) NOT NULL,

[Nombre] [varchar](50) NULL,

[apellido] [varchar](50) NULL,

[idRol] [int] NOT NULL,

[status] [bit] NULL CONSTRAINT [DF\_Usuarios\_status] DEFAULT ((1)),

[password] [varchar](50) NOT NULL,

[FechaRegistro] [date] NULL,

CONSTRAINT [PK\_Usuarios] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

**PROCEDIMIENTOS ALMACENADOS**

|  |  |
| --- | --- |
| create PROCEDURE [dbo].[sp\_A\_table\_ClienteFrecuente]  @nombre varchar (50),  @apellido varchar (50),  @telefono varchar (50),  @email varchar (50),  @idCatCliente int    AS  BEGIN  SET NOCOUNT ON;  Begin Tran transaccion  Begin Try    Declare @idDireccion int  set @idDireccion= (SELECT MAX(id)  from dbo.Direccion);    insert into dbo.clienteFrecuente(nombre,apellido,  teléfono  ,email,idDirecion,idCatCliente,status)  values(@nombre,@apellido,@telefono,@email,  @idDireccion,@idCatCliente,1);    COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_A\_Table\_DetalleIngredienteProducto]  @idProducto int  AS  BEGIN  SET NOCOUNT ON;  Begin Tran transaccion  Begin Try  SELECT dbo.Ingredientes.nombre AS Ingrediente, dbo.Ingredientes.stock  FROM dbo.producto INNER JOIN  dbo.detalleProductoIngrediente ON dbo.producto.id = dbo.detalleProductoIngrediente.idProducto INNER JOIN  dbo.Ingredientes ON dbo.detalleProductoIngrediente.idIngrediente = dbo.Ingredientes.id  where producto.id=@idProducto  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| CREATE PROCEDURE [dbo].[sp\_A\_table\_Direccion]  --datos direccion del cliente  @numero varchar (50),  @calle varchar (50),  @colonia varchar (59)  AS  BEGIN  SET NOCOUNT ON;  Begin Tran transaccion  Begin Try  insert into dbo.Direccion (numero,calle,colonia)  values(@numero,@calle,@colonia);  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_A\_Table\_Ingrediente]  @nombre varchar (59),  @stock int  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  insert into dbo.Ingredientes(nombre,stock,status)  values (@nombre,@stock,1)      COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| create PROCEDURE [dbo].[sp\_A\_Table\_Orden]  -- Se agregan los parameros  @MontoTotal int  AS  BEGIN  -- SET NOCOUNT ON addedos to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  DECLARE @fecha DATETIME = GETDATE()  Begin Try  INSERT INTO orden(monto,fechaa)  VALUES(@MontoTotal,@fecha)  COMMIT TRAN transaccion  End Try  Begin Catch    Rollback TRAN transaccion  End Catch    END | create PROCEDURE [dbo].[sp\_A\_Table\_OrdenLLevar]  --datos direccion del cliente  @idcliente int,  @observacion varchar (50),  @idPersonal int  AS  BEGIN  -- SET NOCOUNT ON added to  prevent extra result sets from  -- interfering with SELECT  statements.  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  declare @TraerIdOrden int  set @TraerIdOrden=(select MAX(id)  as id from orden )    insert into ordenLlevar(idOrden,idResponsable,  observacion,idCliente)  values(@TraerIdOrden,@idPersonal,  @observacion,@idcliente);  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN Transaccion  End Catch  END |
| create PROCEDURE [dbo].[sp\_A\_Table\_OrdenMesaLocal]  --datos direccion del cliente  @idMesa int,  @idcliente int,  @idPersonal int    AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT  statements.  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  declare @TraerIdOrden int  set @TraerIdOrden=(select  MAX(id) as id from orden )    insert into ordenMesa(idOrden,idMesa,idCliente  ,idPersonal) values(@TraerIdOrden,@idMesa,  @idcliente,@idPersonal);    update mesas set  idStatusMesa=2 where id = @idMesa;    COMMIT TRAN transacción End Try  Begin Catch    Rollback TRAN transaccion  End Catch    END | create PROCEDURE [dbo].[sp\_A\_Table\_OrdenProductos]  -- Se agregan los parameros  @NombreProducto varchar(50)  AS  BEGIN  -- SET NOCOUNT ON addedos to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  --Inicia la transaccion    Begin Tran transaccion  declare @TraerIdVenta int  set @TraerIdVenta=(select MAX(id) as id from orden )  declare @TraerIdProducto int  set @TraerIdProducto=(select  id from producto where nombre=@NombreProducto)  Begin Try  INSERT INTO dbo.detalleOrdenProducto(idOrden,idProducto)  VALUES(@TraerIdVenta,  @TraerIdProducto)  --Rollback TRAN transaccion  COMMIT TRAN transaccion  End Try  Begin Catch    Rollback TRAN transaccion  End Catch    END |
| CREATE PROCEDURE  [dbo].[sp\_A\_table\_Producto]  --datos direccion del cliente  @Nombre varchar (50),  @Desc varchar (50),  @Precio int,  @idCat int  AS  BEGIN  -- SET NOCOUNT ON added to  prevent extra result sets from  -- interfering with SELECT  statements.  SET NOCOUNT ON;  --Inicia la transaccion    Begin Tran transaccion  Begin Try  insert into dbo.producto(nombre,Descripcion  ,Precio,idCategoriaProducto,status)  values(@Nombre,@Desc,@Precio,@idCat,1);  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_A\_Table\_Relacion  ProductosEingredinentes]  @idProducto int,  @idIngrediente int  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  insert into detalleProductoIngrediente  (idProducto,idIngrediente)  values (@idProducto,@idIngrediente)      COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| Create PROCEDURE  [dbo].[sp\_A\_Table\_Reservar]  -- Se agregan los parameros  @idMesa int,  @fecha datetime,  @NombreCliente varchar(50)  AS  BEGIN  -- SET NOCOUNT ON addedos to  prevent extra result sets from  -- interfering with SELECT  statements.  SET NOCOUNT ON;  --Inicia la transaccion    Begin Tran transaccion    declare  @TraeridClienteFrecuente int  set  @TraeridClienteFrecuente=(select id from dbo.clienteFrecuente where Nombre=@NombreCliente)  Begin Try  INSERT INTO reservacionMesa  (idMesa,fecheHoraReservacion,idCliente)  VALUES  (@idMesa,@fecha,@TraeridClienteFrecuente)      update mesas set idStatusMesa=3 where id = @idMesa;  --Rollback TRAN transaccion  COMMIT TRAN transaccion  End Try  Begin Catch    Rollback TRAN transaccion  End Catch    END | create PROCEDURE [dbo].[sp\_A\_table\_User]  --datos direccion del cliente  @Nombre varchar (50),  @apellido varchar (50),  @idRol int,    @pass int  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  --Inicia la transaccion    Begin Tran transaccion  Begin Try  insert into Usuarios (Nombre,apellido,  idRol,status,password)  values(@Nombre,@apellido,  @idRol,1,@pass);    COMMIT TRAN transaccion  End Try  Begin Catch    Rollback TRAN transaccion  End Catch    END |
| create procedure [dbo].[sp\_actualizarIngrediente]  @nombre varchar (59),  @stock int  AS  BEGIN    UPDATE dbo.Ingredientes  SET  stock = @stock      WHERE nombre = @nombre  END | Create PROCEDURE [dbo].[sp\_actualizarIngredientes]  -- Se agregan los parameros  @nombreIngrediente varchar (59),  @stockIngrediente int  AS  BEGIN  -- SET NOCOUNT ON addedos to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  --Inicia la transaccion    Begin Tran transaccion    update dbo.Ingredientes  set stock= @stockIngrediente where nombre = @nombreIngrediente;  END |
| CREATE PROCEDURE [dbo].[sp\_altaUser]  --datos direccion del cliente  @Nombre varchar (50),  @apellido varchar (50),  @idRol int,  @fecha date,  @pass int    AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  set @fecha = GETDATE()  insert into Usuarios (Nombre,apellido,idRol,status  ,password,FechaRegistro)  values(@Nombre,@apellido  ,@idRol,1,@pass,@fecha);  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_c\_Tabla\_Mesas]  AS  BEGIN  SET NOCOUNT ON;  Begin Tran transaccion  Begin Try  SELECT mesas.id, numeroMesa,statusMesa.Descripcion from dbo.mesas inner join statusMesa  on mesas.idStatusMesa=statusMesa.id;      COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| create procedure [dbo].[sp\_C\_Tabla\_Personal]  AS  BEGIN  SET NOCOUNT ON;  Begin Tran transaccion  Begin Try  SELECT id,Nombre from dbo.Usuarios  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | CREATE procedure [dbo].[sp\_C\_Table\_AllIngrediente]  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  SELECT id,nombre,stock from Ingredientes where status=1  COMMIT TRAN transacción  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| create procedure [dbo].[sp\_C\_Table\_AllProducts]  @idCategoria int  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  SELECT nombre,Descripcion,Precio,categoría  from producto inner join catProducto  on producto.idCategoriaProducto=catProducto.id where catProducto.id=@idCategoria  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_C\_Table\_CatCliente]  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  SELECT id, Categoria from dbo.catCliente;      COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| create procedure [dbo].[sp\_C\_Table\_CatProducto]  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  SELECT id, categoria from dbo.catProducto  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_C\_Table\_Cliente]  AS  BEGIN  SET NOCOUNT ON;  Begin Tran transaccion  Begin Try  SELECT id,Nombre from dbo.clienteFrecuente  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| create procedure [dbo].[sp\_C\_Table\_MesasLibres]  AS  BEGIN  SET NOCOUNT ON;  Begin Tran transaccion  Begin Try  SELECT mesas.id, numeroMesa from dbo.mesas where idStatusMesa=1;  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_C\_Table\_MesasOcupadas]  AS  BEGIN  SET NOCOUNT ON;  Begin Tran transaccion  Begin Try  SELECT mesas.id, numeroMesa from dbo.mesas where idStatusMesa=3 or idStatusMesa=2 ;  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| create procedure [dbo].[sp\_C\_Table\_ProductosCombo]  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  SELECT id,nombre from producto  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_C\_Table\_Rol]  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  SELECT id, rol from dbo.rol  COMMIT TRAN transacción  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| create PROCEDURE [dbo].[sp\_E\_Table\_Cliente]  @nombrecliente varchar (50)  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  --delete from producto where nombre=;  update clienteFrecuente set status=0 where nombre=@nombrecliente;  END | CREATE procedure [dbo].[sp\_E\_Table\_Ingrediente]  @nombre varchar (59)  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  update dbo.Ingredientes set status=0 where nombre=@nombre  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| create PROCEDURE [dbo].[sp\_E\_Table\_Producto]  --datos direccion del cliente  @idProducto int  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  update producto set status=0 where id=@idProducto  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create PROCEDURE [dbo].[sp\_E\_Table\_Usuario]  @nombreUsario varchar (50)  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  --delete from producto where nombre=;  update Usuarios set status=0 where nombre=@nombreUsario;    END |
| create procedure [dbo].[sp\_insertarRelacionProductosEingredinentes]  @idProducto int,  @idIngrediente int  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  insert into detalleProductoIngrediente(idProducto  ,idIngrediente)  values (@idProducto,@idIngrediente)  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_M\_table\_Ingrediente]  @nombre varchar (59),  @stock int  AS  BEGIN    UPDATE dbo.Ingredientes  SET  stock = @stock      WHERE nombre = @nombre  END |
| CREATE PROCEDURE [dbo].[sp\_M\_Table\_liberarMesa]  -- Se agregan los parameros  @idMesa int  AS  BEGIN  -- SET NOCOUNT ON addedos to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  update mesas set idStatusMesa=1 where id = @idMesa;  --Rollback TRAN transaccion  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch    END | create PROCEDURE [dbo].[sp\_OrdenLLevar]  --datos direccion del cliente  @idcliente int,  @observacion varchar (50),  @idPersonal int  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  declare @TraerIdOrden int  set @TraerIdOrden=(select MAX(id) as id from orden )  insert into ordenLlevar(idOrden,idResponsable,  observacion,idCliente) values(@TraerIdOrden,@idPersonal,  @observacion,@idcliente);  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| create PROCEDURE [dbo].[sp\_OrdenMesaLocal]  --datos direccion del cliente  @idMesa int,  @idcliente int,  @idPersonal int  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT  statements.  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  declare @TraerIdOrden int  set @TraerIdOrden=(select MAX(id) as id from orden )  insert into ordenMesa(idOrden,idMesa,idCliente,  idPersonal) values(@TraerIdOrden,@idMesa,@idcliente  ,@idPersonal);  update mesas set idStatusMesa=2 where id = @idMesa;  COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END | create procedure [dbo].[sp\_selectOrden]  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  SELECT id, monto , Fechaa from dbo.orden      COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |
| CREATE procedure [dbo].[sp\_sesion]  @user varchar (50),  @pass varchar (50)  AS  BEGIN  SET NOCOUNT ON;  --Inicia la transaccion  Begin Tran transaccion  Begin Try  SELECT Nombre,password from Usuarios  where Nombre=@user and password=@pass and status=1;    COMMIT TRAN transaccion  End Try  Begin Catch  Rollback TRAN transaccion  End Catch  END |  |

**CODIGO M-V-C DEL CASO DE USO SELECCIONADO**

**PROCEDIMIENTO ALMACENADO DEL CASO DE USO SELECCIONADO**

* 1. **EVIDENCIAS DE SOFTWARE**

****

****

# CAPITULO IV

# CONCLUSIONES Y RECOMENDACIONES

## CONCLUSIONES

Luego de terminar con el desarrollo del proyecto se concluye con lo siguiente:

* + - La implementación del primer entregable sobrelleva con facilidad las necesidades más básicas registradas en los requerimientos funcionales de la empresa, para registro de clientes, registro de administradores, permitiendo a la vez guardar estos registros en una base de datos que se asocian a registro de ventas y usuarios como de administradores que generan una o varias transacciones.
    - La utilización de sistemas informáticos dentro de restaurantes, hoy en idea es indispensable, debido a la gran demanda de los comensales que a la vez manejan cierta información que nos ayudarían ser mejores que la competencia, teniendo una ventaja sobre ellos.
    - Gracias a las herramientas utilizadas para el desarrollo del sistema y las tecnologías de alto nivel, el sistema desarrollado para esta disertación cuenta con los estándares de seguridad necesarios para un sistema de gestión de pedidos, asimismo la facilidad y rapidez de uso.
    - Teniendo en cuenta el entorno en el vivimos y la competencia debemos considerar algunas opciones tecnológicas para darle al restaurante una solución a los problemas que tiene.
    - Toda aplicación primero debe pasar por un proceso de análisis de requerimientos, estudios de factibilidad, implementación de prototipos, determinar los factores críticos y de éxito, diseño, plan de prueba, que el usuario o el dueño no ven, lo cual nos interesa que cumpla las necesidades por las cuales fue concebido.
    - En la actualidad se conocen demasiadas herramientas que nos ayudan a tener ventaja, en cuanto a la facilidad de diseño, transportación, almacenamiento y diversas plataformas.
    - Durante el proceso de construcción del sistema se realizaron varias pruebas para garantizar la calidad del funcionamiento de los mismos respecto a los criterios der seguridad, rendimiento, interfaz gráfica, almacenamiento de datos, basándose es un plan de pruebas enfocado por iteraciones.
    - El programa realiza cotizaciones y pedidos a proveedores en línea es un proyecto que representa un considerable aporte para la Empresa dado que ésta podría contar con una herramienta segura y optima de Administración de la información.
    - Se concluye en que la mejor manera de evitar conflictos de entrega es optimizando los procesos de producción de los servicios y productos, por ello el sistema al planificar un cronograma accesible, facilita la gestión de entrega generando un valor agregado a la empresa.
    - La administración de los procesos solo depende del dueño y de algunos administradores, el software apoyara con las pautas necesarias para tomar decisiones, pero no afectara con la propia autonomía.

## RECOMENDACIONES

* El proyecto puede ser ampliado logrando sus alcances sea aún mayor y siendo este orientado en un producto que esta implementado para almacenar la información en la computadora, es necesario que los sistemas se adecuen a mejores niveles de seguridad que aseguran que la información no sea eliminada, o corrompida.
  + - Los restaurantes que quieran expandirse deben o tienen que implementar soluciones de este tipo, para que la carga del trabajo disminuya y sea mayor su rendimiento
    - Se recomienda desarrollar las aplicaciones de software para pequeñas o medianas empresas bajo herramientas con licencias, ya que muy difícil es adquirir herramientas de software propietarias a un costo de licenciamiento.
    - El sistema debe ser administrado con responsabilidad ya que la información que va a contener de suma importancia. A pesar de que un sistema cuenta con altos estándares de seguridad, las personas que tengan acceso al programa como administradores deberá ser muy prudentes al respecto.
    - Se sugiere la inclusión o implementación de los módulos de facturación, adquisición e inventarios pues este les facilitaría el trabajo a ellos y les ahorraría tiempo a sus clientes en toma de decisiones respecto al pedido a realizar.
    - Al desarrollar el sistema se debe tener en cuenta las mejoras herramientas, como también cuales son las más fáciles al implementar en el sistema, ya que al terminar el software debe de ser fácil de utilizar.

## BIBLIOGRAFIA

Management, E. E. (29 de Junio de 2017). Escuela Europea de Management. Obtenido de <http://www.escuelamanagement.eu/direccion-general-2/4143>

Manene, L. M. (28 de Julio de 2011). Luis Miguel Manene. Obtenido de [http://www.luismiguelmanene.com/2011/07/28/los-diagramas-de-flujo-su-definicion-objetivo-](http://www.luismiguelmanene.com/2011/07/28/los-diagramas-de-flujo-su-definicion-objetivo-ventajas-elaboracion-fases-reglas-y-ejemplos-de-aplicaciones/) [ventajas-elaboracion-fases-reglas-y-ejemplos-de-aplicaciones/](http://www.luismiguelmanene.com/2011/07/28/los-diagramas-de-flujo-su-definicion-objetivo-ventajas-elaboracion-fases-reglas-y-ejemplos-de-aplicaciones/)

Microsoft. (13 de Marzo de 2017). Obtenido de [https://docs.microsoft.com/es-es/sql/relational-](https://docs.microsoft.com/es-es/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-2017) [databases/stored-procedures/stored-procedures-database-engine?view=sql-server-2017](https://docs.microsoft.com/es-es/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-2017)

Miró, A. (17 de Agosto de 2017). DEUSTO FROMACIÓN. Obtenido de [https://www.deustoformacion.com/blog/programacion-diseno-web/que-es-para-que-sirve-](https://www.deustoformacion.com/blog/programacion-diseno-web/que-es-para-que-sirve-github) [github](https://www.deustoformacion.com/blog/programacion-diseno-web/que-es-para-que-sirve-github)

1. Pressman. Ingeniería del Software: Un Enfoque Práctico. 7 Ed. McGraw Hill, 2010.
2. Sanchez, M. Sicilia, D. Rodriguez. Ingenieria del software. Un enfoque desde la guía SWEBOK. Alfaomega. 2012.

Sommerville Ian. Ingeniería del Software. Pearson 7a. Edición. 2005.

TutorialesPoint. (2016). TUTORIALESPOINT. Obtenido de SIMPLYEASYLEARING: [https://www.tutorialspoint.com/es/software\_engineering/software\_development\_life\_cycle.ht](https://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm) [m](https://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm)

Wikipedia. (14 de enero de 2019). Obtenido de <https://es.wikipedia.org/wiki/Procedimiento_almacenado>

## INDICE DE ABREVIATURAS DE SIGLAS

**SQL (Structured Query Language):** es un lenguaje estándar e interactivo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

##### **Transact-SQL (TSQL):** Es una extensión al SQL de Microsoft y Sybase. SQL. Es un lenguaje de cómputo estandarizado, desarrollado originalmente por IBM para realizar búsquedas, alterar y definir bases de datos relacionales utilizando sentencias declarativas.

**XML:** Es un subconjunto de SGML(Estándar Generalised Mark-up Language),simplificado y adaptado a Internet

## GLOSARIO DE TERMINOS

**Automatización**: El conjunto de métodos que sirven para realizar tareas repetitivas en un ordenador. Algunos métodos para la automatización de tareas son la programación simple, los macros, los intérpretes y las bombas lógicas. También hay algunos programas específicos que automatizan tareas. Incluso los virus informáticos utilizados de forma benéfica podrían considerarse otro método para la automatización de tareas para el usuario. Existen dos tipos de archivo, los ejecutables y no ejecutables o archivos de datos.

**Extensión**: Las extensiones nos ayudan a diferenciar los tipos de archivo, y generalmente están compuestas por 3 letras, considerando algunas excepciones (.jpeg; .java; .html; etc). Interfaz: Es la representación gráfica de una aplicación interactiva, es lo que media y facilita la comunicación e interacción entre el individuo y la máquina, ella es lo que el usuario ve en la pantalla y con la que interactúa finalmente.

**Plataforma**: Es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software (incluyendo entornos de aplicaciones).

**Servidor**: Es el computador en el que se ejecutan los programas que realizan alguna tarea en beneficio de otras computadoras. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un sitio web, así como ejecutar sus aplicaciones asociadas, en beneficio directo del usuario final.

Sistemas Operativos: Constituyen un conjunto de programas básicos y utilidades, compilados y distribuidos en formato sencillo para su fácil instalación, y de los cuales depende el funcionamiento de la computadora.

**Wireframes**: Un wireframe para un sitio web, también conocido como un esquema de página o plano de pantalla, es una guía visual que representa el esqueleto o estructura visual de un sitio web. El wireframe esquematiza el diseño de página u ordenamiento del contenido del sitio web, incluyendo elementos de la interfaz y sistemas de navegación, y cómo funcionan en conjunto.

## APENDICE

**4.6.1. CARTA DE ACEPTACION DEL RESTAURANTE DOÑA RUFFINA**

