

UNIVERSIDAD CATOLICA DE SANTA MARIA

FACULTAD DE CIENCIAS E INGENIERIAS FISICAS Y FORMALES

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS



“MEJORA DE EFICIENCIA DE PROCESO DE VENTAS EN RESTAURANTES UTILIZANDO SISTEMAS EXPERTOS”

Tesis presentada por el Bachiller:

BARROS DEL CASTILLO, EDUARDO GUILLERMO

para optar el Título Profesional de:

INGENIERO DE SISTEMAS

Asesor: Ing. Guillermo Calderón Ruiz

**AREQUIPA – PERÚ
2017**

PRESENTACION

El presente trabajo, tiene por objetivo, el desarrollo de sistema experto para la mejorar la eficiencia de los procesos de ventas en el rubro de restaurantes. Para el cual se hace uso de metodologías para el desarrollo; así como también, de la creación métodos para la explotación de datos para que aporten un valor significativo en la toma de decisiones del área de análisis y procesamiento de información.



DEDICATORIA

El presente trabajo de tesis primeramente me gustaría agradecer a Dios por bendecirme para llegar hasta donde he llegado.

Le doy gracias a mi familia por acompañarme en todos los momentos de mi vida, en especial a mis padres Guadalupe y Guillermo por su amor incondicional, apoyándome en todo momento y por darme la oportunidad de tener una excelente educación en el transcurso de mi vida.

A mi hermano Sebastián, por ser mi compañero y la persona que me ha brindado grandes lotes de felicidad y diversas emociones.

A mi abuela Rosario, por ser parte importante en mi vida, en sus enseñanzas.

A mi pareja Allison, por ser mi fiel compañera, y la persona que me ha brindado su apoyo y comprensión en mi vida.

Le agradezco a mi asesor, Ing. Guillermo Calderón Ruiz, por la confianza, apoyo, dedicación de su tiempo y principalmente por brindarme su amistad.

INDICE DE CONTENIDO

PRESENTACION	2
DEDICATORIA.....	3
ABSTRACT.....	12
INTRODUCCIÓN.....	13
CAPÍTULO I: PLANTENAMIENTO TEÓRICO.....	14
1.1. Problema	14
1.2. Descripción del problema	14
1.3. Justificación	14
1.4. Tipo y nivel de investigación	15
1.5. Variables.....	15
1.6. Hipótesis.....	15
1.7. Objetivos.....	16
1.8. Alcances	17
1.9. Delimitaciones	17
CAPITULO II: MARCO TEÓRICO	18
2.1. Fundamentos Teóricos.....	18
2.2. Estado del arte (Antecedentes del proyecto)	18
2.3. Bases Teóricas del proyecto.....	21
2.4. Técnicas y Herramientas	25
2.5. Herramientas de Desarrollo	26
2.6. Aspectos relevantes del desarrollo	29
CAPÍTULO III: DOCUMENTACIÓN TÉCNICA	31
3.1. Documentación Técnica del sistema experto.....	32
3.2. Plan del proyecto Informático.....	32
3.3. Arquitectura del sistema	77
3.4. Diseño de las bases de datos	79

CAPÍTULO IV: DISEÑO DEL SISTEMA EXPERTO	84
4.1. Documentación técnica de programación	84
4.2. Mapa del Sistema	85
4.3. Arquitectura de programación.....	86
4.4. Aspectos relevantes de la programación del sistema	86
CAPITULO V: VALIDACIÓN DE LOS MÓDULOS DEL SISTEMA EXPERTO	92
5.1. Módulo de Usuarios	92
5.2. Módulo de roles	94
5.3. Módulo de asignar actividades.....	95
5.4. Módulo de almacén.....	97
5.5. Módulo de alimentos.....	98
5.6. Módulo de medidas.....	100
5.7. Módulo de categorías de ingredientes	101
5.8. Módulo de platos	102
5.9. Módulo de Menú	104
5.10. Módulo de categorías de platos	106
5.11. Módulo de preparación de platos.....	107
5.12. Módulo de asignación de platos a un menú.....	109
5.13. Módulo de promociones.....	110
5.14. Módulo de asignación de platos a promociones	112
5.15. Módulo de mantenimiento de pedidos	113
5.16. Módulo de mantenimiento de mesas	115
5.17. Módulo de mesas disponibles.....	116
5.18. Módulo de factura	117
5.19. Módulo de mantenimiento de clientes.....	119
5.20. Módulo de reporte de almacén del día siguiente.....	120
5.21. Módulo de reporte de ventas diarias	121
5.22. Módulo de reporte de almacén actual	122
5.23. Módulo de reporte de tiempo de espera por día	124
5.24. Módulo de reporte de histórico de ventas por producto	125
5.25. Módulo de reporte de hora de demanda	126
5.26. Módulo de reporte de histórico de ventas por meses.....	127
5.27. Módulo de reporte de histórico del promedio de ventas por día de semana.....	128
5.28. Módulo de promociones sugeridas	129

5.29. Módulo de relación de productos	130
5.30. Módulo de hechos automáticos	131
5.31. Población del data warehouse y cierre del sistema	131
5.32. Sistema de reportería	131
5.33. Árbol de decisión del sistema experto.....	132
5.34. Programación del Job	134
5.35. Instalación de la aplicación	135
5.36. Vista del Sistema en diferentes dispositivos	137
5.37. Pruebas del sistema	140
5.38. Manuales de usuario.....	140
CAPÍTULO VI : RESULTADOS	141
CONCLUSIONES.....	146
RECOMENDACIONES	147
REFERENCIAS BIBLIOGRÁFICAS	148
APÉNDICES.....	150
Anexo A.....	150
Anexo B.....	177
Anexo C.....	200
Anexo D.....	232
Anexo E.....	269

INDICE DE FIGURAS

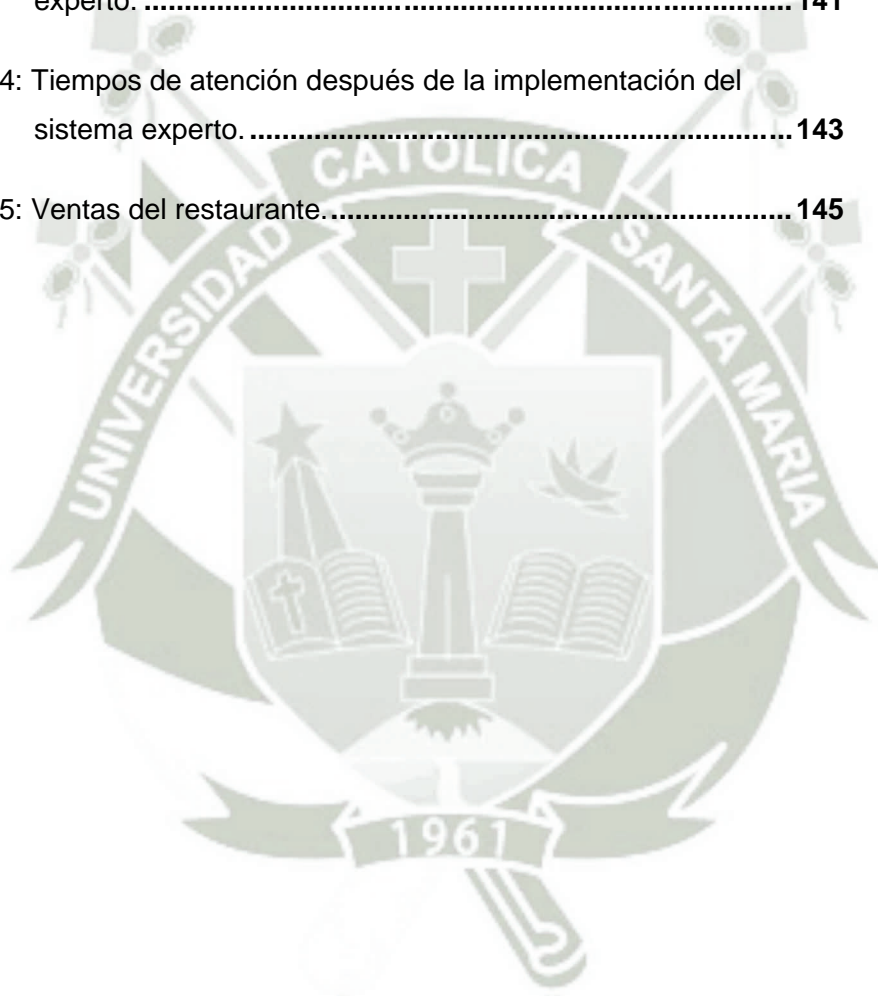
Figura 1: Estructura del sistema experto.	23
Figura 2: Entidad y relación de la base de datos transaccional.	46
Figura 3: UML de registro almacén.	47
Figura 4: UML del administrador del sistema.	49
Figura 5: UML del encargado de ventas.	50
Figura 6: UML del cajero.	50
Figura 7: UML del encargado de cocina.	51
Figura 8: UML del mesero.	51
Figura 9: Diagrama de secuencia de mantenimiento de almacén.	55
Figura 10: Diagrama de secuencia del mantenimiento de insumos.	56
Figura 11: Diagrama de secuencia del mantenimiento de medidas.	57
Figura 12: Diagrama de secuencia del mantenimiento de categoría de ingrediente.	58
Figura 13: Diagrama de secuencia del mantenimiento de usuarios.	59
Figura 14: Diagrama de secuencia del mantenimiento de roles.	60
Figura 15: Diagrama de secuencia del mantenimiento de platos.	61
Figura 16: Diagrama de secuencia del mantenimiento de promoción.	62
Figura 17: Diagrama de secuencia del registro de platos.	63
Figura 18: Diagrama de secuencia del cierre del sistema.	64
Figura 19: Diagrama de secuencia de la visualización de reportes.	65
Figura 20: Diagrama de secuencia de sugerencia del sistema experto.	66
Figura 21: Diagrama de secuencia de mesas disponibles.	67
Figura 22: Diagrama de secuencia del mantenimiento de caja.	68
Figura 23: Diagrama de secuencia del mantenimiento de clientes.	69
Figura 24: Diagrama de secuencia del mantenimiento de pedidos.	70
Figura 25: Diagrama de clases.	71
Figura 26: Proceso de ventas.	76
Figura 27: Arquitectura del sistema.	77
Figura 28: Modelo relacional de la base de datos transaccional.	80
Figura 29: Modelo relacional del data warehouse de la tabla de hechos ingredientes.	81

Figura 30: Modelo relacional del data warehouse de la tabla de hechos de pagos.	82
Figura 31: Modelo relacional de la base de datos de conocimiento.	83
Figura 32: Mapa del sistema.	85
Figura 33: Modelo de capas del sistema.	87
Figura 34: Conexión a la base de datos.	88
Figura 35: Conexiones a la base de datos.	89
Figura 36: Grilla del sistema.	90
Figura 37: Llamado a un procedimiento almacenado del sistema.	90
Figura 38: Ejemplo de procedimiento almacenado en la base de datos del sistema.	91
Figura 40: Figura del módulo de mantenimiento de roles.	95
Figura 41: Figura del módulo de asignar actividades a un rol.	96
Figura 42: Figura del módulo de Mantenimiento de almacén.	98
Figura 43: Figura del módulo de Mantenimiento de alimentos.	99
Figura 44: Figura del módulo de Mantenimiento de medidas.	101
Figura 45: Figura del módulo de Mantenimiento de categoría de ingrediente.	102
Figura 46: Figura del módulo de Mantenimiento de platos.	104
Figura 47: Figura del módulo de Mantenimiento de menús.	105
Figura 48: Figura del módulo de Mantenimiento de categoría de platos.	107
Figura 49: Figura del módulo de asignar ingredientes a un plato.	109
Figura 50: Figura del módulo de asignar platos a un menú.	110
Figura 51: Figura del módulo de creación de promociones.	112
Figura 52: Figura del módulo de asignar una promoción a un plato.	113
Figura 53: Figura del módulo de mantenimiento de pedidos.	115
Figura 54: Figura del módulo de mantenimiento de mesas.	116
Figura 55: Figura del módulo del reporte de mesas disponibles.	117
Figura 56: Figura del módulo de facturación de pedidos.	118
Figura 57: Figura del módulo de mantenimiento de clientes.	120
Figura 58: Figura del reporte del almacén promedio del día siguiente.	121
Figura 59: Figura del reporte de las ventas del día.	122
Figura 60: Figura del reporte del almacén actual.	123
Figura 61: Figura del reporte del tiempo de espera por día de semana.	124
Figura 62: Figura del reporte del histórico de ventas por producto.	125
Figura 63: Figura del reporte del histórico de hora de demanda.	126

Figura 64: Figura del reporte del histórico de monto de ventas por meses.	127
Figura 65: Figura del reporte del promedio de ventas por día de semana.	128
Figura 66: Figura del reporte de promociones sugeridas.	129
Figura 67: Figura del reporte de relación de productos.	130
Figura 68: Figura del reporte de hechos automáticos.	131
Figura 69: Figura del sistema de reportería.	132
Figura 70: Figura de la funcionabilidad del árbol del sistema experto.	133
Figura 71: Figura de la programación de la ejecución del árbol de decisiones del sistema experto.	134
Figura 72: Figura donde se encuentra el IIS del sistema.	135
Figura 73: Figura donde se encuentra los archivos del sistema.	136
Figura 74: Figura de la vista del sistema desde un ordenador.	137
Figura 75: Figura de la vista del sistema desde una tableta.	138
Figura 76: Figura de la vista del sistema desde un Smartphone.	139
Figura 77: Figura de la atención antes de la implementación del sistema experto.	142
Figura 78: Figura de la atención antes de la implementación del sistema experto.	143
Figura 79: Figura de la diferencia de tiempos de atención de clientes.	144
Figura 80: Figura de la evolución de las ventas en el restaurante.	145

INDICE DE TABLAS

Tabla 1: Actividades para el desarrollo de la tesis.....	32
Tabla 2: Análisis de costo – beneficio.....	43
Tabla 3: Tiempos de atención antes de la implementación del sistema experto.	141
Tabla 4: Tiempos de atención después de la implementación del sistema experto.....	143
Tabla 5: Ventas del restaurante.....	145



RESUMEN

El presente proyecto tiene como finalidad el desarrollo de un sistema experto para mejorar la eficiencia en el proceso de ventas en el ámbito de los restaurantes peruanos.

Para mejorar la eficiencia en el proceso, se desarrolló un sistema transaccional; en el cuál se guardarán todas las transacciones del proceso de ventas; además de la creación un data warehouse del que se compondrá de dos datamarts; uno de ellos, un datamart de pedidos y el otro datamart del almacén de los insumos.

Por medio del data warehouse se analizará la evolución del negocio con el análisis de los datos históricos, además permitirá realizar predicciones de la relación de ventas de los productos; esta base de datos será la fuente de ingreso para poblar la base de conocimiento.

La base de conocimiento realizará un análisis en línea, respondiendo las reglas predefinidas por los expertos, para tomar decisiones en línea sobre el negocio.

Los tipos de decisiones que tomará el sistema experto; pueden ser de crear una promoción en un periodo de tiempo, regalar un producto, promocionar un tipo específico de producto o el cierre del sistema automático.

Palabras Clave: Sistema experto, Inteligencia de negocios, gestión del conocimiento.

ABSTRACT

The present project aims to develop an expert system to improve efficiency in the sales process in the field of peruvian restaurants.

To improve efficiency in the process, a transactional system was developed; in which all the transactions of the sales process will be stored; in addition to creating a data warehouse that will be made up of two datamarts; one of them a datamart of orders and the other datamart of the warehouse of the inputs.

Through the data warehouse will analyze the evolution of the business with the analysis of historical data, in addition will allow to make predictions of the sales relationship of the products; this database will be the source of income to populate the knowledge base.

The knowledge base will perform an online analysis, responding to rules predefined by experts, to make online decisions about the business. The types of decisions that the expert system will make; can be to create a promotion in a period of time, give a product, promote a specific type of product or the closing of the automatic system.

Keywords: Expert system, Business intelligence, knowledge management.

INTRODUCCIÓN

El objetivo de este Proyecto es mejorar la eficiencia del proceso de ventas, por ello se creó un sistema transaccional, un sistema de análisis de datos y un sistema experto, de acuerdo a los requerimientos de las personas en el proceso de ventas. En este trabajo detalla el análisis, diseño y desarrollo del sistema.

En el primer capítulo se realiza la descripción del problema, la descripción del problema, la justificación, el tipo y nivel de investigación, las variables, la hipótesis, los objetivos, el alcance y las delimitaciones.

En el segundo capítulo se realizará el análisis de marco teórico, en el que contendrá los apartados de fundamentos teóricos, estado del arte, las bases teóricas del proyecto, las técnicas y herramientas, las herramientas de desarrollo y los aspectos relevantes del desarrollo.

En el tercer capítulo se analizará la documentación técnica del sistema experto.

En el cuarto capítulo se realizará el diseño del sistema experto.

En el quinto capítulo se realizará la validación de los módulos del sistema experto.

En el sexto capítulo se realizará el análisis de los resultados del trabajo de tesis.

CAPITULO I

PLANTENAMIENTO TEÓRICO

1.1. Problema

Mejora de eficiencia de proceso de ventas en restaurantes utilizando sistemas expertos.

1.2. Descripción del problema

En el proceso de ventas de los restaurantes presentan una mala eficiencia, generando el descontento de la clientela y la pérdida de la fidelización de estos, evitando el crecimiento de los restaurantes.

El presente proyecto tiene como finalidad la creación de un sistema experto para mejorar la eficiencia del proceso de ventas; ya que tomará decisiones propias de acuerdo a las reglas predefinida por los expertos involucrados en el proceso de ventas.

El sistema experto se desarrollará en un ambiente web, el cual permitirá el acceso al sistema experto desde diferentes dispositivos; en el sistema se permitirá el registro de todas las transacciones del restaurante; estos será la fuente para el poblamiento del data warehouse; mediante el cual se obtendrá los reportes del sistema; a su vez servirá de fuente para el poblamiento de la base de conocimiento del sistema experto.

1.3. Justificación

Se desarrolló este proyecto, para aumentar la eficiencia en el procesos de ventas en los restaurantes peruanos; este aumento de la eficiencia se dará por medio de la reducción de los costos de insumos, la promoción de los productos menos vendidos, la seguridad de los datos

de la empresa y las sugerencias que dará el sistema a la empresa, para que pueda servir de apoyo en la toma de decisiones.

Surgió la idea de realizar este proyecto, ya que se identificó que el negocio de los restaurantes está en constante crecimiento en el mercado peruano; sin embargo presenta déficit en la eficiencia de sus procesos de ventas; motivo por el cual pierde la confianza de sus clientes; generando así que no pueda consolidarse en el mercado.

1.4. Tipo y nivel de investigación

El tipo de la investigación es aplicada.

El nivel de la investigación es descriptiva.

1.5. Variables

En la presente tesis se distinguen las siguientes variables.

Sistema experto. (Variable independiente)

Indicadores

T: Tiempo de atención a los clientes.

Eficiencia del proceso de ventas. (Variable dependiente)

Indicadores

E: Eficiencia del proceso de ventas.

V: Número de ventas del restaurante.

G: Ganancia del restaurante.

1.6. Hipótesis

En la actualidad el proceso de ventas de los restaurantes, presentan una baja eficiencia, por lo que no satisface las necesidades de su clientela; por

ello es probable, que al mejorar el proceso del ventas por medio de un sistema experto; se pueda tomar decisiones sin la necesidad de una persona experta involucrada en el proceso, de acuerdo a las necesidades del negocio de la forma más rápida posible.

El sistema experto deberá satisfacer las necesidades del restaurante, en el proceso de ventas; para que satisfazca las necesidades de las personas involucradas en el proceso, y aumente la calidad de atención a la clientela.

1.7. Objetivos

1.7.1. Objetivo General:

Desarrollar un sistema experto que permita aumentar la eficiencia de los restaurantes.

1.7.2. Objetivos Específicos:

Los objetivos específicos son los siguientes:

1. Analizar los puntos críticos de ventas para poder identificar los puntos críticos del mismo.
2. Determinar y especificar los requerimientos en conjunto con las personas involucradas en el proceso de ventas de los módulos que presentará el sistema experto.
3. Analizar y desarrollar un data warehouse de los datos transaccionales de la empresa para poder analizar la evolución de las ventas del restaurante de una forma más óptima.
4. Realizar el desarrollo de un sistema experto que permite realizar la funcionabilidad del proceso de ventas.

5. Analizar con las personas involucradas en el proceso de ventas para el desarrollo del sistema experto, de acuerdo a las especificaciones de las reglas que presente.

1.8. Alcances

El proyecto desarrollado tiene como alcance la implementación de un sistema experto que permita el control de las transacciones de un restaurante y que tome decisiones propias sobre el negocio.

En los restaurantes peruanos existe la necesidad de poder generar diversos reportes para que puedan tener el control y facilitar accesos a sus datos; para poder mejorar la eficiencia de sus procesos.

Se implementó un módulo experto que permite que el sistema tome decisiones por sí solas; generando la creación de promociones de acuerdo a los eventos que se presenten.

1.9. Delimitaciones

El tema de investigación se delimitará, a que el sistema experto almacenará los datos de almacén de sólo los ingredientes físicos del mismo.

CAPITULO II

MARCO TEÓRICO

2.1. Fundamentos Teóricos.

En este apartado se analizará la recopilación de los artículos encontrados sobre el tema de sistemas expertos aplicados en el proceso de ventas, para poder analizar el estado actual del tema en dicho segmento.

2.2. Estado del arte (Antecedentes del proyecto)

En Bogotá, Colombia en el año 2015 se realizó un proyecto de inteligencia de negocios para un restaurante, teniendo como objetivo del proyecto brindar información de las ventas del negocio, para ello se usó un sistema de inteligencia de negocios; este proyecto ayudó a que la empresa pudiera aumentar las ventas y tengan una mayor aceptación de sus clientes, con la implementación del sistema permitirá tomar mejores decisiones, generando ventaja competitiva en el mercado. (Giraldo, 2015)

En Talca, Chile, se realizó un proyecto de software y un rediseño del proceso de ventas en un restaurante de su localidad, la problemática se generó debido a la insatisfacción de la clientela; ya que demoraban en la atención de los clientes, para ello se rediseñó el procesos de ventas, generaron que el tiempo de espera de los clientes se reduzca en un 20%, teniendo como efecto el aumento de las ventas y mejorar la rentabilidad del negocio. (Schmal, 2014)

En Guayaquil, Ecuador se realizó un sistema experto para el mantenimiento de almacenes de una empresa de venta por

departamentos, el cual consistía en la creación de reglas para tener los almacenes en constante rotación y evitar el estancamiento de sus insumos; el proyecto tuvo como resultado la creación de promociones para que los insumos en almacén no se queden sin rotación. (Crespo, 2013)

En Quito, Ecuador se realizó una mejora en el proceso de inventarios de una licorería, en la cuál presentaban problemas en su almacén, por lo cual se desarrolló una plantilla en Excel para poder comprar los insumos necesarios en la empresa, el trabajo tuvo como resultado tener un seguimiento de la entrada y salida de los insumos, generando que se pueda solucionar los problemas cotidianos de manera más rápida. (Gallegos, 2016)

En Castilla, España se realizó una aplicación móvil de reserva de mesas en los restaurantes, se desarrolló esta aplicación, ya que se presentaba un gran descontento de la clientela; se obtuvieron como resultados, la reducción de tiempo en espera de los clientes, generando la fidelización de la clientela y aumentando las ventas del restaurante. (Ramirez, 2016)

En San Salvador, El Salvador se generaba la problemática en los restaurantes de ventas de pollo rostizado, ya que no costeaban de manera correcta la elaboración de sus productos, por ello se desarrolló un sistema que evaluaba los costos de producción, teniendo información que es de soporte a la toma de decisiones y generando el correcto valor de venta de los productos. (Romero, 2016)

En Etiopía se realizó un documento técnico con el nombre “*self learning computer troubleshooting expert system*”; en el cual realizó un sistema experto basado en reglas, que servía para la reducción de tiempos en la resolución de los problemas basados en infraestructura de computación, por ello se realizó un árbol de decisión, en el cual tuvo una aceptación de 83% de aceptación por los expertos en el área, teniendo un aumento en la satisfacción de los clientes involucrados con la problemática. (Ayde, 2016)

En Estados Unidos (2015) se realizó un documento técnico con el nombre “*An ergonomic expert system for risk assessment of work-related musculo-skeletal disorders*”; en el cual permitía identificar los riesgos ergonómicos relacionados con el trabajo; el trabajo consiste en resolver ciertas preguntas para poder identificar el riesgo que se tendría de acuerdo a la evaluación de las respuestas.

En conclusión de los artículos revisados sobre los sistemas expertos; se identificó que están basados en reglas, y en la rama que más se utiliza es la medicina y la tecnología; sin embargo en el área de las ventas, la información es muy escasa; dado que para el análisis de los datos en los procesos de ventas se usan la inteligencia de negocios y la minería de datos; permitiendo que en la organizaciones mejoren sus procesos de ventas.

2.3. Bases Teóricas del proyecto

2.3.1. Sistema Experto

Según Castillo define al sistema experto como: *“Un sistema experto puede definirse como un sistema informático (hardware y software) que simula a los expertos humanos en un área de especialización dada.”* (Castillo, 2016)

Como tal, un sistema experto tiene la funcionabilidad de tomar decisiones propias de acuerdo a los eventos presentados en una determinada área, por medio de la recopilación de la información de los expertos de dicha área.

a) Tipos de sistemas expertos

Los sistemas expertos se pueden clasificar en dos tipos según la naturaleza del problema; estos pueden ser de tipo determinista y estocástico.

Los problemas de tipo deterministas son formulados de acuerdo a un conjunto de reglas, este tipo de sistema se denomina sistema experto basado en reglas; estos sistemas expertos obtienen sus conclusiones basándose en un conjunto de mecanismos de razonamiento lógico.

Los sistemas expertos estocásticos utilizan la probabilidad como medida de incertidumbre, estos sistemas incluyen las redes bayesianas para poder realizar una presentación gráfica entre las relaciones entre variables. Este tipo de sistema obtiene sus

conclusiones a través de la probabilidad de causa entre las decisiones a tomar.

Ventajas de un Sistema Experto

- Los conocimientos de un sistema experto pueden ser copiados, estructurados y almacenados fácilmente.
- Toman decisiones de forma interrumpidas.
- A diferencia de los humanos siempre están en pleno rendimiento.
- Trabajan con gran cantidad de información.

b) Estructura de un Sistema Experto

La estructura principal de un sistema experto es el siguiente:

- **Base de conocimiento:** Contiene el conocimiento del experto, expresado en una base de datos.
- **Base de hechos:** Contiene los hechos sobre un problema.
- **Motor de inferencia:** Trabaja con la información para poder dar soluciones sobre un problema.
- **Interfaz de usuario:** Interacción entre el sistema experto y el usuario que lo utilizará.

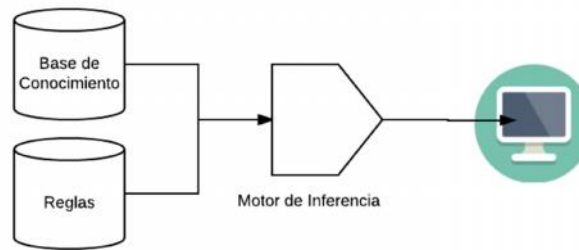


Figura 1: Estructura del sistema experto.

Fuente: Creación propia.

En la figura 1 se puede observar la estructura del sistema experto; en ella se identifican los componentes de la base de conocimiento, las reglas, el motor de inferencia y la interfaz a los usuarios.

2.3.2. Inteligencia de Negocios

“Es un término genérico que incluye las aplicaciones, la infraestructura, las herramientas, y las mejores prácticas que permitan el acceso y el análisis de la información para mejorar y optimizar las decisiones y el desempeño.”(Dresner, 2017)

a) Proceso de la inteligencia de negocios

La inteligencia de negocios presenta tres procesos principales, los cuales son:

- **Integración de datos**

En este proceso se busca la recopilación de los datos, que por lo general se encuentran en diferentes bases de datos de

distintos sistemas; con el propósito de integrarlos y tener una única fuente de información; buscando la homogenización de los datos, por el cual se utiliza el proceso de ETL, el cual consiste en la extracción, transformación y la carga.

- **Procesamiento de datos**

Tiene el propósito de generar información a través de la integración de los datos, para buscar una visión estratégica del negocio, con el fin de predecir el comportamiento del negocio.

- **Plataforma de distribución y entrega de datos**

Son herramientas que nos permite consultar los datos de la información generada; esta información se presenta a través de gráficos estadísticos o tablas. Estos reportes son entregados al usuario final por medio de correo electrónico o por una interfaz web, para que el usuario pueda accederla cuando lo necesite.

2.3.3. Gestión del conocimiento

“La gestión de conocimiento es información, reglas y experiencia que se deben usar en un contexto familiar, social, profesional, educativo, en una organización, cuando se toman decisiones.” (Bernuy, 2014)

La gestión del conocimiento surge debido a la problemática en la toma de decisiones en las empresas, para que la información sea útil en la empresa.

a) Ventajas

- Optimiza el flujo de la información en la empresa, evitando la duplicidad de las tareas.
- Genera ventaja competitiva.
- Obtiene el máximo conocimiento de los trabajadores de la empresa.
- Mejora la eficiencia.
- Mejora la rentabilidad.

2.4. Técnicas y Herramientas

Se usará la siguiente metodología:

- Identificar los puntos críticos del proceso.
- Diseño de las bases de datos.
- Diseño del sistema.
- Diseño de la seguridad de datos.
- Desarrollo del sistema.
- Pruebas del sistema.
- Implantación del sistema.

Existen 7 pasos a seguir dentro de la metodología.

- **Identificar los puntos críticos del proceso:** Análisis del proceso actual del proceso de ventas, para poder identificar los puntos críticos del proceso.
- **Diseño de las base de datos:** Diseño de los objetos de las bases de datos a usar, diseñando la arquitectura y aplicando la normalización de las bases de datos.

- **Diseño del sistema:** Diseño de los módulos del sistema.
- **Diseño de la seguridad de datos:** Diseño de los métodos de respaldo de la información.
- **Desarrollo del sistema:** Desarrollo de los módulos del sistema, según se hallan diseñado.
- **Pruebas del sistema:** Recolección de datos de dos meses en un ambiente real, para poder identificar el cambio que generó la aplicación del sistema.
- **Implantación del sistema:** Implantación del sistema en un ambiente real.

2.5. Herramientas de Desarrollo

Se analizarán las herramientas de desarrollo para poder elaborar el sistema experto.

A) C#

Es un lenguaje diseñado por Microsoft para su plataforma .NET, fué desarrollado por Scott Wiltamuth y Anders Hejlsberg.

Presenta una sintaxis muy parecida al lenguaje Java, con la diferencia que presenta mejoras a comparación de este lenguaje; además dicho lenguaje es la evolución de los lenguajes C y C++.

(Seco, 2001)

Las principales características que tiene el lenguaje son:

- **Sencillez en su uso:** Elimina muchos elementos de diferentes lenguajes que son innecesarios, por ello el código en C# es

autocontenido ya que no requiere de ficheros adicionales para la creación de interfaces.

- **Orientado a componentes:** Incluye elementos del diseño de componentes; que en otros lenguajes su construcción es compleja.
- **Gestión automática de memoria:** Presenta un recolector de basura propio de .NET, motivo por el cual no es necesario la destrucción de los objetos.
- **Seguridad de tipos:** Presenta mecanismos para asegurar el acceso de los datos, ya que cuenta con un control de conversiones de datos, inicialización de variables y control de desbordamientos.
- **Instrucciones seguras:** En las instrucciones de control presenta una serie de restricciones como la sentencia break.
- **Sistema de tipos unificado:** Todos los tipos de datos derivan de una clase base llamada System.Objects.
- **Eficiente:** Permite el uso de punteros de los objetos, generando una mayor velocidad de procesamiento.

B) SQL Server

Es un sistema para el manejo de datos relacionales, el lenguaje utilizado es T-SQL; presenta la herramienta SQL Server Management Studio (SSMS) que sirve para la administración del servidor y de sus bases de datos.

SSMS proporciona una interfaz basada en GUI para escribir y ejecutar consultas, las cuales son:

- **XQuery:** Lenguaje de consulta y programación para datos XML.
- **Sqlcmd Scripts:** Es una aplicación de líneas de comandos que expone las características de administración de SQL Server.
- **Transacs-SQL:** Es el medio principal de programación y administración de SQL Server ya que puede crear y administrar las bases de datos y componentes del mismo.

Las características de SQL Server son:

- Posee gran cantidad de herramientas administrativas como de desarrollo.
- Una de sus principales características es la seguridad, ya que para la protección de los datos utiliza criptografía integrada, cifrado transparente, además de soportar hashes. Maneja el control de accesos mediante el reparto de responsabilidades y roles.
- Utiliza métodos de restricción por medio de mecanismos como Primary Key, Foreign Key, Unique, Check, Default.
- Permite la creación de un diccionario de datos dentro de sus bases de datos.
- Permite la migración de los datos a la nube a través de Microsoft Azure.

C) Reporting Services

Es una solución de reportes para crear, publicar y administrar informes; tiene como principal característica, que puede ser visualizado en diferentes tipos de dispositivos; ya sean en el

explorador web, dispositivos móviles o por medio de correo electrónico.

Las principales características del reporting services son:

- **Paginación:** Reporting Services presenta la paginación en sus reportes ya que de acuerdo a la cantidad de filas, generará un correlativo de páginas; teniendo así un reporte mucho más entendible.
- **Informes móviles:** Los reportes podrán ser visualizados desde diferentes plataformas como son las de Windows, Android y IOS; además están implementados con HTML5 por lo que pueden ser leídos desde cualquier navegador que soporte HTML5.
- **Administración de los reportes:** Una de sus principales características es que presenta orígenes compartidos, lo cual permite compartir la conexión hacia las bases de datos entre distintos reportes; además permite asignar roles sobre los usuarios que tengan acceso a los reportes para que puedan editarlos o sólo visualizarlos.
- **Envío email:** Reporting services permite la ejecución y exportación de los reportes en distintos formatos, teniendo principalmente la exportación en PDF y Excel; para que puedan ser enviados a través de correo electrónico por medio de la ejecución de un job propio de la herramienta.

2.6. Aspectos relevantes del desarrollo

El sistema desarrollado cubre los requerimientos presentados por el usuario; que permite mejorar la eficiencia del proceso de ventas en el

restaurante. El acceso al sistema experto se podrá realizar desde diferentes dispositivos; es decir desde dispositivos de escritorio, como de dispositivos móviles.

El sistema permitirá la creación de reportes, para poder analizar los datos diarios, como datos históricos del negocio; a su vez generará un sistema de backups el cual permitirá el respaldo de los mismos.

Los reportes que presentará el sistema serán los siguientes:

- Reportes de ventas.
- Relación entre las ventas de productos.
- Reportes de almacén.
- Reportes históricos.
- Reportes por días de semana.

El sistema experto permitirá tomar decisiones de acuerdo a los eventos relacionados en el día, estos serán de tres tipos.

- Promociones de los productos.
- Regalo a los clientes.
- Cierre del sistema.

CAPÍTULO III

DOCUMENTACIÓN TÉCNICA

El sistema experto que se desarrollará, servirá para que tome decisiones propias de acuerdo a las circunstancias del día a día; ya que analizará los datos históricos y lo comparará en un árbol de decisión.

El sistema tendrá todos los módulos transaccionales del proceso de ventas; es decir permitirá el registro de los insumos como de los productos del restaurante; a su vez permitirá el registro de pedidos y la cancelación de los mismos.

El sistema contará con un poblamiento automático a un data warehouse; el cual permitirá el análisis de los datos históricos del restaurante; a su vez este servirá como fuente para la base del conocimiento del sistema experto; base de datos por la cual el sistema experto tomará decisiones.

Las decisiones que el sistema experto tomará, será la creación de promociones de productos específicos, regalar productos o la realización del cierre diario del sistema.

Por medio del sistema experto se mejorará la eficiencia del proceso de ventas; además del incremento de ventas y de la fidelización de la clientela.

3.1. Documentación Técnica del sistema experto

En el presente capítulo se presentará la documentación y requerimientos a nivel técnico del sistema experto.

3.2. Plan del proyecto Informático

Se definió el plan de trabajo de la siguiente manera.

Tabla 1: Actividades para el desarrollo de la tesis.

Fuente: Creación propia.

	Actividad	Días	Fecha Inicio	Fecha Fin	Predecesora
1	Análisis del proceso de ventas	7	15/08/16	23/08/16	
2	Identificar los puntos críticos	5	24/08/16	30/08/16	1
3	Análisis de los requerimientos	5	31/08/16	6/09/16	2
4	Determinar los módulos del sistema	4	7/09/16	12/09/16	3
5	Diseño de la BD transaccional	7	13/09/16	21/09/16	4
6	Creación de los objetos de la BD	5	22/09/16	28/09/16	5
7	Desarrollo mantenimiento del sistema	3	29/09/16	3/10/16	6
8	Desarrollo mantenimiento de insumos	3	4/10/16	6/10/16	7
9	Desarrollo mantenimiento de productos	3	7/10/16	11/10/16	8
10	Desarrollo mantenimiento de preparación	3	12/10/16	14/10/16	9
11	Desarrollo de promociones	2	17/10/16	18/10/16	10
12	Desarrollo de pedidos	3	19/10/16	21/10/16	11
13	Desarrollo de mantenimiento de mesas	2	24/10/16	25/10/16	12
14	Desarrollo defacturación	3	26/10/16	28/10/16	13
15	Desarrollo de mantenimiento de clientes	3	31/10/16	2/11/16	14
16	Diseño del Data Warehouse	6	3/11/16	10/11/16	15
17	Creación del ETL	5	11/11/16	17/11/16	16
18	Procedimiento de Carga de datos	4	18/11/16	23/11/16	17
19	Diseño de reportería	6	24/11/16	1/12/16	18
20	Diseño de la Base de conocimiento	3	2/12/16	6/12/16	19
21	Creación de algoritmo del Sistema Experto	6	7/12/16	14/12/16	20
22	Procedimiento de Carga de datos	3	15/12/16	19/12/16	21
23	Creación del módulo de seguridad	2	20/12/16	21/12/16	22
24	Módulo del cierre del sistema	2	22/12/16	23/12/16	23
25	Implementación del sistema	2	26/12/16	27/12/16	24
26	Prueba en funcionamiento	50	28/12/16	7/03/17	25
27	Recopilación de datos	7	8/03/17	16/03/17	26
28	Análisis de resultados	15	17/03/17	6/04/17	27

3.2.1. Descripción de las actividades

En el apartado se describirán los objetivos, recursos, entradas y salidas de cada actividad, que se realizará para el desarrollo del sistema experto.

A) Análisis del proceso de ventas

Objetivo: Analizar el proceso de ventas desde la llegada del cliente hasta el pago de los pedidos.

Recursos: Se tomarán 7 días para el desarrollo de la actividad; en la cual participarán las personas involucradas en el proceso de ventas y el tesista.

Entradas: No aplica.

Salidas: Análisis del proceso de ventas.

B) Identificar los puntos críticos:

Objetivo: Identificar los puntos más importantes del proceso de ventas.

Recursos: Se tomarán 5 días para el desarrollo de la actividad, en la cual participarán las personas involucradas en el proceso de ventas y el tesista.

Entradas: Análisis del proceso de ventas.

Salidas: Identificación de los puntos críticos del proceso de ventas.

C) Análisis de los requerimientos:

Objetivo: Analizar los requerimientos técnicos y funcionales para el desarrollo del sistema.

Recursos: Se tomarán 5 días para el desarrollo de la actividad, en la cual participarán las personas involucradas en el proceso de ventas y el tesista.

Entradas: Identificación de los puntos críticos del proceso de ventas.

Salidas: Análisis de los requerimiento.

D) Determinar los módulos del sistema:

Objetivo: Definir los módulos que tendrá el sistema

Recursos: Se tomarán 4 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Análisis de los requerimientos.

Salidas: Definición de los módulos del sistema.

E) Diseño de la BD transaccional:

Objetivo: Realizar el diseño de la base de datos transaccional.

Recursos: Se tomarán 7 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Análisis de los requerimientos.

Salidas: Diagrama de entidad y relación.

F) Creación de los objetos de la BD:

Objetivo: Crear las tablas, procedimientos almacenados, funciones y sinónimos de la base de datos.

Recursos: Se tomarán 5 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Diagrama de entidad y relación.

Salidas: Creación de los objetos de la base de datos.

G) Desarrollo mantenimiento del sistema:

Objetivo: Desarrollar los módulos de mantenimiento de usuarios, roles y de asignación de actividades.

Recursos: Se tomarán 3 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Definición de los módulos del sistema.

Salidas: Módulo de mantenimiento de usuarios, roles y de asignación de actividades.

H) Desarrollo mantenimiento de insumos:

Objetivo: Desarrollar el módulo de mantenimiento de insumos y de almacén.

Recursos: Se tomarán 3 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Definición de los módulos del sistema.

Salidas: Módulo de mantenimiento de mantenimiento de insumos y de almacén.

I) Desarrollo mantenimiento de productos:

Objetivo: Desarrollar el módulo de mantenimiento de platos y productos de la empresa.

Recursos: Se tomarán 3 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Definición de los módulos del sistema.

Salidas: Módulo de mantenimiento de mantenimiento de platos y productos de la empresa.

J) Desarrollo mantenimiento de preparación:

Objetivo: Desarrollar el módulo de asignación de insumos a los productos de la empresa.

Recursos: Se tomarán 3 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Definición de los módulos del sistema.

Salidas: Módulo de mantenimiento de asignación de insumos a los productos de la empresa.

K) Desarrollo de promociones:

Objetivo: Desarrollar el módulo de promociones de los productos.

Recursos: Se tomarán 2 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Definición de los módulos del sistema.

Salidas: Módulo de promociones de los productos.

L) Desarrollo de pedidos:

Objetivo: Desarrollar el módulo de ingreso de la comanda de los pedidos.

Recursos: Se tomarán 3 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Definición de los módulos del sistema.

Salidas: Módulo de pedidos.

M) Desarrollo de mantenimiento de mesas:

Objetivo: Desarrollar el módulo de mantenimiento de las mesas del restaurante.

Recursos: Se tomarán 2 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Definición de los módulos del sistema.

Salidas: Módulo de mantenimiento de las mesas del restaurante.

N) Desarrollo de facturación:

Objetivo: Desarrollar el módulo de caja y facturación.

Recursos: Se tomarán 3 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Definición de los módulos del sistema.

Salidas: Módulo de caja y facturación.

O) Desarrollo de mantenimiento de clientes:

Objetivo: Desarrollar el módulo de mantenimiento de los clientes.

Recursos: Se tomarán 3 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Definición de los módulos del sistema.

Salidas: Módulo de mantenimiento de clientes.

P) Diseño del Data Warehouse:

Objetivo: Crear la arquitectura del data warehouse de los datos transaccionales.

Recursos: Se tomarán 6 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Creación de los objetos de la base de datos.

Salidas: Creación del data warehouse.

Q) Creación del ETL:

Objetivo: Crear el proceso automatizado para la población del data warehouse.

Recursos: Se tomarán 5 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Base de datos transaccional.

Salidas: Proceso de población de datos para el data warehouse.

R) Procedimiento de Carga de datos:

Objetivo: Crear el proceso de ejecución automatizado.

Recursos: Se tomarán 4 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Proceso de población de datos para el data warehouse.

Salidas: Programación de carga de datos al data warehouse.

S) Diseño de reportería:

Objetivo: Crear los reportes del sistema.

Recursos: Se tomarán 6 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Data warehouse.

Salidas: Reportes del sistema.

T) Diseño de la Base de conocimiento:

Objetivo: Diseñar la arquitectura de la base de datos de conocimiento.

Recursos: Se tomarán 3 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Data warehouse.

Salidas: Base de conocimiento.

U) Creación de algoritmo del Sistema Experto:

Objetivo: Diseñar y crear los algoritmos de creación del sistema experto para que pueda tomar decisiones propias.

Recursos: Se tomarán 6 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Base de conocimiento.

Salidas: Algoritmo de decisiones del sistema experto.

V) Procedimiento de Carga de datos:

Objetivo: Ingresar el conocimiento de los expertos a la base de datos de conocimiento.

Recursos: Se tomarán 3 días para el desarrollo de la actividad, en la cual participará el tesista y las personas expertas del proceso de ventas.

Entradas: Conocimiento de los expertos.

Salidas: Creación de las reglas en el sistema experto.

W) Creación del módulo de seguridad:

Objetivo: Crear el proceso de respaldo de las bases de datos del sistema.

Recursos: Se tomarán 2 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Bases de datos del sistema.

Salidas: Proceso automatizado de back ups de las bases de datos del sistema.

X) Módulo del cierre del sistema:

Objetivo: Diseñar y crear el módulo del cierre del sistema.

Recursos: Se tomarán 2 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Bases de datos del sistema.

Salidas: Proceso automatizado del cierre del sistema.

Y) Implementación del sistema:

Objetivo: Implementar el sistema en el restaurante seleccionado.

Recursos: Se tomarán 2 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Sistema experto.

Salidas: Implementación del sistema experto.

Z) Prueba en funcionamiento:

Objetivo: Seguimiento al sistema, luego de la implementación.

Recursos: Se tomarán 50 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Sistema experto.

Salidas: Datos de la implementación.

AA) Recopilación de datos:

Objetivo: Recopilar los datos del sistema luego de la implementación.

Recursos: Se tomarán 7 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Datos del sistema.

Salidas: Resultados de la implementación.

BB) Análisis de resultados:

Objetivo: Analizar el impacto del sistema.

Recursos: Se tomarán 15 días para el desarrollo de la actividad, en la cual participará el tesista.

Entradas: Resultados de la implementación.

Salidas: Análisis de impacto.

3.2.2. Análisis del Costo – Beneficio

Para realizar el análisis del costo beneficio se realizará una tabla con los costos que generará el sistema y los beneficios que generará por medio de la implantación.

El beneficio que aportará el sistema, será estimado en un 15 por ciento anual en el aumento de las ventas.

El promedio de las ganancias mensuales en el restaurante; según el análisis del flujo de caja es de 6 000 soles mensuales; entonces para poder calcular el monto que generará el sistema, se realizará la siguiente fórmula:

$$10800 = \frac{6000 \times 12 \times 15}{100}$$

Sabiendo que la ganancia que generará el sistema será de 10800 soles anuales, se realizará una tabla de los costos que generará el sistema.

Tabla 2: Análisis de costo – beneficio

Fuente: Creación propia

Costos Anual	Cantidad	Valor monetario S/. Unidad	Valor monetario S/. Final
Luz	12	80	960
Internet	12	120	1440
Computadora	1/5	4000	800
Tablet	5	570	2850
Soporte Técnico	12	400	4800
Disco Duro	1	200	200
Repetidor de Wi fii	1	200	200
Total			11250

Para poder realizar el coste beneficio se realizará la siguiente fórmula

$$\text{Coste – Beneficio} = \frac{\text{Beneficio}}{\text{Coste}}$$

Si reemplazamos los valores obtenidos obtendremos lo siguiente:

$$0,96 = \frac{10800}{11250}$$

Como nuestro indicador de costo – beneficio es mayor a 1 se puede llegar a la conclusión que el proyecto es rentable; ya que por sol invertido se tendrá de ganancia 0,96 soles.

a) Especificación de requisitos del software

En este apartado se analizarán los requisitos que se necesitará el sistema para el desarrollo e implantación del sistema experto.

3.2.3. Introducción

En este apartado se definirá el propósito, el alcance, las definiciones, siglas, abreviaturas y las referencias.

A) Propósito

El propósito de los requisitos de software, será identificar los requerimientos que necesitará el sistema para su implantación; estos requisitos estarán dirigidos a la empresa que incorporará el sistema en su proceso de ventas.

B) Alcance

El sistema experto llevará el nombre de “Mi restaurante experto”. Tendrá la funcionalidad de realizar el proceso de ventas de forma automatizada, además de realizar el poblamiento automatizado de su data warehouse, para que por medio de este, pueda realizar los cálculos en la base de datos del conocimiento del sistema; tomando decisiones propias del sistema.

El principal beneficio que generará el sistema, es la mejora de la eficiencia en el proceso de ventas del restaurante.

El sistema usará una computadora; la cual tendrá la función de ser el servidor del sistema, teniendo como clientes de este servidor los dispositivos móviles y las computadoras del restaurante.

C) Definiciones, siglas y abreviaciones

- **DM_Restaurante:** Nombre de la base de datos transaccional del sistema.
- **DM_Restaurante:** Nombre del data warehouse del sistema.
- **KB_Restaurante:** Nombre de la base de datos del conocimiento del sistema.
- **ETL:** Proceso de carga automática y poblamiento del data warehouse desde la base de datos DM_Restaurante.
- **Job:** Programación automatizada para la toma de decisiones por parte de la base de datos KB_Restaurante.
- **Sistema de Reportería:** Servidor de reportería del sistema, dicho servidor, usa a reporting services como su gestor de información.

D) Referencias

Se usó el estándar IEEE Std. 830 para la realización de los requisitos del software.

3.2.4. Apreciación global

En este apartado se definirán la perspectiva del producto, los diagramas UML, diagramas de secuencias y diagramas de clases.

A) Perspectiva del producto

El producto tendrá la funcionabilidad de realizar el proceso de ventas en el restaurante; a su vez permitirá el análisis de sus datos y la toma de decisiones de acuerdo a circunstancias que se presenten en el día a día para el incremento de las ventas. El sistema experto tendrá una interfaz web para el uso del mismo.

B) Modelo entidad Relación

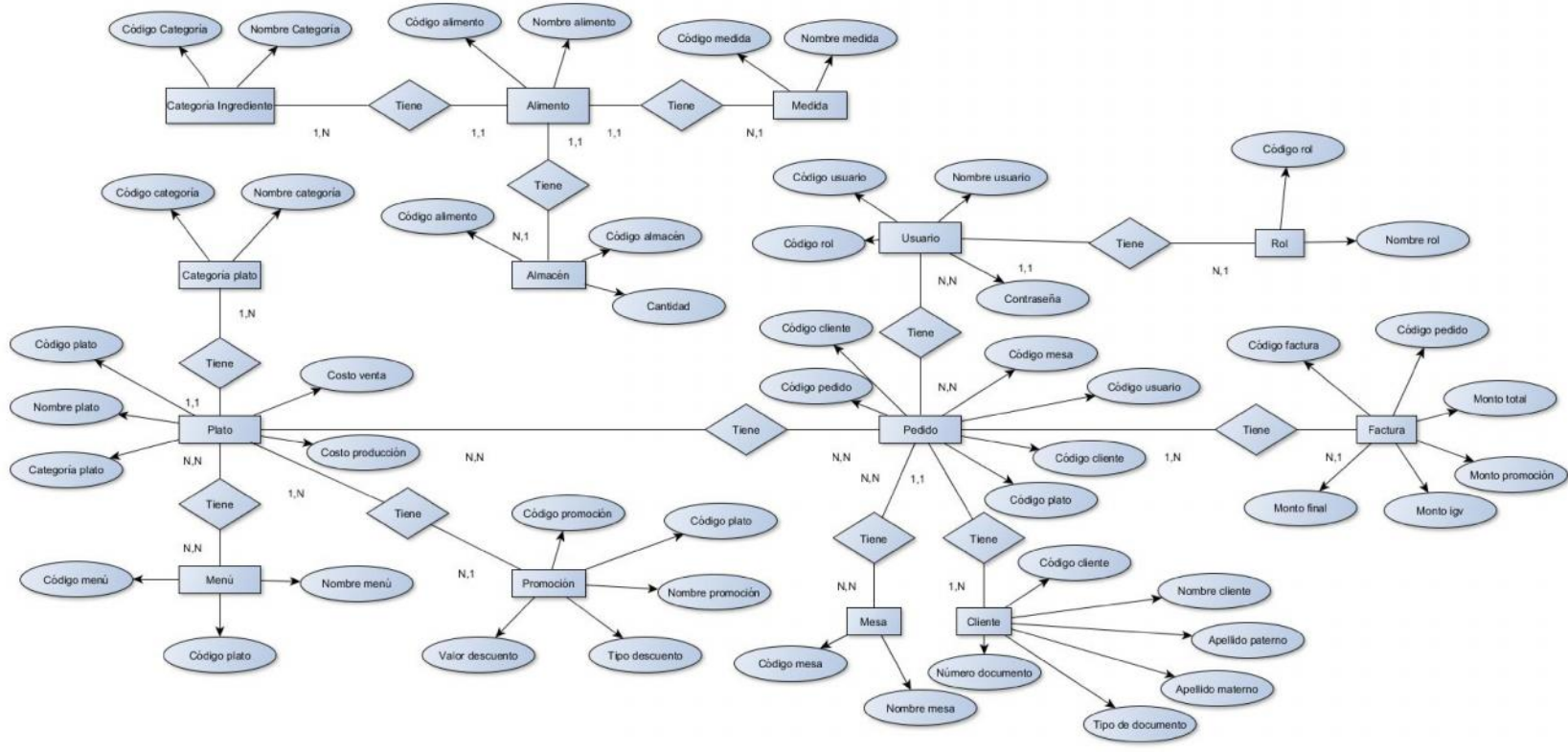


Figura 2: Entidad y relación de la base de datos transaccional.

Fuente: Creación propia.

C) Diagramas UML

En este apartado se mostrarán los diagramas UML de las personas que intervendrán con el sistema experto en el proceso de ventas.

El detalle de los flujos principal como alternativo de los casos de uso se encuentra en el Anexo E.

o Proceso de registro de almacén

En la figura 3 se puede observar el diagrama UML del proceso de registro de almacén, realizado por el encargado de ventas.

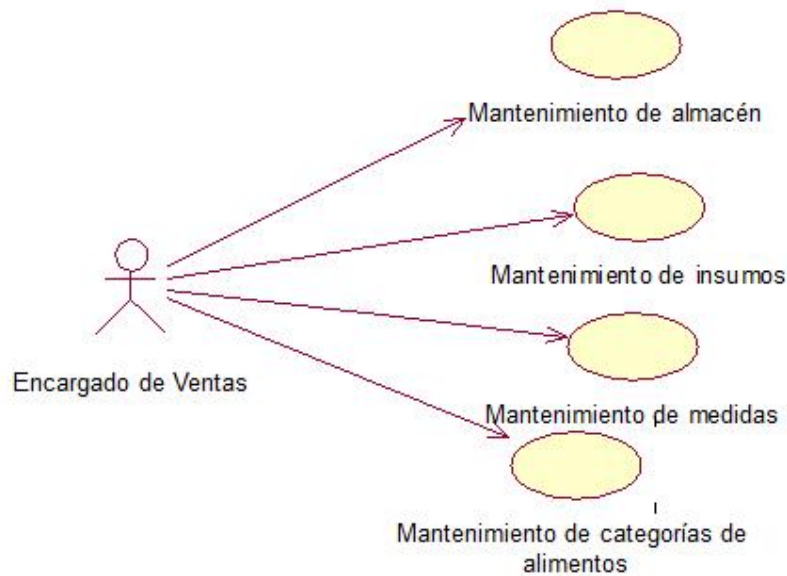


Figura 3: UML de registro almacén.

Fuente: Creación propia.

- **Caso de uso Mantenimiento de almacén**

Objetivo: Registrar la cantidad de productos ingresada al almacén, ver la cantidad máxima permitida por el mismo y el tiempo de caducidad del producto.

- **Caso de uso Mantenimiento de insumos**

Objetivo: Crear los insumos y el costo de los ingredientes.

- **Caso de uso Mantenimiento de medidas**

Objetivo: Crear las medidas de los productos.

- **Caso de uso Categoría de alimentos**

Objetivo: Ingreso de categorías de alimentos.

- **Proceso de Ventas**

En la figura 4 se observa el diagrama UML en el se identifican las actividades por el encargo del sistema. En la figura 5 se observa las actividades que tendrá a cargo el encargado de ventas, a su vez en la figura 6 se muestra las actividades que tendrá a cargo el cajero, en la figura 7 se observa las actividades que tendrá a cargo el encargado de cocina y por último en la figura 8 se observan las actividades que presentará el mesero.

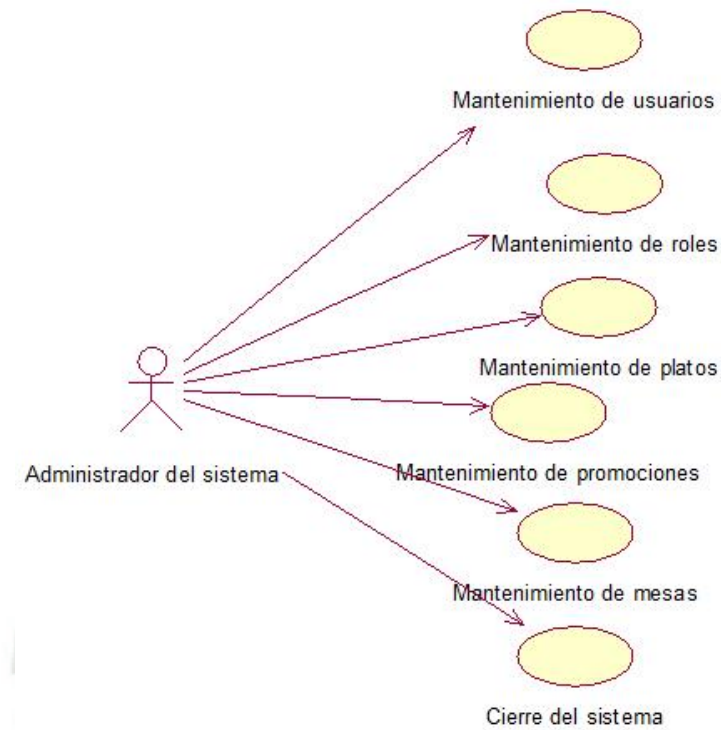


Figura 4: UML del administrador del sistema.

Fuente: Creación propia.

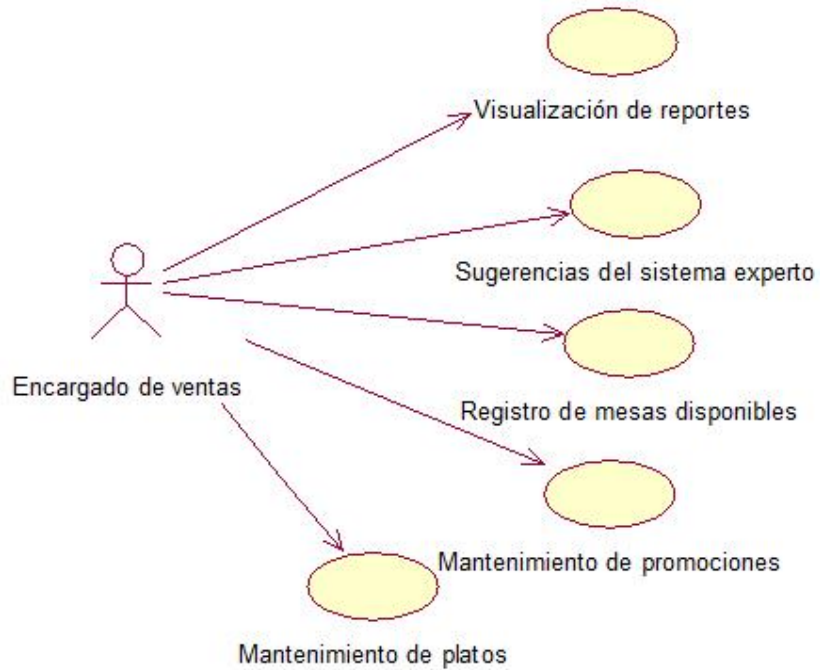


Figura 5: UML del encargado de ventas.

Fuente: Creación propia.

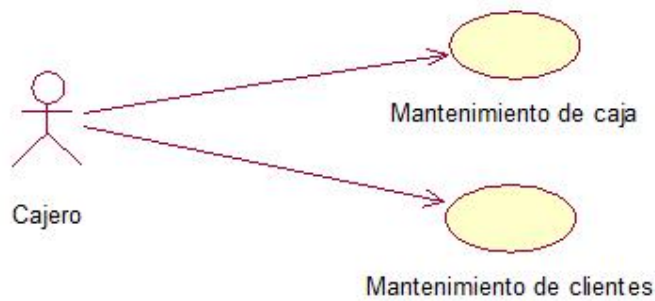


Figura 6: UML del cajero.

Fuente: Creación propia.



Figura 7: UML del encargado de cocina.

Fuente: Creación propia.

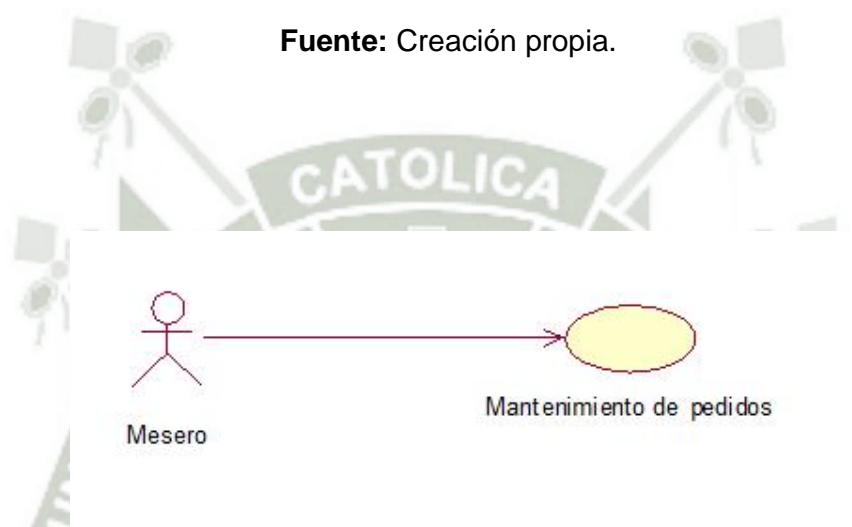


Figura 8: UML del mesero.

Fuente: Creación propia.

- **Caso de uso de Mantenimiento de usuarios**

Objetivo: Crear, actualizar y asignar los roles a los usuarios.

- **Caso de uso de Mantenimiento de roles**

Objetivo: Crear roles y asignar los módulo a cada rol.

Flujo alternativo: Si el rol ya existe en el sistema, le mostrará un mensaje al usuario que el rol ya está registrado en el sistema.

- **Caso de uso de Mantenimiento de platos**

Objetivo: Creación de los productos de la empresa

- **Caso de uso de Mantenimiento de promociones**

Objetivo: Crear promociones de acuerdo a una determinada fecha, para que sea aplicada de acuerdo a un monto fijo o a un descuento porcentual.

- **Caso de uso de Mantenimiento de mesas**

Objetivo: Registro de las mesas del restaurante.

- **Caso de uso de Cierre del sistema**

Objetivo: Realizar una copia de seguridad y la población de la base de datos del data warehouse y de la base de datos del conocimiento, además de cambiar la fecha del sistema.

- **Caso de uso de Visualización de reportes**

Objetivo: Presentar los reportes para que el usuario pueda tomar decisiones a través de datos históricos como de las predicciones por el sistema.

- **Caso de uso de Sugerencias del sistema experto**

Objetivo: Presentar posibles promociones de los productos menos vendidos, además de identificar la relación de ventas de los productos de un determinado día de la semana.

- **Caso de uso de Registro de mesas disponibles**

Objetivo: Identificar el estado de las mesas del establecimiento.

- **Caso de uso de Mantenimiento de caja**

Objetivo: Cobrar a los consumidores por medio de la facturación.

- **Caso de uso de Mantenimiento de los clientes**

Objetivo: Crear y actualizar los datos de los clientes.

- **Caso de uso de Mantenimiento de preparación de platos**

Objetivo: Asignar la cantidad exacta de insumos

- **Caso de uso de Mantenimiento de pedidos**

Objetivo: Ingresar la comanda generada por los clientes.

- **Características del usuario**

Las características o roles de los usuarios son los siguientes:

- **Administrador del sistema**

Es el encargado de administrar los permisos a los usuarios, además de la creación de nuevos productos a la empresa.

- **Encargado de Ventas**

Es la persona responsable de tomar las decisiones para el crecimiento de la empresa, por medio del aumento de las ventas, además de asegurarse del correcto funcionamiento del proceso de ventas; también es el encargado de la compra de los insumos.

- **Cajero**

Es la persona que realiza la cancelación de los pedidos de los clientes.

- **Encargado de cocina**

Es la persona que recibe la orden del cliente para poder preparar los productos de la empresa utilizando los insumos necesarios para dicho producto.

- **Mesero**

Es la persona de recibir las órdenes de los clientes para poder entregársela al encargado de cocina.

D) Diagrama de secuencia

En el siguiente apartado se mostrará los diagramas de secuencia que presentará el sistema experto.

- **Mantenimiento de almacén**

En la figura 9 se observa el diagrama de secuencia del mantenimiento de almacén; el administrador del sistema debe ingresar al sistema y realiza una petición a los ingredientes y al almacén, el sistema devuelve estos datos, para que administrador del sistema registre los ingredientes en el módulo de almacén; para ello el administrador deberá ingresar el nombre del ingrediente y la cantidad que se guardarán en el almacén.

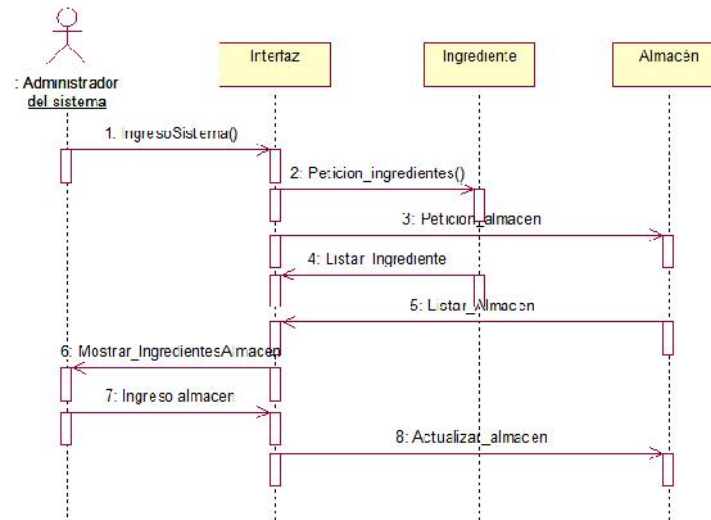


Figura 9: Diagrama de secuencia de mantenimiento de almacén.

Fuente: Creación propia.

- **Mantenimiento de insumos**

En la figura 10 se observa el diagrama de secuencia del mantenimiento de insumos, para ello el administrador del sistema deberá ingresar al sistema y realizará una petición al sistema de los insumos, el sistema listará los insumos registrados en el sistema, para que el administrador pueda registrar por medio de la interfaz el insumo nuevo o realizar una actualización de los insumos existentes.

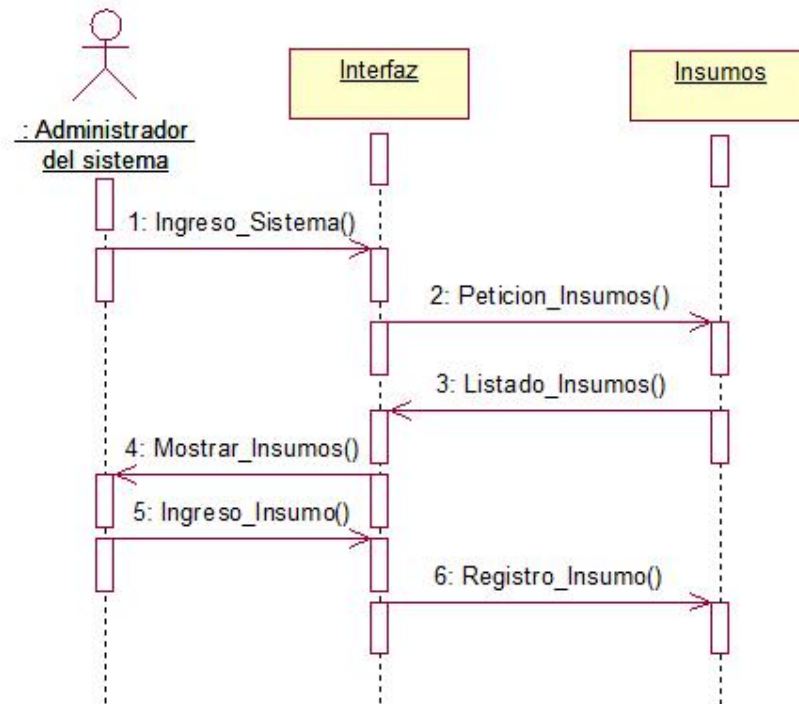


Figura 10: Diagrama de secuencia del mantenimiento de insumos.

Fuente: Creación propia.

- **Mantenimiento de medidas**

En la figura 11, se observa el diagrama de secuencia del mantenimiento de medidas, por ello el encargado de ventas, deberá ingresar al sistema por medio de la interfaz; la cual realizará una petición de las medidas la cuál listará las medidas existentes en el sistema. El encargado de ventas ingresará la nueva medida o actualizará una medida por medio de la interfaz.

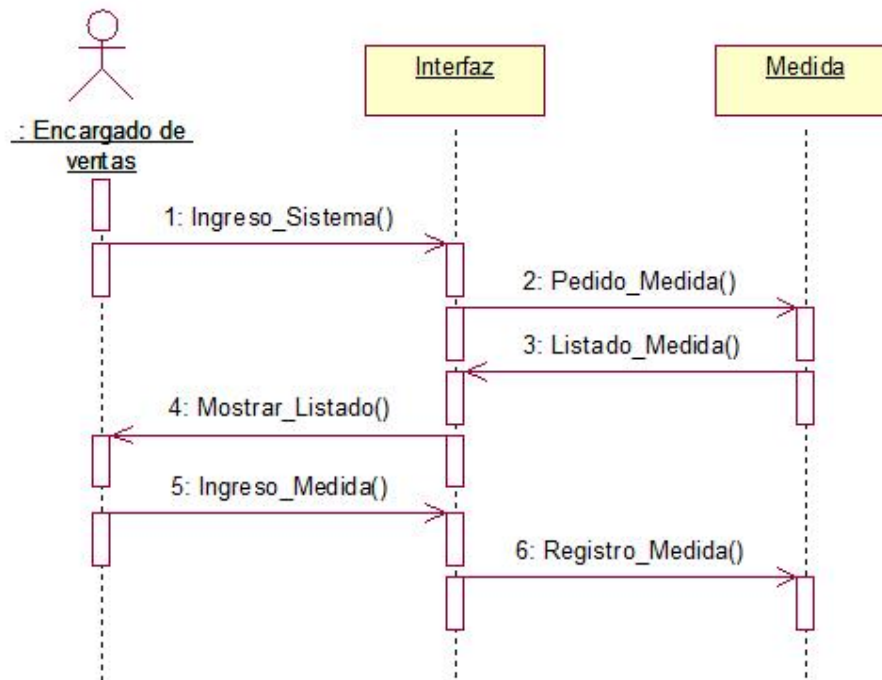


Figura 11: Diagrama de secuencia del mantenimiento de medidas.

Fuente: Creación propia.

- **Mantenimiento categoría ingrediente**

En la figura 12, se observa el diagrama de secuencia del mantenimiento de categoría de ingrediente, en la cual el encargado de ventas deberá ingresar al sistema por medio de la interfaz, la cual realizará dos peticiones, una a los ingredientes y la otra a la categoría de los ingredientes; la interfaz mostrará los datos de estos para que el encargado de ventas asigne los ingredientes a una determinada categoría. Luego de realizar la selección la interfaz actualizará la categoría de los ingredientes.

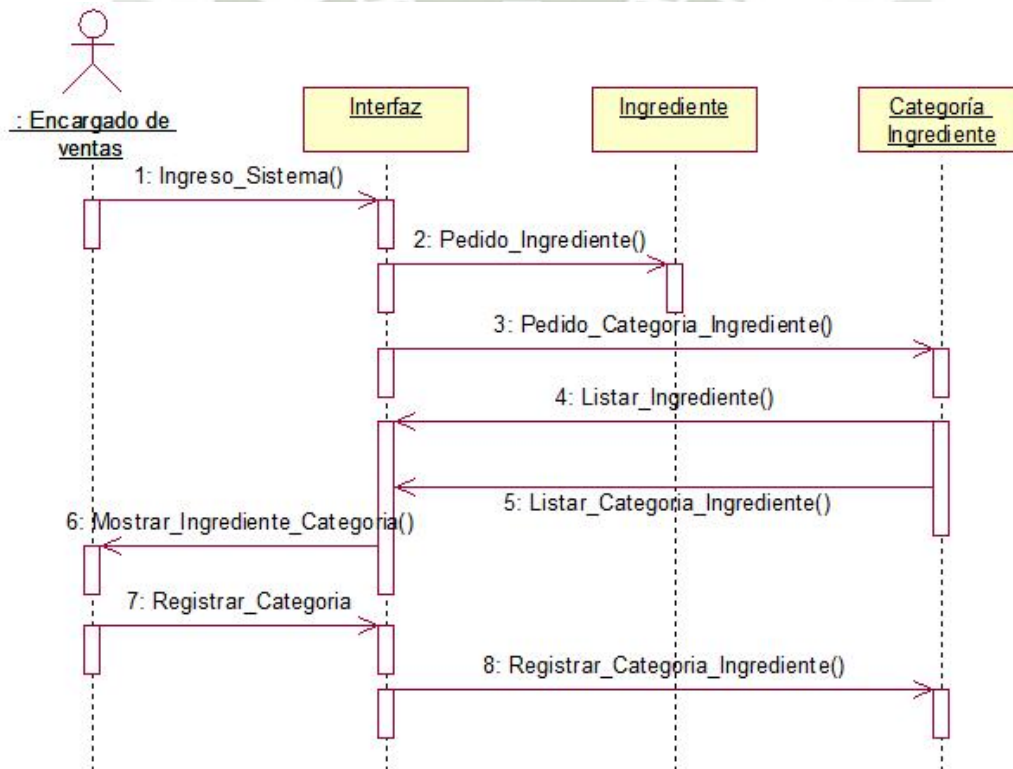


Figura 12: Diagrama de secuencia del mantenimiento de categoría de ingrediente.

Fuente: Creación propia.

- **Mantenimiento de usuarios**

En la figura 13, se observa el diagrama de secuencia del mantenimiento de usuarios, en la cual el administrador del sistema deberá ingresar al sistema por medio de la interfaz, este realizará una petición al sistema sobre los usuarios, devolverá el listado de los usuarios registrados en el sistema; por medio de la interfaz se mostrará todos los usuarios registrados en el sistema, el administrador del sistema ingresará al nuevo usuario por medio de la interfaz para poder registrarlo en el sistema.

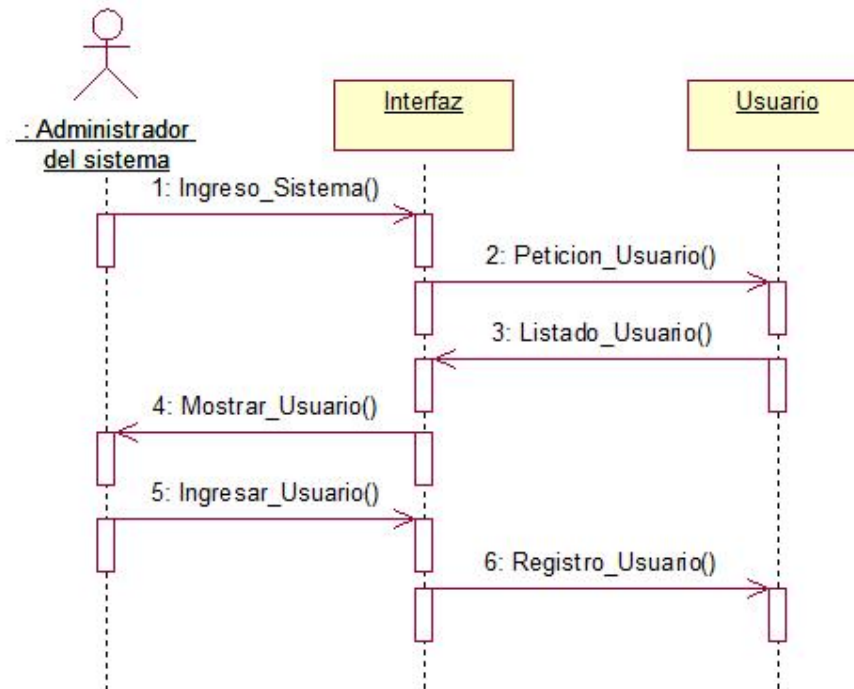


Figura 13: Diagrama de secuencia del mantenimiento de usuarios.

Fuente: Creación propia.

- **Mantenimiento de roles**

En la figura 14, se observa el diagrama de secuencia del mantenimiento de roles, en el cual el administrador del sistema, ingresa al sistema por medio de la interfaz y realizará una petición para que se pueda realizar el listado de roles en el sistema. La interfaz mostrará dicho listado para que el administrador del sistema ingrese un nuevo rol o actualice un rol en el sistema.

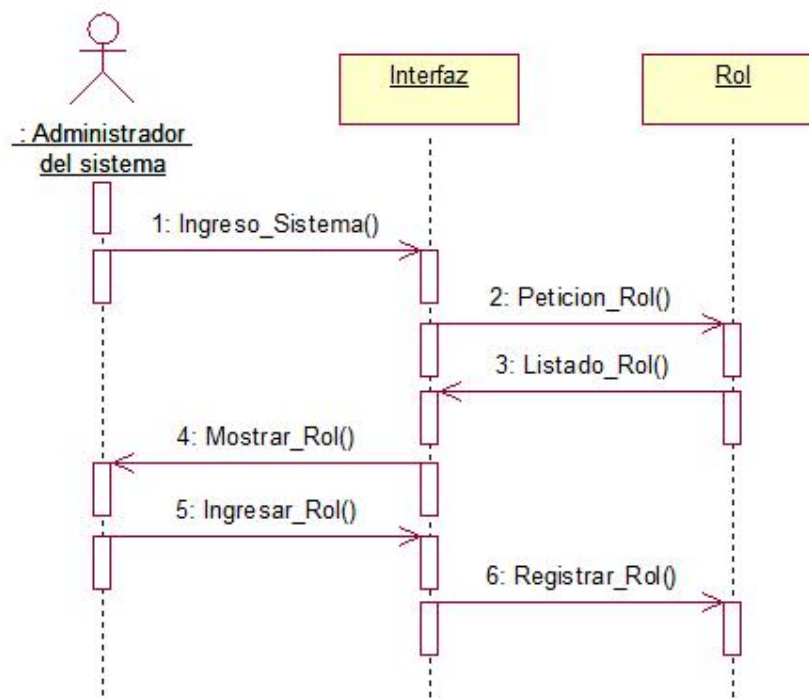


Figura 14: Diagrama de secuencia del mantenimiento de roles.

Fuente: Creación propia.

- **Mantenimiento de platos**

En la figura 15, se observa el diagrama de secuencia del mantenimiento de platos, en el cual el administrador del sistema ingresará al sistema por medio de la interfaz; la cual realizará una petición para obtener el listado de los platos. La interfaz mostrará el listado de los platos para que el administrador del sistema pueda crear o actualizar el plato; luego de su selección se registrará el plato en el sistema.

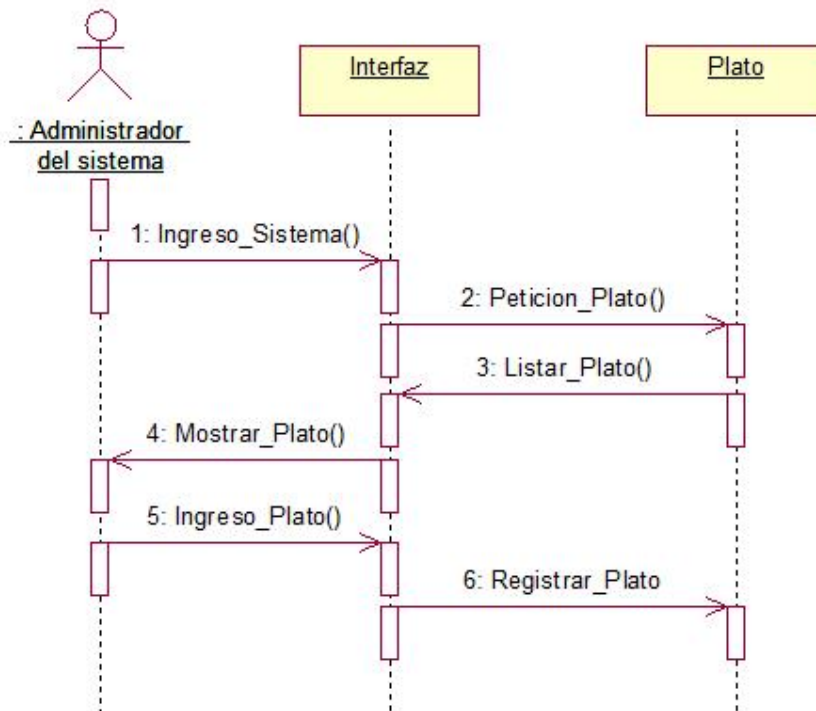


Figura 15: Diagrama de secuencia del mantenimiento de platos.

Fuente: Creación propia.

- **Mantenimiento de promociones**

En la figura 16, se observa el diagrama de secuencia del registro de promociones, en la cual el administrador del sistema ingresará al sistema por medio de la interfaz y realizará dos peticiones una al listado de platos y la otra al listado de promociones, la interfaz mostrará estos datos para que el administrador del sistema ingrese la promoción seleccionando a su vez un plato y registrando esta promoción en el sistema.

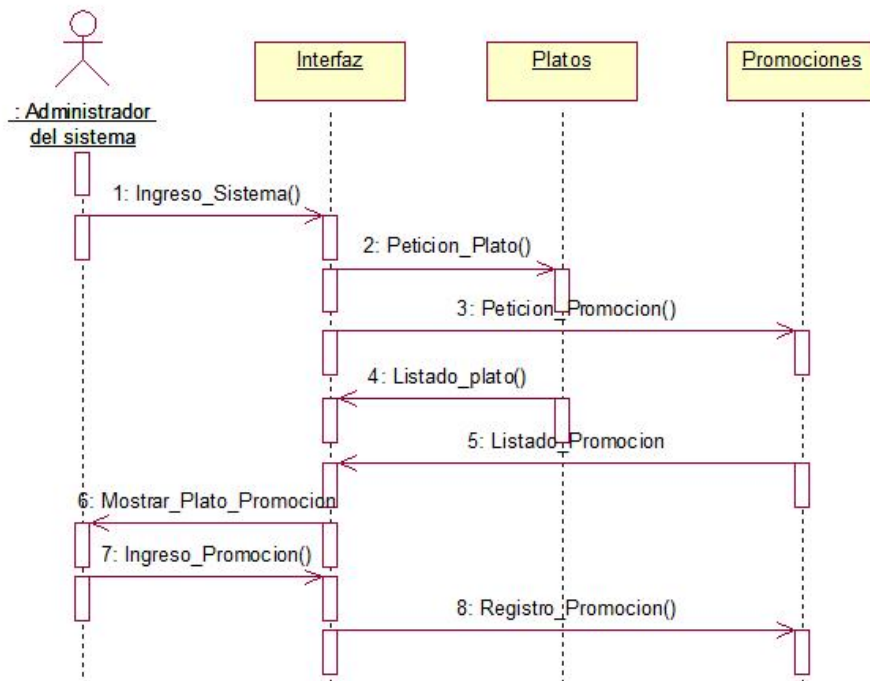


Figura 16: Diagrama de secuencia del mantenimiento de promoción.

Fuente: Creación propia.

- **Mantenimiento de mesas**

En la figura 17, se observa el diagrama de secuencia del mantenimiento de mesas, en el cual el administrador del sistema ingresará al sistema por medio de la interfaz, la cual realiza una petición para obtener el listado de las mesas; la interfaz muestra estos datos, para que el administrador del sistema ingrese o actualice una mesa existente y sea registrado en el sistema.

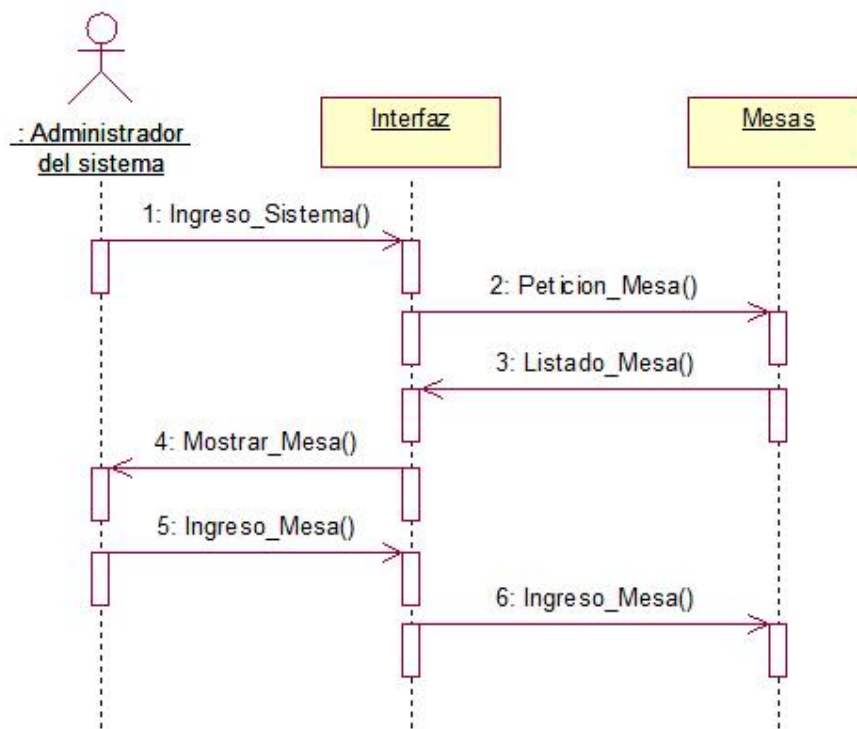


Figura 17: Diagrama de secuencia del registro de platos.

Fuente: Creación propia.

- **Cierre del sistema**

En la figura 18, se observa el diagrama de secuencia del cierre del sistema, en la cual el administrador del sistema ingresará a la interfaz, la cual mostrará la fecha de cierre, el administrador aceptará el cierre y actualizará los pedidos y modificará la fecha del sistema.

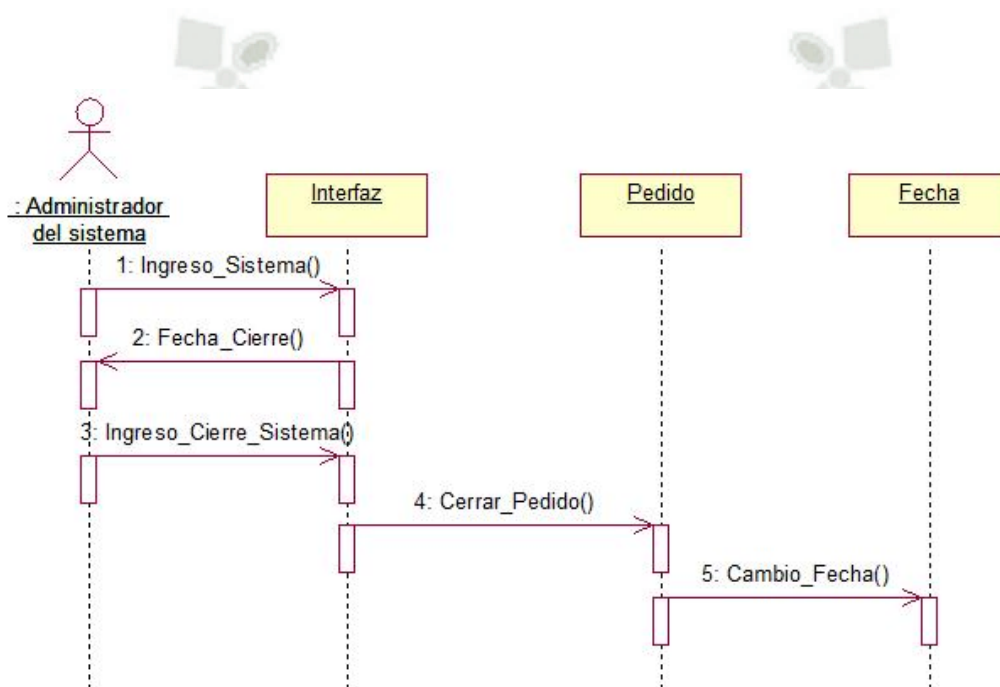


Figura 18: Diagrama de secuencia del cierre del sistema.

Fuente: Creación propia.

- **Visualización de reportes**

En la figura 19, se observa el diagrama de secuencia de la visualización de reportes, en el cual el encargado de ventas ingresará al sistema por medio de la interfaz el cual realizará un pedido para listar los reportes registrados en el sistema.

El administrador seleccionará el reporte y el sistema procesará el reporte seleccionado para que lo muestre por medio de la interfaz del sistema.

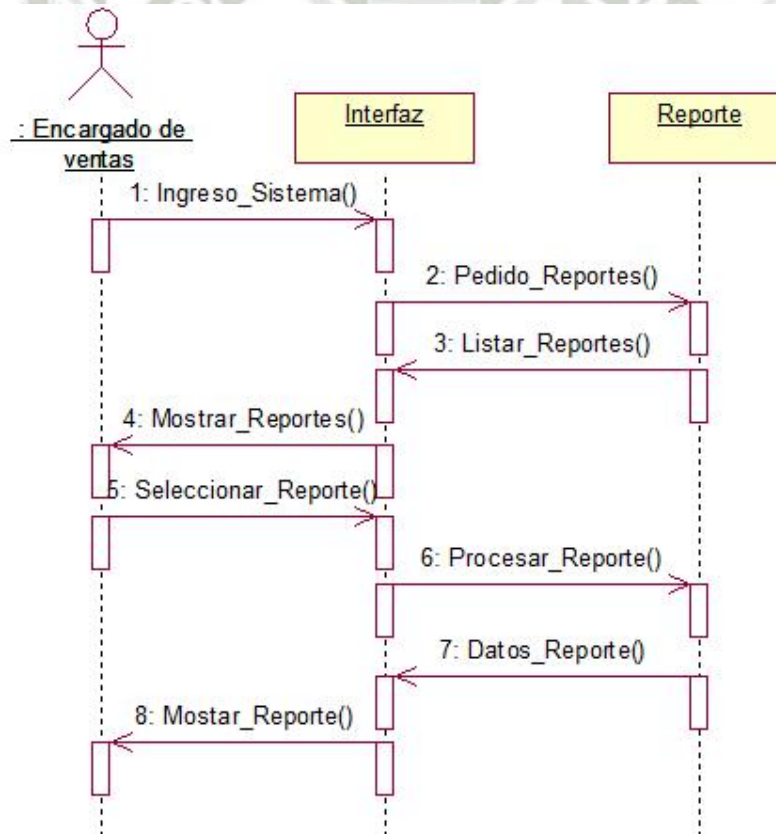


Figura 19: Diagrama de secuencia de la visualización de reportes.

Fuente: Creación propia.

- **Sugerencias del sistema experto**

En la figura 20 se observa el diagrama de secuencia de sugerencias del sistema experto, en el cual el encargado de ventas ingresará al sistema por medio de la interfaz el cual realizará un pedido para listar los reportes del sistema experto registrados en el sistema.

El administrador seleccionará el reporte del sistema experto y el sistema procesará el reporte seleccionado para que lo muestre por medio de la interfaz del sistema.

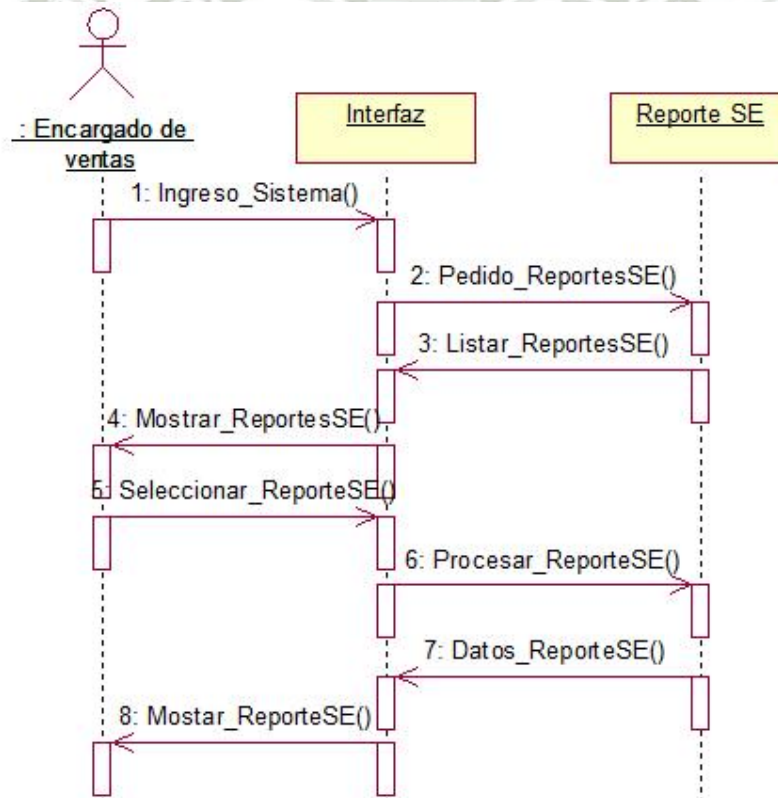


Figura 20: Diagrama de secuencia de sugerencia del sistema experto.

Fuente: Creación propia.

- **Mesas disponibles**

En la figura 21, se observa el diagrama de secuencia de las mesas disponibles, en el cual el encargado de ventas ingresará al sistema por medio de la interfaz, la cual realizará una petición de las mesas disponibles la cual se mostrarán por medio de la interfaz al encargado de ventas.

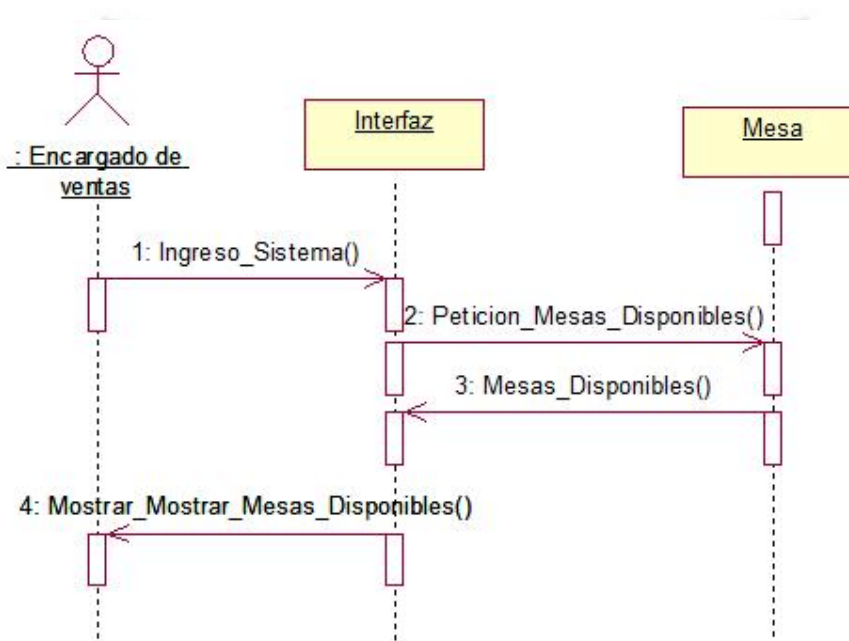


Figura 21: Diagrama de secuencia de mesas disponibles.

Fuente: Creación propia.

- **Mantenimiento de caja**

En la figura 22, se observa el diagrama de secuencia del mantenimiento de caja, en el cual el cajero ingresa al sistema por medio de la interfaz, realizando una petición de los pedidos de una determinada mesa. La interfaz mostrará los datos al cajero por medio de la interfaz para que pueda registrar el cobro de los pedidos de la mesa seleccionada.

El sistema actualizará la mesa y los pedidos y guardará la transacción de la facturación en el sistema.

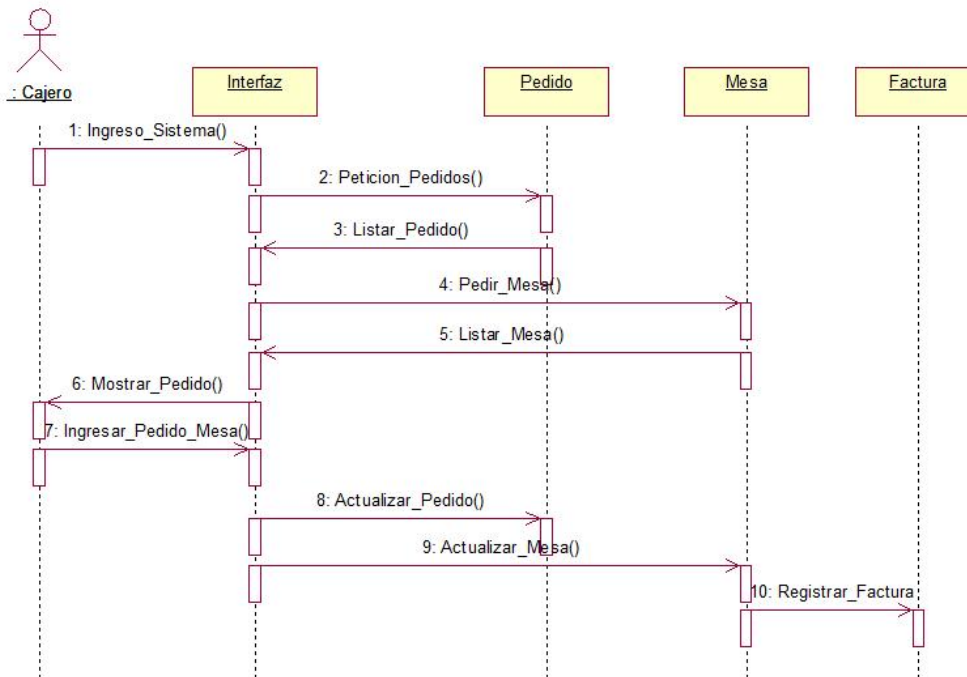


Figura 22: Diagrama de secuencia del mantenimiento de caja.

Fuente: Creación propia.

- **Mantenimiento de clientes**

En la figura 23, se observa el diagrama de secuencia del mantenimiento de clientes, en el cual el cajero ingresará al sistema por medio de la interfaz realizando una petición que retornará el listado de los clientes registrados en el sistema, el cajero ingresará al nuevo cliente o actualizará un cliente registrado en el sistema, para que sea guardado en el sistema.

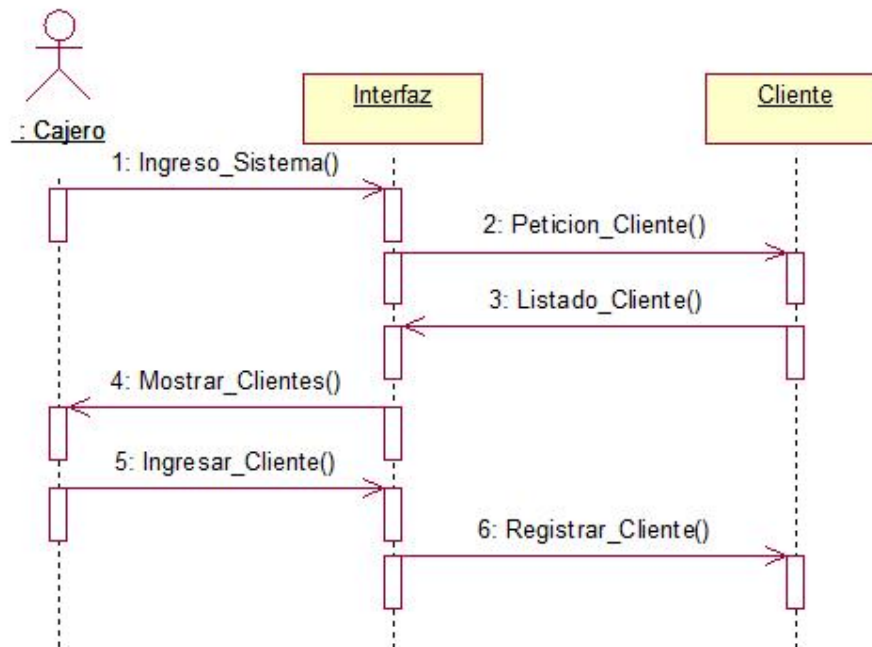


Figura 23: Diagrama de secuencia del mantenimiento de clientes.

Fuente: Creación propia.

- **Mantenimiento de pedidos**

En la figura 24, se observa el diagrama de secuencia del mantenimiento de pedidos, en el cual el mesero ingresará al sistema por medio de la interfaz, realizando una petición del listado de los platos, la interfaz mostrará este listado para que el mesero seleccione el plato seleccionado y sea registrado el pedido en el sistema.

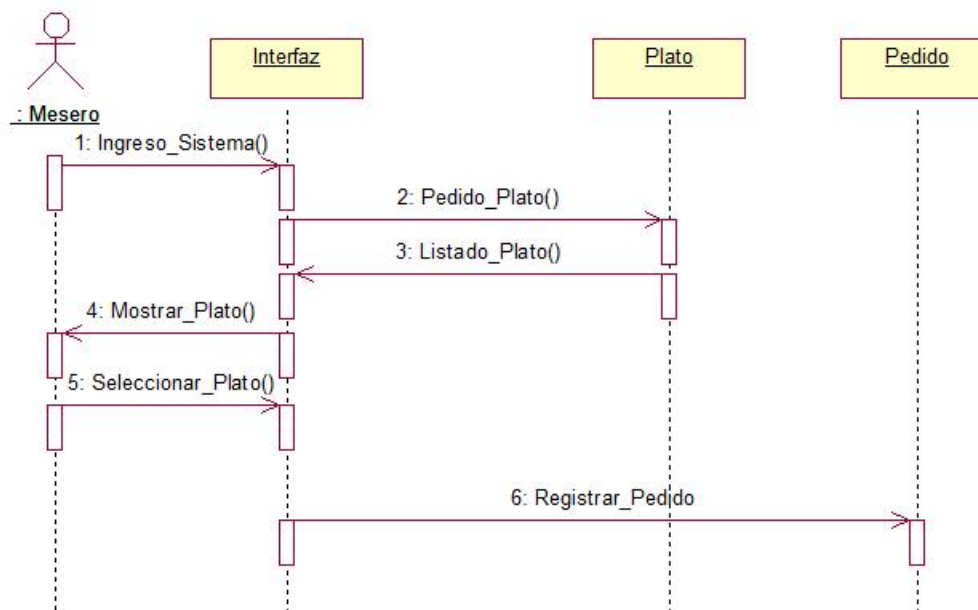


Figura 24: Diagrama de secuencia del mantenimiento de pedidos.

Fuente: Creación propia.

E) Diagrama de clases

En la figura 25 se observa el diagrama de clases del sistema.

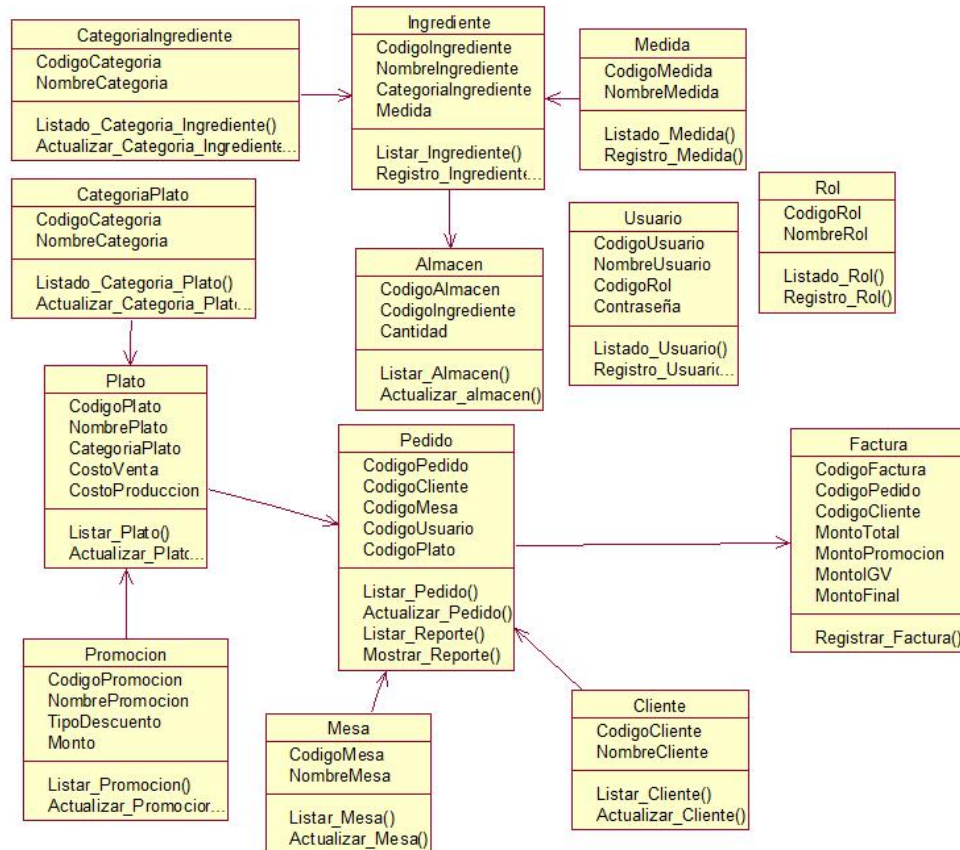


Figura 25: Diagrama de clases.

Fuente: Creación propia.

Se realizó los siguientes casos de uso que intervendrán con el sistema.

3.2.5. Requisitos específicos del sistema

Los requisitos que tendrá el sistema serán los siguientes:

- Tener un sistema de información que integre todo el proceso de ventas del restaurante, el cual registrará las transacciones del

ingreso de insumos, la asignación de los alimentos, el registro de los pedidos y el cobro de los mismos.

- Mantenimiento de usuarios y roles por medio del sistema, el cual permitirá la creación de diferentes roles y darle la capacidad de ingreso a los diferentes módulos del sistema.
- Mantenimiento de productos, el cual registrará los costos de ventas y realizar un cálculo del costo de producción de dicho producto.
- Mantenimiento de almacén, el cual permitirá el ingreso de las cantidades de los insumos al restaurante, en ella se asignará de acuerdo a las medidas registradas en el sistema.
- Mantenimiento de promociones, el cual permitirá realizar descuentos de diferentes tipos de categorías, las cuales serán por un descuento fijo o por un descuento porcentual.
- Mantenimiento de mesas, el cual permitirá registrar el nombre de las mesas en el sistema.
- Creación de los pedidos, el cual permitirá el ingreso de las comandas de los clientes, para que el tiempo de producción de estos se realice de forma más eficiente.
- Facturación de los pedidos, el cual permitirá el revisar los pedidos de los clientes y proceder con el pago de estos.
- Reportes diarios como históricos, dado a este módulo se podrá analizar el estado actual e histórico del restaurante para que sea de principal importancia en la toma de decisiones de este.

- Análisis de los datos a través de un data warehouse; se creará este proceso para que se analicen los datos de una forma más estructurada; además de evitar la sobrecarga de datos en la base de datos transaccional del sistema.
- Seguridad de los datos, por medio de un sistema de backup de las bases de datos que intervendrán en el desarrollo del sistema.
- Creación de un sistema experto que genere promociones automáticas, el cual tendrá como función realizar una evaluación en línea del restaurante para que pueda tomar decisiones de acuerdo a las reglas preestablecidas en el sistema.
- Cierre automático del sistema, permitirá el poblamiento del data warehouse; además de sacar un backup de las bases de datos del sistema y la actualización de la base de datos de conocimiento del sistema.
- El sistema deberá estar en un entorno web, se encontrará en dicho entorno para que tenga un acceso desde diferentes tipos de dispositivos.
- Acceso desde diferentes dispositivos ya sea desde dispositivos de escritorios como dispositivos móviles.
- Identificar la relación de las ventas entre los productos, el cual se realizará un proceso interno en el proceso de cierre diario del sistema para poder calcular la relación de los productos y tomar decisiones de acuerdo a esta relación de productos.

3.2.6. Restricciones del sistema

Las restricciones que presenta el sistema “Mi Restaurante Experto” son las siguientes:

- Tendrá que tener una conexión a internet permanente.
- Espacio mínimo de 500 G de disco de memoria en el servidor.
- Memoria Ram mínima de 4 GB en el servidor
- Funcionará en una red local.
- Los dispositivos cliente deben tener acceso a internet por medio de la conexión Wi fii.

Los requisitos no funcionales serán los siguientes:

- El sistema deberá soportar diversas conexiones simultáneamente.
- El sistema sólo podrá ser accedido por los usuarios registrados en el sistema.
- Se obtendrá backups de forma diaria en una ubicación predefinida, la cual deberá ser transportada a otra locación fuera del recinto del restaurante para poder tener una mejor seguridad de los datos.
- Las bases de datos deberán tener integridad de datos.
- La base de datos deberá tener una estructura en cuarta forma normal, la cual optimizará el desempeño de esta y evitará la redundancia de los datos.

A) Especificación del diseño

Se analizó el proceso de ventas del restaurante, para poder analizar el estado actual del proceso de ventas, una de las características principales fue que todo el registro de los pedidos se realizaba de forma manual, es decir que no utilizaban ningún software para el manejo de los datos.

El proceso de ventas se inicia con la realización del pedido por medio del cliente; el mesero obtiene la orden y crea un documento llamado comanda, dicho documento contiene el detalle del pedido del cliente; el mesero le entrega la comanda al cajero, generando la apertura de la cuenta de dicha mesa; dicha comanda se le entrega al cocinero para que lo pueda preparar y entregar dichos platos al mesero, quien es el encargado de llevarlo a la mesa de los comensales, en dicho transcurso de tiempo el cliente puede agregar más pedidos que serán agregados al documento comanda.

El cliente solicitará la cuenta al mesero, el cual pedirá la comanda al cajero, el mesero realizará el cobro del dinero hacia el cliente y este llevará el dinero a caja para poder generar el comprobante de pago, el cual se entregará al cliente, finalizando el proceso.

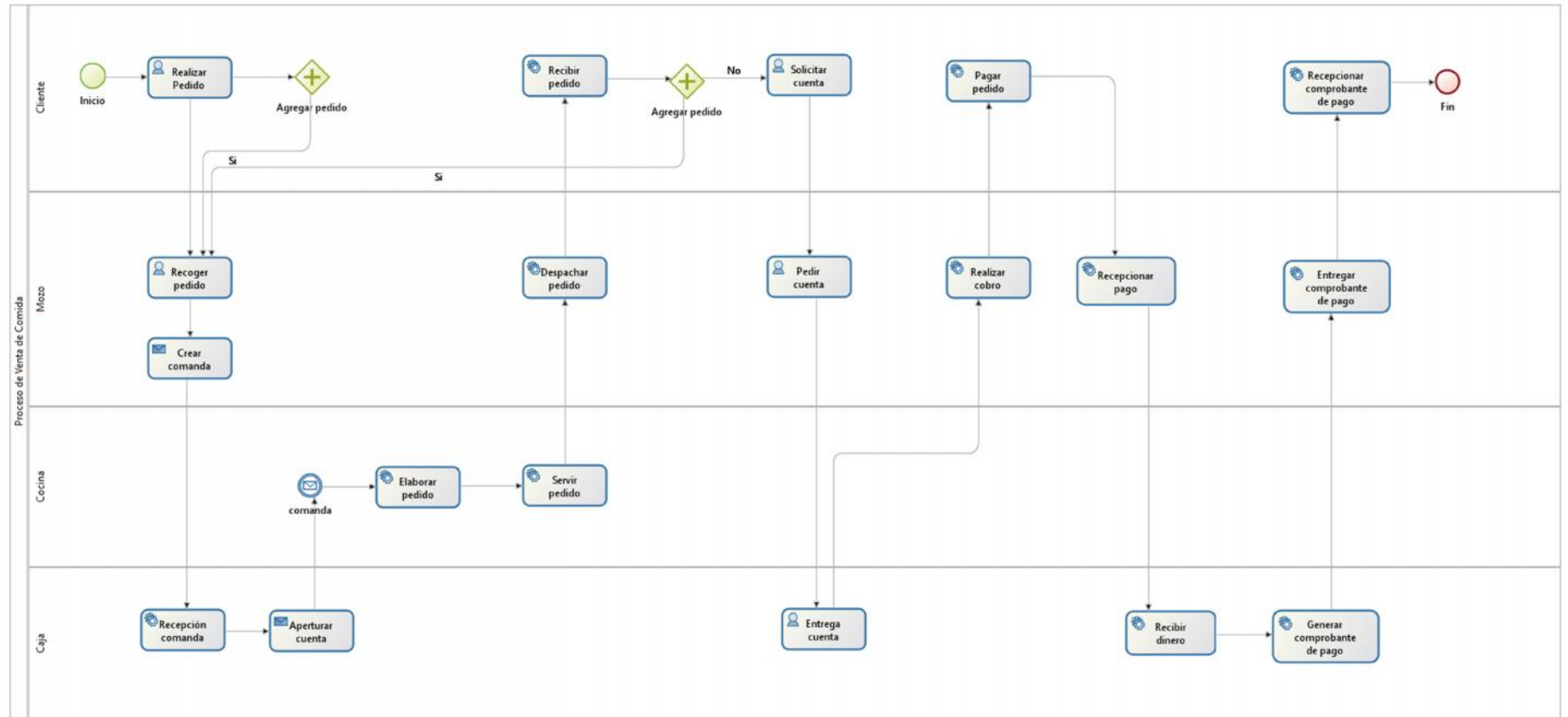


Figura 26: Proceso de ventas.

Fuente: Creación propia.

3.3. Arquitectura del sistema

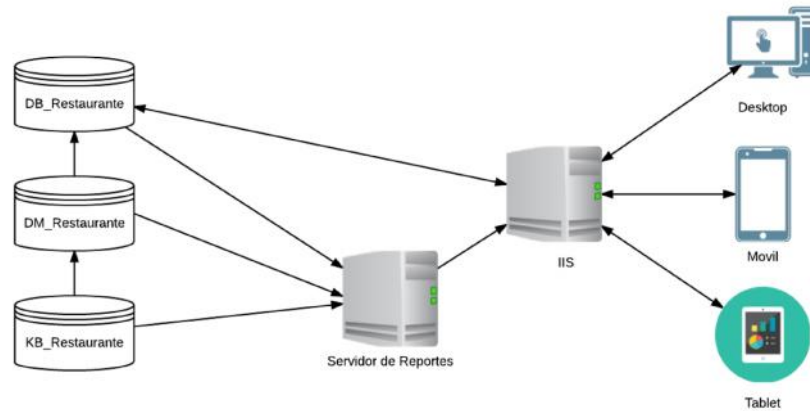


Figura 27: Arquitectura del sistema.

Fuente: Creación propia

A) DB_Restaurante

Es la base de datos transaccional del sistema, dicha base de datos envía los datos del sistema al servidor de reportes y al IIS, lugar donde se encuentran el sistema. Por medio del IIS se realizan las inserciones, actualización y eliminación de datos.

B) DM_Restaurante

Es el data warehouse, presenta la tabla de hechos de las ventas históricas y la tabla de hechos de almacén. Al cierre diario del sistema se ejecuta un script que realiza el ETL teniendo como fuente de datos origen la base de datos transaccional para poder poblar dicho data warehouse.

C) KB_Restaurante

Es la base de datos de conocimiento y se actualiza cada cierre diario por medio del data warehouse, realiza un algoritmo para calcular la relación de venta de productos y las reglas del mismo.

D) Servidor de reportes

El sistema de reportes utilizado es el SSRS, este sistema guarda los objetos, presenta 3 orígenes de datos los cuales son DB_Restaurante, DM_Restaurante y KB_Restaurante.

E) IIS

En el IIS se encuentra alojada la aplicación del sistema y es el único acceso para la escritura de datos del mismo. Desde el IIS se realizan las conexiones hacia la base de datos DB_Restaurante y hacia el servidor de reportes, ya que por ella se pueden visualizar los mismos.

F) Dispositivos de Acceso

El sistema está desarrollado de forma adaptativa por ende se puede acceder desde diversos dispositivos que se encuentren en la misma red local, por lo tanto al sistema se puede acceder desde terminales como computadoras; así como también desde dispositivos móvil es decir Smartphone y Tablets.

3.4. Diseño de las bases de datos

El sistema tendrá 3 bases de datos las cuales son:

- **DB_Restaurante:** Base de datos que guardará las transacciones del proceso de ventas.
- **DM_Restaurante:** Base de datos en la que conformarán dos datamarts, uno de las ventas y el otro de almacén
- **KB_Restaurante:** Base de datos del conocimiento, que se actualizará de forma diaria y tomará decisiones de acuerdo al día a día del negocio.

A) DB_Restaurante

Base de datos que guardará las transacciones del proceso de ventas.



B) Modelo relacional:

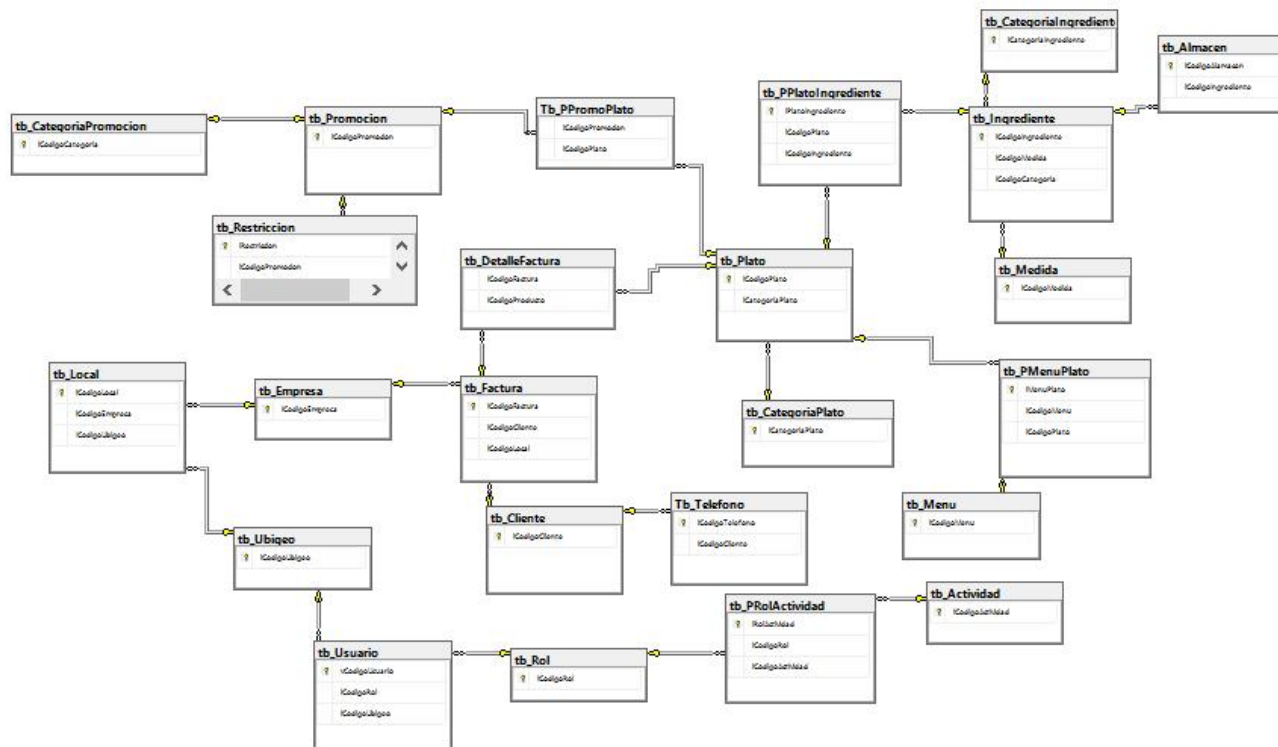


Figura 28: Modelo relacional de la base de datos transaccional.

Fuente: Creación propia.

El diccionario de datos de la base de datos se encuentra en el anexo A.

- **DM_Restaurantes**

Base de datos en la que conformarán dos datamarts, uno de las ventas y el otro de almacén.

El diccionario de datos de la base de datos se encuentra en el anexo A.

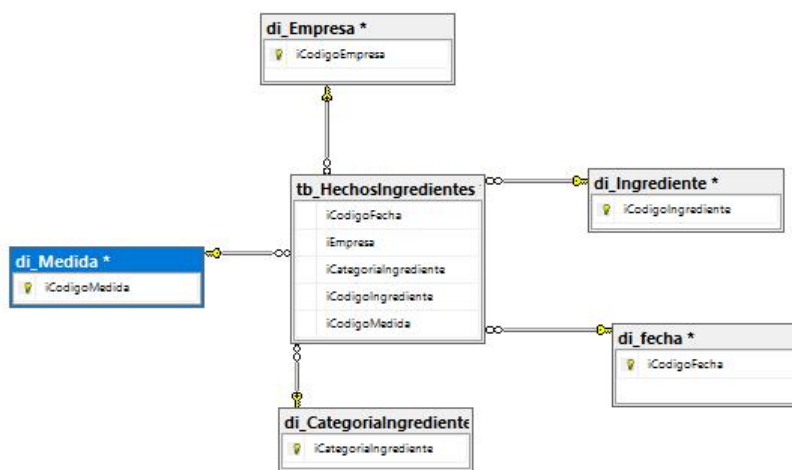


Figura 29: Modelo relacional del data warehouse de la tabla de hechos ingredientes.

Fuente: Creación propia.

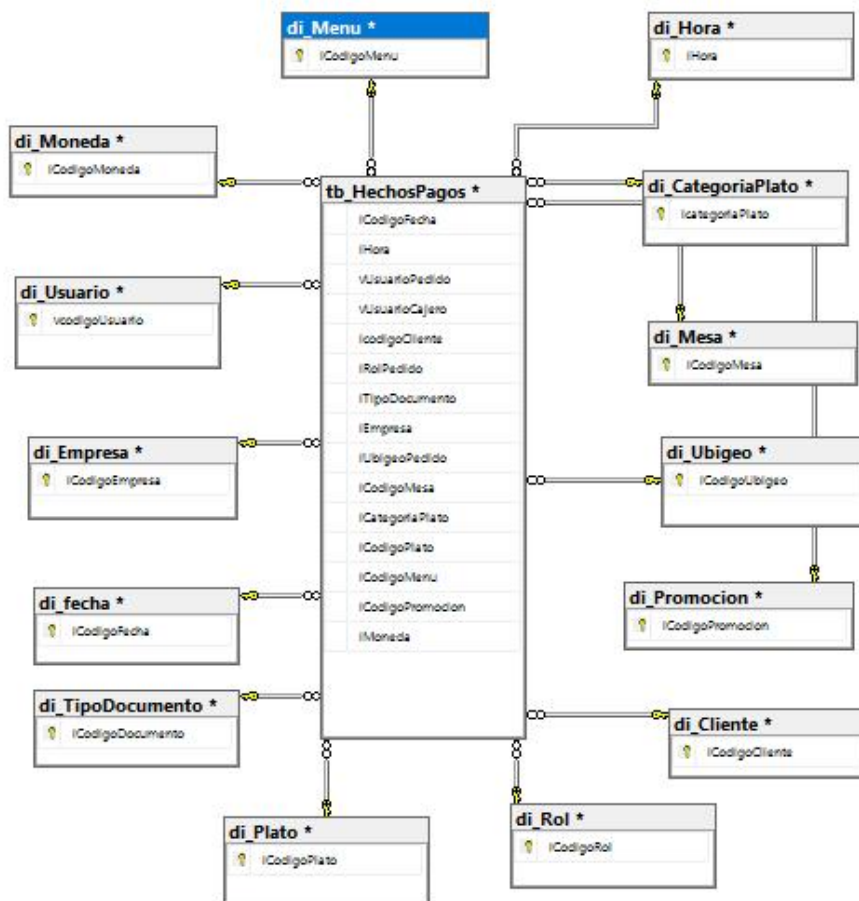


Figura 30: Modelo relacional del data warehouse de la tabla de hechos de pagos.

Fuente: Creación propia

- **KB_Restaurante**

Base de datos del conocimiento, que se actualizará de forma diaria y tomará decisiones de acuerdo al día a día del negocio.

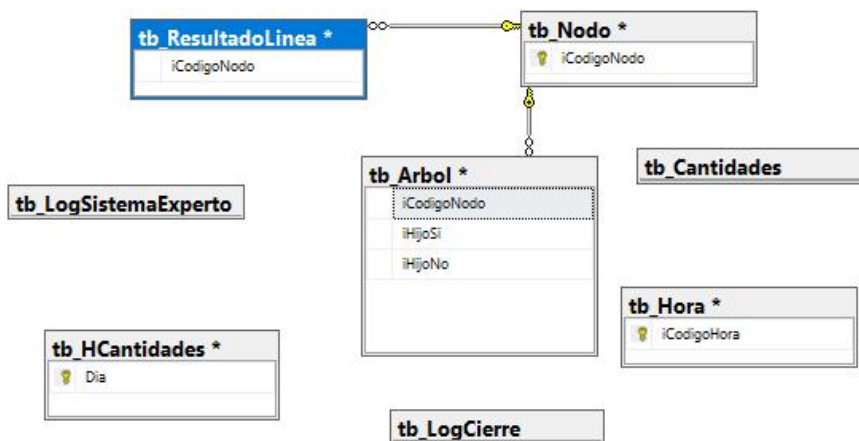


Figura 31: Modelo relacional de la base de datos de conocimiento.

Fuente: Creación propia.

CAPÍTULO IV

DISEÑO DEL SISTEMA EXPERTO

4.1. Documentación técnica de programación

La documentación técnica servirá para el análisis del desarrollo del sistema.

El lenguaje para el desarrollo del sistema es C# y las herramientas para el desarrollo del mismo son las siguientes:

- Visual Studio 2015
- Reporting Services 2016
- Sql Server 2016 Express
- IIS 10

4.2. Mapa del Sistema

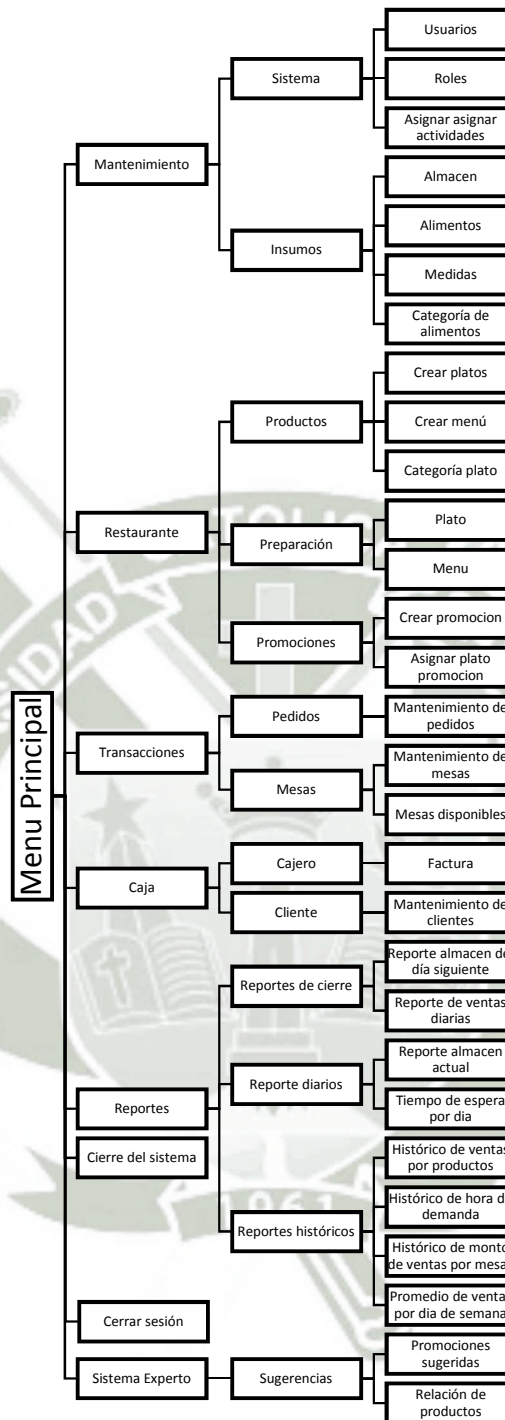


Figura 32: Mapa del sistema.

Fuente: Creación propia.

4.3. Arquitectura de programación

Se usó la arquitectura de programación modelo, vista y controlador para la creación del sistema “Mi Restaurante Experto”.

- **Capa del Modelo**

Es la capa donde se implementa la lógica del negocio, es la responsable de obtener los datos para que puedan ser mostrados en el sistema, además realizan la validación de los datos para que puedan realizar la manipulación de los mismos.

- **Capa de la Vista**

Es la capa donde que permite construir una página web, realizando una petición a la capa de modelo para que pueda presentar los datos en el sistema.

- **Capa del Controlador**

Es la capa responsable de la gestión de las peticiones de los usuarios, es el responsable de realizar las validaciones de acuerdo para la capa del modelo y de la vista.

4.4. Aspectos relevantes de la programación del sistema

4.4.1. Desarrollo del sistema transaccional

El sistema transaccional se agrupó en tres proyectos para la creación del mismo; cada proyecto tiene la funcionalidad de la creación de las capas de la arquitectura de software modelo, vista y controlar.

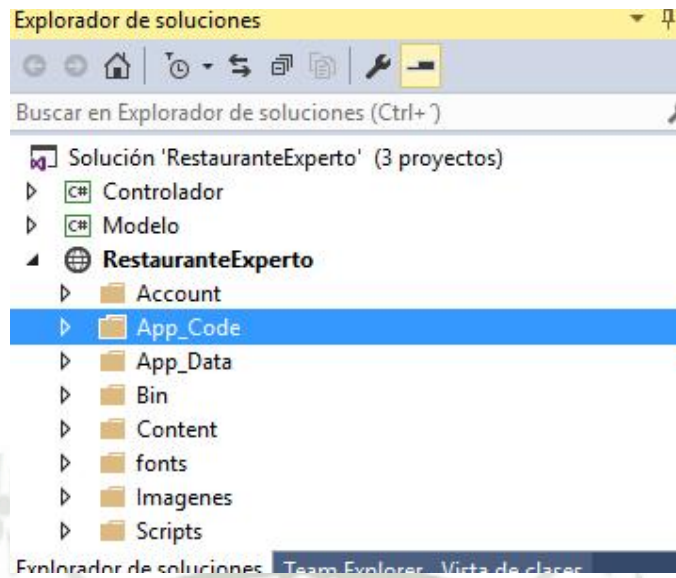


Figura 33: Modelo de capas del sistema.

Fuente: Creación propia.

4.4.2. Proyecto Modelo

Es el encargado de realizar las conexiones a la base de datos transaccional del sistema.

En la figura 34 se observa cómo se realiza el llamado a la conexión de las bases de datos.

```
232 referencias
public class Conexion
{
    public SqlConnection con;
    0 referencias
    public SqlConnection Conectar()
    {
        Abrir();
        return con;
    }
    117 referencias
    public SqlConnection Abrir()
    {
        string Cadena = "";
        Cadena = ConfigurationManager.ConnectionStrings["Transacciones"].ConnectionString;
        SqlConnection con = new SqlConnection(Cadena);
        con.Open();
        return con;
    }
    1 referencia
    public void Cerrar()
    {
        con.Close();
    }
}
```

Figura 34: Conexión a la base de datos.

Fuente: Creación propia.

La conexión hacia la base de datos se realiza mediante un usuario de base de datos el cual se encuentra en el archivo web.config de la aplicación, en ella también se encuentra la conexión hacia la base de datos de conocimiento ya que el sistema realiza una conexión hacia dicha base de datos.

```
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityframework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Cu
  </configSections>
  <connectionStrings>
    <add name="Transacciones" connectionString="Data Source=(local)\local2016C; user id= sa; password=9deJulio1;Initial Catalog=CO_Restaurante
    <add name="Cierre" connectionString="Data Source=(local)\local2016C; user id= sa; password=9deJulio1;Initial Catalog=master;"/>
  </connectionStrings>
  <system.web>
    <authentication mode="None"/>
    <compilation debug="true" targetFramework="4.5.2">
  </compilation>
  <assemblies>
    <add assembly="Microsoft.ReportViewer.WebForms, Version=10.0.0.0, Culture=neutral, PublicKeyToken=803F5F7F11D50A3A"/>
    <add assembly="Microsoft.ReportViewer.Common, Version=10.0.0.0, Culture=neutral, PublicKeyToken=803F5F7F11D50A3A"/>
    <add assembly="Microsoft.Build.Framework, Version=4.0.0.0, Culture=neutral, PublicKeyToken=803F5F7F11D50A3A"/>
    <add assembly="System.Management, Version=4.0.0.0, Culture=neutral, PublicKeyToken=803F5F7F11D50A3A"/>
  </assemblies>
</configuration>
```

Figura 35: Conexiones a la base de datos.

Fuente: Creación propia.

En la figura 35 se observa las cadenas de conexiones que se encuentran en el archivo web.config

4.4.3. Ejemplo de módulos del sistema

El sistema sólo llamará a procedimientos almacenados para la creación de las transacciones que realice este, se utilizó esta forma de programación ya que es la más útil al momento de realizar los cambios en el sistema.

Es decir si realizamos el llenado de una grilla, en la capa de vista se llama a la función llenar grilla y de ahí se llama a una función del proyecto controlador, el cual será el encargo de llamar al procedimiento almacenado encontrado en la base de datos.

En la figura 36, se muestra un ejemplo de cómo poblar una grilla en el sistema.

```

> referencias
public void LlenarGrilla()
{
    Factu objAlimento = new Factu();
    DataTable dt = new DataTable();
    dt = objAlimento.FN_ListaPedido(Int32.Parse(DDL.Mesa.Selected.Value.ToString()));
    GridView1.DataSource = dt;
    GridView1.DataBind();
}
    
```

Figura 36: Grilla del sistema.

Fuente: Creación propia.

En la figura 37, se muestra como una función se llama al procedimiento de la base de datos.

```

public DataTable FN_ListaPedido(int Mesa)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand("SP_ListarPedFactura", con);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Mesa", Mesa);
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    da.Fill(dt);
    return dt;
}
    
```

Figura 37: Llamado a un procedimiento almacenado del sistema.

Fuente: Creación propia.

En la figura 38 se muestra como están creados los procedimientos almacenados en la base de datos.

```

Use [US Restaurante]
GO
/***** Object: StoredProcedure [dbo].[SP_ListarPedFactura]    Script Date: 11/06/2017 21:00:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER Procedure [dbo].[SP_ListarPedFactura]
(
    @Mesa int
)
as
begin
select
count(*) as Cantidad,
pla.NombrePlato,
pla.fCosto,
isnull(ppe.NombrePromocion,'') as NombrePromocion,
cast(count(*)*pla.fCosto as decimal(10,2)) as MontoBase,
cast(count(*)*(case when pro.itipoDescuento=1 then pro.fmonto when pro.itipoDescuento=2 then cast(pro.fPorcentaje*pla.fCosto/100 as decimal(10,2)) e
cast(count(*)*pla.fCosto as decimal(10,2)) -
cast(count(*)*(case when pro.itipoDescuento=1 then pro.fmonto when pro.itipoDescuento=2 then cast(pro.fPorcentaje*pla.fCosto/100 as decimal(10,2)) e
from tb_Hadido pad
inner join tb_Plato pla
on ped.iCodigoPlato=pla.iCodigoPlato and pla.iEstado=1
left join tb_Promocion pro
on pro.iCodigoPlato=pla.iCodigoPlato and pro.iEstado=1
left join tb_Promocion pro
on pro.iCodigoPromocion= app.iCodigoPromocion and pro.iEstado=1
where ped.iEstado=1
and ped.iCodigoMesa=@Mesa
group by
pla.NombrePlato,

```

Figura 38: Ejemplo de procedimiento almacenado en la base de datos del sistema.

Fuente: Creación propia.

CAPITULO V

VALIDACIÓN DE LOS MÓDULOS DEL SISTEMA EXPERTO

5.1. Módulo de Usuarios

El módulo realiza la creación de los usuarios en el sistema, dichos usuario se les tiene que asignar algún rol registrado en el mismo.

Para poder agregar un nuevo usuario el módulo tendrá un botón que lleva por descripción Agregar usuario, el cual en hacer click en dicho botón nos permitirá ingresar los siguientes campos:

- Código de usuario.
- Contraseña.
- Nombre.
- Apellido materno.
- Apellido paterno.
- Teléfono.
- Celular.
- Rol.
- Tipo de documento.
- Número de documento.
- Estado.
- Ubigeo.

Luego de ingresar al usuario, el sistema validará si el código o el documento de identidad ya fueron registrados; en el caso que esto suceda, mostrará una alerta y no se podrá registrar a dicho usuario.

El módulo muestra una grilla de los usuarios que están registrados en el sistema, en ella se contará con dos botones por registro; uno de ellos es el botón editar el cual tiene la funcionabilidad de actualizar los datos del usuario seleccionado; el segundo botón es eliminar el cual tiene la función de modificar el estado del usuario seleccionado a desactivado.

5.1.1. Validación

El sistema validará que no existan dos códigos repetidos, ni dos usuarios que cuenten con el mismo documento de identidad.

Procedimientos usados:

- **SP_Usuario:** Busca el usuario en la base de datos y retorna las características de este, tiene la funcionabilidad de retornar a sólo un usuario o a toda la lista de usuarios.
- **SP_TipoDocumento:** Lista los tipos de documentos del sistema.
- **SP_Rol:** Lista los roles de usuarios.
- **SP_EstadoUsuario:** Indica el estado de usuario buscado.
- **SP_Departamento:** Lista los departamentos registrados en el sistema.
- **SP_Provincia:** Lista las provincias registradas en el sistema de acuerdo al departamento seleccionado.
- **SP_Distrito:** Lista los distritos registrados en el sistema de acuerdo al departamento seleccionado.
- **SP_CodigoUbigeo:** Obtiene el código de ubigeo de acuerdo al departamento, provincia y distrito seleccionado.
- **SP_InsertarUsuario:** Crea al usuario en la base de datos.

- **SP_ModificarUsuario:** Actualiza el usuario y guarda la fecha y el código de usuario que realizó la modificación.
- **SP_EliminarUsuario:** Elimina el usuario seleccionado cambiando el estado a inactivo.

En la figura 39, se muestra el módulo de creación de usuarios luego de realizar click en el botón agregar usuario.

Codigo Usuario	Nombre Usuario	Apellido Paterno	Apellido Materno	Telefono	Celular	Rol	Documento	Departa	Provincia	Distrito	Estado	Editar	Eliminar
aceipa	alison	Ceipa	Yauri	979717979	995959595	Encargado de Ventas	71452402	Arequipa	Arequipa	La Joya	Activo	Editar	Eliminar
cremirez	CARLOS	RAMIREZ	MOSQUERA	221643	99456213	Personal de Atención	23055412	Arequipa	Arequipa	Arequipa	Activo	Editar	Eliminar
ppataco	PLEROY	CHALLOO	BALACC	260514	995959595	Personal de Atención	45106269	Arequipa	Arequipa	Yanahuara	Activo	Editar	Eliminar

Figura 39: Figura del módulo de registro usuarios.

Fuente: Creación propia.

5.2. Módulo de roles

El módulo de roles permite la creación de roles en el sistema. El módulo permitirá el registro del nombre del rol; si el nombre ya existe el sistema emitirá una alerta de que el rol ya ha sido creado; el módulo también contará con una grilla en la cual listará los roles registrados en el sistema; en cada registro mostrará un botón de editar; el cual permitirá modificar el nombre del rol creado.

5.2.1. Validación

El sistema validará que no existan dos roles con el mismo nombre.

5.2.2. Procedimientos usados

- **SP_ListaRol:** Lista los roles registrados en el sistema.
- **SP_ActualizarRol:** Actualiza el nombre del rol.
- **SP_ExisteRol:** Indica si el nombre del rol ingresado ya existe en el sistema.

En la figura 40, se observa el módulo de mantenimiento de roles.



Codigo Rol	Nombre Rol	Fecha Modificación	Editar
5	Encargado de Ventas secundario	11/06/2017	Editar
4	Encargado de Ventas	12/03/2017	Editar
3	Personal de Atención	8/10/2016	Editar
2	Cajero	8/10/2016	Editar
1	Administrador	18/09/2016	Editar

Figura 40: Figura del módulo de mantenimiento de roles.

Fuente: Creación propia.

5.3. Módulo de asignar actividades

El módulo de asignar actividades tiene la funcionalidad de asignar las actividades del sistema a un determinado rol, dicho módulo tendrá dos listados; uno de ellos será de la listas de roles y el otro mostrará las listas de actividades que no pertenezcan al rol seleccionado. En el módulo se mostrará una grilla de las actividades que presente el rol seleccionado,

cada registro mostrará un botón de eliminar el cual permitirá eliminar la actividad del rol seleccionado.

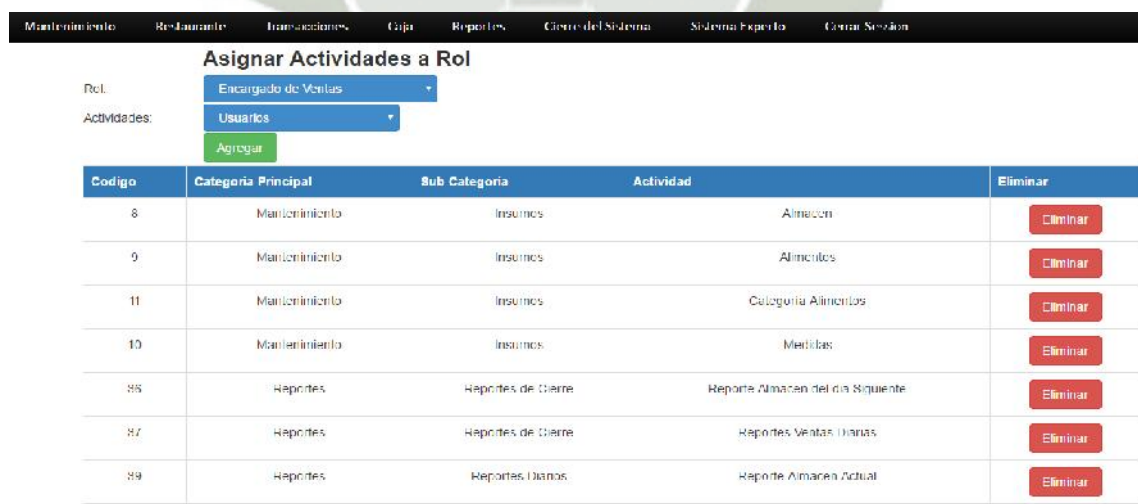
5.3.1. Validación

En el listado de actividades sólo aparecerán las actividades que no fueron asignadas al rol buscado, para que no exista una redundancia en la asignación.

5.3.2. Procedimientos usados

- **SP_CrearActRol:** Crea asignación al rol de la actividad seleccionada.
- **SP_ListaRol:** Lista los roles creados en el sistema.
- **SP_ListarActividad:** Lista las actividades faltantes al rol seleccionado.
- **SP_EliminarRolActi:** Elimina la actividad al rol seleccionado.
- **SP_ListaRolActividad:** Lista las actividades que han sido agregadas al rol seleccionado.

En la figura 41, se observa el módulo de asignar actividades a un rol.



Codigo	Categoria Principal	Sub Categoría	Actividad	Eliminar
8	Mantenimiento	Insumos	Almacen	Eliminar
9	Mantenimiento	Insumos	Alimentos	Eliminar
11	Mantenimiento	Insumos	Categoría Alimentos	Eliminar
10	Mantenimiento	Insumos	Medidas	Eliminar
85	Reportes	Reportes de Cierre	Reporte Almacen del día Siguiente	Eliminar
87	Reportes	Reportes de Cierre	Reportes Ventas Diarias	Eliminar
89	Reportes	Reportes Diarios	Reporte Almacen Actual	Eliminar

Figura 41: Figura del módulo de asignar actividades a un rol.

Fuente: Creación propia.

5.4. Módulo de almacén

El módulo de almacén permite registrar los insumos del restaurante, además permite el registro de tiempo de caducidad de los alimentos. El módulo contará con un listado de los ingredientes registrados en el sistema, a su vez se deberá ingresar los campos de cantidad, cantidad máxima y tiempo de caducidad. El módulo también contará una grilla de los alimentos y un botón editar por cada registro, el cual permitirá actualizar los datos del alimento seleccionado.

5.4.1. Validación

Las cantidad máxima, cantidad y tiempo de caducidad debe ser mayor a 0.

5.4.2. Procedimientos usados

- **SP_ListarAlmacen:** Lista el estado en el almacén del ingrediente seleccionado.
- **SP_ListarIngrediente:** Lista los ingredientes registrados en el sistema.
- **SP_AgregarAlmacen:** Realiza la creación y la actualización del ingrediente al almacén.

En la figura 42, se observa el módulo de inicio del mantenimiento de almacén.

Mantenimiento de Almacén

Ingrediente:
 Cantidad:
 Cantidad Máxima:
 Tiempo de Caducidad:

Código Almacén	Nombre Categoría	Nombre Ingrediente	Cantidad	Monto	Cantidad Máxima	Tiempo Caducidad	Cantidad Mínimo	Editar
3	Vegetales	Ajo	36.55	328.95	50.00	30.00	1.08	<input type="button" value="Editar"/>
4	Vegetales	Arveja	11.70	46.80	12.00	3.00	0.38	<input type="button" value="Editar"/>
5	Cereal	Arroz	39.20	137.20	100.00	60.00	7.47	<input type="button" value="Editar"/>
6	Camarón	Camarón	10.00	300.00	10.00	2.00	5.75	<input type="button" value="Editar"/>
7	Tubérculos	Camote	24.00	72.00	50.00	7.00	2.81	<input type="button" value="Editar"/>
8	Vaca	Carne Molida	3.95	56.30	15.00	10.00	3.68	<input type="button" value="Editar"/>

Figura 42: Figura del módulo de Mantenimiento de almacén.

Fuente: Creación propia.

5.5. Módulo de alimentos

El módulo de alimentos permite la creación de los alimentos o insumos del sistema, en dicho módulo se tiene que ingresar el nombre del alimento, la medida y el precio por medida del alimento.

El módulo permitirá el ingreso de la categoría del alimento, el nombre, la medida y el precio por unidad de medida seleccionada. Luego de realizar click en el botón agregar mostrará una grilla los alimentos registrados en el sistema, realizando un filtro de la categoría seleccionada; a su vez por cada registro mostrará un botón editar que tendrá la funcionalidad de modificar las características del registro seleccionado.

5.5.1. Validación

Valida que no exista un ingrediente con el mismo nombre, a su vez que el monto del costo del ingrediente sea mayor a 0.

5.5.2. Procedimientos usados

- **SP_AgregarIngrediente:** Realiza la creación del ingrediente en el sistema.
- **SP_ListaCategorol:** Lista las categorías de ingrediente.
- **SP_ListarIngre:** Lista los ingredientes del sistema.
- **SP_ValidarIngrediente:** Retorna el indicador si el ingrediente ya se encuentra registrado en la base de datos.
- **SP_ListarMedidas:** Lista las medidas de los ingredientes.
- **SP_ActualizarIngrediente:** Realiza la actualización del ingrediente seleccionado.

En la figura 43, se muestra el módulo de mantenimiento de alimentos.



Codigo Alimento	Nombre Categoría	Medida	Alimento	Costo	Editar
1	Aves	Pieza	Pierna de Pollo	5.00	Editar
2	Aves	Kilogramos	Pechuga de Pollo	12.00	Editar
3	Aves	Kilogramos	Pechuga de Pavo	8.50	Editar
21	Aves	Kilogramos	Gallina	8.50	Editar

Figura 43: Figura del módulo de Mantenimiento de alimentos.

Fuente: Creación propia.

5.6. Módulo de medidas

El módulo de medidas permitirá la creación de las medidas de los alimentos en el sistema, en dicho módulo se debe registrar el nombre y abreviatura de la medida.

En el módulo se deberá ingresar el nombre y la abreviatura de la medida a registrar en el sistema; el sistema validará si dicha medida ya se encuentra registrada en el sistema, en el caso que sea así mostrará una alerta. El módulo contará con una grilla de las medidas registradas en el sistema; a su vez por cada registro mostrará un botón editar, el cual nos podrá modificar las características de la medida seleccionada.

5.6.1. Validación

Valida si existe un nombre de medida ya creado en el sistema.

5.6.2. Procedimientos usados

- **SP_ValidarMedida:** Indicador si la medida ya se encuentra registrada en el sistema.
- **SP_InsertarMedida:** Realiza la creación de la medida.
- **SP_ListarMedidas:** Lista las medidas registradas en el sistema.
- **SP_ActualizarMedida:** Realiza la actualización de la medida.

En la figura 44, se muestra el módulo de mantenimiento de medidas, al inicio de ingresar a dicho módulo.



© 2017 - Mi Restaurante Experto

Figura 44: Figura del módulo de Mantenimiento de medidas.

Fuente: Creación propia.

5.7. Módulo de categorías de ingredientes

El módulo de categorías de ingredientes permite la creación de las categorías de los ingredientes, en dicho módulo se debe registrar el nombre de la categoría del ingrediente.

El módulo permite el ingreso del nombre de categoría, el sistema mostrará una alerta si la categoría que deseamos ingresar ya se encuentra registrada en el sistema, el módulo cuenta con una grilla que muestra las medidas registradas en el sistema, por cada registro de la grilla mostrará un botón editar, el cual tendrá las función de modificar el nombre de la categoría seleccionada.

5.7.1. Validación

El nombre de la categoría no puede estar registrada en el sistema.

5.7.2. Procedimientos usados

- **SP_AgregarCategorial:** Crea la categoría del ingrediente.

- **SP_ActualizarCategorial:** Actualiza la categoría del ingrediente.
- **SP_ValidarCategorial:** Indicador si la categoría ya ha sido creada en el sistema.
- **SP_ListaCategorial:** Listado de las categorías registradas en el sistema.

En la figura 45, se muestra el módulo de mantenimiento de categoría cliente.

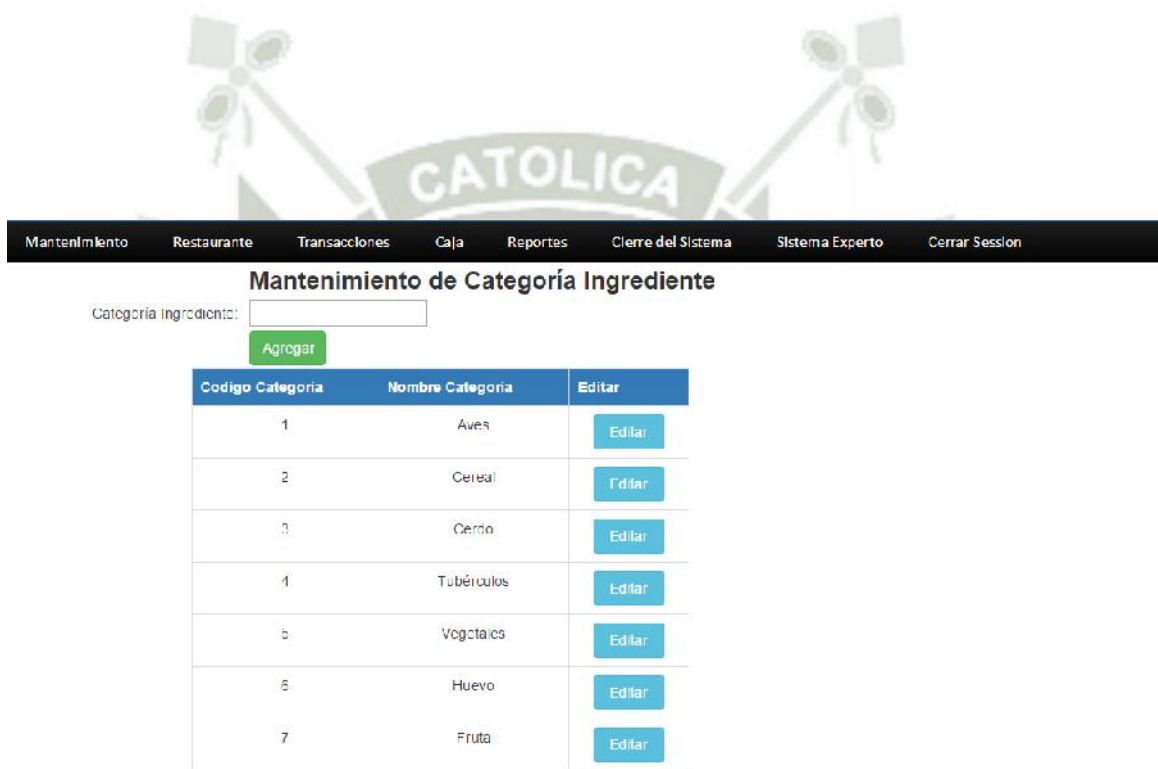


Figura 45: Figura del módulo de Mantenimiento de categoría de ingrediente.

Fuente: Creación propia.

5.8. Módulo de platos

El módulo de platos permite el registro de los platos en el sistema, en dicho módulo se debe registrar la categoría del plato, el nombre del plato, la moneda y el valor monetario del mismo.

El módulo mostrará un listado de las categorías de los platos, en él se ingresará el nombre del plato, se seleccionará la moneda y el costo del plato; el módulo también contará con una grilla de los productos registrados en el sistemas realizando un filtro de la categoría seleccionada; a su vez cada registro de la grilla contará con dos botones; uno de ellos será el botón editar el cual permitirá modificar las características del plato del seleccionado, y otro un botón eliminar el cual desactivará el plato seleccionado.

5.8.1. Validación

El nombre del plato no debe estar registrado en el sistema, además el monto del costo debe ser mayor a 0.

5.8.2. Procedimientos usados

- **SP_CategoriaP:** Muestra la lista de las categorías de los platos.
- **SP_Moneda:** Muestra el listado de los tipos de moneda.
- **SP_AgregarPato:** Agrega el plato al sistema.
- **SP_CodCategoriaPlato:** Obtiene el código de la categoría del plato.
- **SP_CodigoMoneda:** Obtiene el código de la moneda.
- **SP_ActualizarPlato:** Actualización de los atributos del plato.
- **SP_EliminarPlato:** Cambia el estado del plato a desactivado.
- **SP_ValidarPlato:** Indicador si el plato ha sido creado.

En la figura 46, se muestra el módulo de mantenimiento de platos, luego de realizar un filtro de la categoría de platos, por ello, la grilla sólo muestra los platos de dicha categoría.

Mantenimiento de Platos

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Categoría Plato:

Nombre de Plato:

Moneda:

Monto:

Codigo Plato	Categoria	Plato	Moneda	Costo	Editar	Eliminar
3	Aves	Pechuga Chicaclada	S/	25.00	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
4	Avcs	Milanesa de Pollo	S/	26.00	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
5	Aves	Pollo Chocado	S/	25.00	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
6	Aves	Chicharron de Pollo	S/	27.00	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
7	Avcs	Pollo a la Hawaiana	S/	30.00	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
8	Aves	Pollo con Champiñones	S/	27.00	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

Figura 46: Figura del módulo de Mantenimiento de platos.

Fuente: Creación propia.

5.9. Módulo de Menú

El módulo permite la creación de los menús del restaurante, en dicho módulo se debe registrar el nombre del menú, la moneda y el monto del menú.

En el módulo se deberá ingresar el nombre del menú, se seleccionará la moneda y se ingresará el monto del plato seleccionado; el módulo tendrá una grilla que listará los menús registrados en el sistema, por cada registro la grilla mostrará dos botones uno de ellos será el botón editar, el cual permitirá modificar las características del menú seleccionado y un botón eliminar el cual permitirá desactivar el menú seleccionado.

5.9.1. Validación

El nombre del menú no debe estar registrado en el sistema, además el monto de costo debe ser mayor a 0.

5.9.2. Procedimientos usados

- **SP_ListaMenu:** Muestra la lista de menú.
- **SP_Moneda:** Muestra el listado de monedas.
- **SP_CrearMenu:** Registra el menú en el sistema.
- **SP_CodigoMoneda:** Obtiene el código de la moneda.
- **SP_ContarMenu:** Valida si el menú ha sido creado.
- **SP_ActualizarMenu:** Actualiza los atributos del menú.
- **SP_EliminarMenu:** Cambia el estado a inactivo del menú.

En la figura 47, se muestra el módulo de mantenimiento de menús.



Codigo Menú	Menú	Moneda	Costo	Editar	Eliminar
2	Menu infantil	S/.	20.00	Editar	Eliminar
3	Menú 1	S/.	25.00	Editar	Eliminar

Figura 47: Figura del módulo de Mantenimiento de menús.

Fuente: Creación propia.

5.10. Módulo de categorías de platos

El módulo permite la creación de la categoría del plato, en dicho módulo se debe registrar sólo el nombre de la categoría del plato.

En el módulo se deberá ingresar el nombre de la categoría a crear, el sistema validará si la categoría ya ha sido registrada en el sistema; el módulo mostrará una grilla en la cual se visualizará las categorías de platos de platos registrados en el sistema, cada registro contará con dos botones; uno de ellos un botón editar el cual permitirá modificar las características de la categoría de plato seleccionada y otro botón eliminar el cual tendrá la función de desactivar la categoría seleccionada.

5.10.1. Validación

Valida que el nombre de la categoría del producto no se encuentre registrado en el sistema

5.10.2. Procedimientos usados

- **SP_AgregarCategoriaPlato:** Crea la categoría plato en el sistema.
- **SP_ContarCategoriaP:** Indicador que la categoría del plato no este registrado en el sistema.
- **SP_ActualizarCategoriaP:** Actualiza los atributos de la categoría del plato.

En la figura 48, se muestra el módulo de mantenimiento de categoría de platos.



Figura 48: Figura del módulo de Mantenimiento de categoría de platos.

Fuente: Creación propia.

5.11. Módulo de preparación de platos

En el módulo de preparación de platos permite la asignación de ingredientes para la preparación del plato seleccionado. Se debe ingresar la cantidad del ingrediente usado y calculará el costo de producción del plato seleccionado.

En el módulo se tendrá un botón de buscar en el cual se deberá ingresar la categoría y el nombre del palto seleccionado, luego de hacer click en dicho botón nos permitirá ingresar los siguientes campos:

- Categoría del ingrediente
- Nombre del ingrediente
- Cantidad del ingrediente usado para la preparación del plato

El módulo tendrá la función de calcular el costo de producción de dicho plato, el módulo mostrará una grilla, en la cual mostrará los ingredientes del plato seleccionado, en cada registro de dicha grilla mostrará dos botones; uno de ellos un botón editar el cual podrá modificar las características del plato seleccionado y otro un botón eliminar el cual eliminará el ingrediente seleccionado del plato buscado.

5.11.1. Validación

La cantidad al momento de realizar la actualización no debe estar en blanco, además si el plato ya cuenta con el ingrediente agregado.

5.11.2. Procedimientos usados

- **SP_CategoriaP:** Muestra la categoría de los platos.
- **SP_NombrePlato:** Muestra los nombres de los platos.
- **SP_ListaCategor:** Lista las categorías de los ingredientes.
- **SP_ListarPlatng:** Lista los ingredientes de un determinado plato.
- **SP_Medidal:** Lista las medidas de los ingredientes.
- **SP_EliminarPlatol:** Elimina el ingrediente del plato seleccionado.
- **SP_ActualizarPlatol:** Actualiza el ingrediente del plato seleccionado.
- **SP_AgregarPlatol:** Agrega el ingrediente al plato seleccionado.
- **SP_ValidarPlatol:** Indicador de ingrediente ya pertenece al plato seleccionado.

En la figura 49, se muestra el módulo de ingredientes por plato, luego de realizar el filtro de plato.

Ingredientes por Plato

Categoría de Plato: Aves
 Categoría de Ingrediente: Aves
 Cantidad: Kg
 Nombre de Plato: Chicharrón de Pollo
 Ingrediente: Gallina
 Costo: 7.41

Codigo	Nombre Ingrediente	Medida	Cantidad	Costo	Editar	Eliminar
20	Pierna de Pollo	Pz	1.00	5.30	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
29	Arroz	Kg	0.25	0.88	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
30	Pepe	Kg	0.25	0.63	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
31	Mandioca	Kg	0.10	0.60	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

Figura 49: Figura del módulo de asignar ingredientes a un plato.

Fuente: Creación propia.

5.12. Módulo de asignación de platos a un menú

El módulo de asignación de platos a un menú permite la asignación de un determinado plato al menú seleccionado.

En el módulo se deberá seleccionar el menú al que se desee agregar un plato, además se deberá seleccionar la categoría y el plato que se agregará al menú elegido, el módulo mostrará una grilla de los platos seleccionados al menú seleccionado, por cada registro se mostrará un botón eliminar el que permitirá borrar el plato seleccionado del menú deseado.

5.12.1. Procedimientos usados

- **SP_NombreMenu:** Lista los menús registrados en el sistema.
- **SP_CategoriaP:** Lista la categorías de platos.
- **SP_NombrePlato:** Lista los platos de acuerdo a la categoría elegida.

- **SP_InsertarMenuPlato:** Registra el plato al menú seleccionado.
- **ListaMenuPlato:** Lista los platos registrados al menú seleccionado.
- **SP_EliminarMenuPlato:** Elimina el plato del menú seleccionado.

En la figura 50, se muestra el módulo de asignar los platos al menú.



Codigo	Categoría	Plato	Eliminar
9	Bebidas	Caseosa de 1 Litro	Eliminar
10	Aves	Chicharron de Pollo	Eliminar

Figura 50: Figura del módulo de asignar platos a un menú.

Fuente: Creación propia.

5.13. Módulo de promociones

El módulo de promociones permite la creación de promociones en el sistema, para ello se debe ingresar el tipo de descuento, el monto del descuento que mantendrá la promoción, además del tiempo de inicio y fin de la misma.

En el módulo se deberá ingresar las siguientes características de la promoción a crear:

- Nombre de la promoción.
- Tipo de descuento, el cual podrá ser por monto y por porcentaje.
- Monto.
- Fecha de inicio.

- Fecha fin.

El módulo contará con una grilla del listado de las promociones, la cual que por cada registro mostrará dos botones, uno de ellos un botón editar el cual permitirá modificar las características de la promoción y un botón eliminar el cual permitirá eliminar la promoción del sistema.

5.13.1. Validación

El nombre de la promoción no debe estar registrado, además el monto debe ser mayor a 0.

5.13.2. Procedimientos usados

- **SP_TipoPromo:** Lista los tipos de promoción.
- **SP_CrearPromocion:** Registra la promoción en el sistema.
- **SP_ActualizarPromocion:** Actualiza la promoción en el sistema.
- **SP_ValidarNombrePromocion:** Valida si el nombre de la promoción ingresado ya existe en el sistema.
- **SP_EliminarPromocion:** Elimina la promoción en el sistema.
- **SP_ListarPromocion:** Muestra el listado de promociones.

En la figura 51, se muestra el módulo de crear promociones.

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Crear Promociones

Nombre Promoción:

Tipo Descuento: **Monto**

Monto:

Fecha Inicio:

Fecha Fin:

Codigo	Nombre Promoción	Tipo	Monto/Porcentaje	Fecha Inicio	Fecha Fin	Editar	Eliminar
22	Carveza Cuzqueña 2017-04-28T19:00:00	Porcentaje	20.00	23/04/2017 7:00:00 p. m.	23/04/2017 8:59:59 p. m.	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
23	Carveza Pisen 2017-04-23 20:00:00	Porcentaje	20.00	23/04/2017 8:00:00 p. m.	23/04/2017 8:59:59 p. m.	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
24	Piqueo de Pollo 2017-04-25 12:00:00	Porcentaje	20.00	25/04/2017 9:00:00 p. m.	25/04/2017 10:59:59 p. m.	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
25	Recoto Relleno 2017-04-30T12:00:00	Porcentaje	20.00	30/04/2017 12:00:00 p. m.	30/04/2017 12:59:59 p. m.	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

Figura 51: Figura del módulo de creación de promociones.

Fuente: Creación propia.

5.14. Módulo de asignación de platos a promociones

El módulo de asignación de platos a promoción permitirá la asignación de un determinado plato a la promoción seleccionada.

El módulo mostrará dos listados, uno de ellos será las promociones y el otro un listado de los productos en el sistema, además mostrará una grilla de los productos asignados a la promoción seleccionada, por cada registro la grilla mostrará un botón eliminar el cual permitirá eliminar el producto de la promoción seleccionada.

5.14.1. Validación

No se puede registrar una promoción en la cual el plato ya se tenga registrada en el sistema.

5.14.2. Procedimientos usados

- **SP_ListadePlatoMenu:** Lista los platos y menús registrados en el sistema.

- **SP_ListaPromocion:** Lista las promociones registradas en el sistema.
- **SP_AgregarPromoPlato:** Agrega el plato seleccionado a la promoción seleccionada.
- **SP_ListaPlatoPromocion:** Lista los platos relacionados a la promoción seleccionada.
- **SP_EliminarPlatoPromocion:** Elimina el plato a la promoción seleccionada.
- **SP_ValidarPromoPlato:** Indicador si el plato ya cuenta con la promoción seleccionada.

En la figura 52, se muestra el módulo de asignar platos a promoción, luego de realizar el filtro de promoción.



Codigo	Nombre Plato	Eliminar
29	Cerveza Cuzqueña	Eliminar

Figura 52: Figura del módulo de asignar una promoción a un plato.

Fuente: Creación propia.

5.15. Módulo de mantenimiento de pedidos

El módulo de mantenimiento de pedidos permite la creación de la comanda en el sistema, en dicho módulo se debe seleccionar la mesa y se elegirá los platos que desee el cliente, en ella se deberá ingresar la cantidad de los platos y si desea se agregará una observación del pedido.

En el módulo se deberá ingresar los siguientes campos:

- Seleccionar una mesa de la lista de mesas registradas en el sistema.
- Categoría de platos.
- Plato.
- Cantidad.
- Observación

El módulo mostrará una grilla que mostrará los pedidos de acuerdo a la mesa seleccionada, en ella mostrará un botón eliminar por cada registro de la grilla el cual permitirá eliminar el pedido de la mesa seleccionada.

5.15.1. Validación

El plato elegido debe tener los insumos en stock, además la cantidad por pedido debe ser mayor a 0.

5.15.2. Procedimientos usados

- **SP_EliminarPedido:** Elimina el plato registrado en el pedido.
- **SP_CodigosPedido:** Obtiene el código del pedido.
- **SPListaMesa:** Lista las mesas registradas en el sistema.
- **SP_ListaCatePedido:** Lista las categorías de los platos.
- **SP_ListaPlatoPedido:** Lista los platos de acuerdo a la categoría seleccionada.
- **SP_ValidarPedido:** Valida si cuenta con stock de insumos del plato seleccionado.
- **SP_ActualizarPedido:** Actualiza el pedido seleccionado.

En la figura 53, se muestra el módulo de pedidos, luego de realizar el filtro por la mesa seleccionada.



Codigo Pedido	Categoria Plato	Plato	Cantidad	Observacion	Eliminar
58311	Aves	Chicharron de Pollo	1	Con arroz y papas	Eliminar

Figura 53: Figura del módulo de mantenimiento de pedidos.

Fuente: Creación propia.

5.16. Módulo de mantenimiento de mesas

El módulo de mantenimiento de mesas permitirá la creación de las mesas en el sistema.

En el módulo permitirá ingresar el listado de las mesas, además el módulo tendrá una grilla de las mesas registradas en el sistema, por cada registro de la grilla tendrá un botón eliminar.

5.16.1. Validación

No se pueden tener 2 mesas con el mismo nombre.

5.16.2. Procedimientos usados

- **SP_ValidarNombreMesa:** Valida si el nombre ingresado de la mesa ya existe en el sistema.
- **SP_EliminarMesa:** Elimina la mesa del sistema.
- **SP_ListarMesa:** Lista las mesas registradas en el sistema.

En la figura 54, se muestra el módulo de mantenimientos de mesas.



Figura 54: Figura del módulo de mantenimiento de mesas.

Fuente: Creación propia.

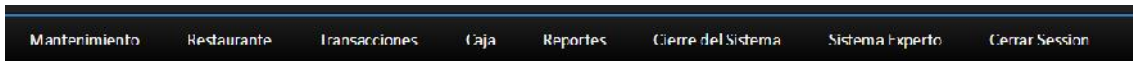
5.17. Módulo de mesas disponibles

El módulo de mesas disponibles mostrará un listado del estado de las mesas del restaurante.

5.17.1. Procedimientos usados

- **SP_ListarEstadoMesa:** Lista el estado de las mesas.

En la figura 55, se muestra el módulo de mesas disponibles.



Mesas Disponibles

Codigo	Nombre de Mesa	Estado de Mesa
2	Mesa 1	Mesa Ocupada
3	Mesa 2	Mesa Libre
4	Mesa 3	Mesa Libre
5	Mesa 4	Mesa Libre
6	Mesa 5	Mesa Libre
7	Mesa 6	Mesa Libre
8	Mesa 7	Mesa Libre

Figura 55: Figura del módulo del reporte de mesas disponibles.

Fuente: Creación propia.

5.18. Módulo de factura

Es el módulo de factura realizará la cancelación de los pedidos de acuerdo a la mesa seleccionada.

En el módulo se deberá seleccionar una mesa, que listará en una grilla los pedidos de la mesa seleccionada, además el módulo mostrará la fecha del sistema, y permitirá el ingreso del DNI del cliente que tendrá un botón buscar que obtendrá el nombre del cliente. Cuando se realice click en el botón aceptar el estado de la mesa cambiará a libre y se podrá registrar nuevos pedidos.

5.18.1. Validación

Valida si la mesa seleccionada cuenta con pedidos, además que se ingrese un cliente para la realización de la compra.

5.18.2. Procedimientos usados

- **SP_CrearFactura:** Se realiza el pago del pedido.
- **SPListaMesa:** Lista las mesas registradas en el sistema.
- **SP_ListarPedFactura:** Lista los pedidos de la mesa seleccionada.
- **SP_NombreCliente:** Obtiene el nombre del cliente.
- **SP_MontosFinales:** Obtiene los montos de la factura.

En la figura 56, se muestra el módulo de facturación de pedidos, luego de realizar el filtro de mesa seleccionada.



Cantidad	Plato	Monto C/U	Promoción	Monto Base	Monto Descuento	Monto Final
1	Chicharrón de Pollo	27.00		27.00	0.00	27.00

Monto Liruto: 20.50
IGV: 3.50
Monto Final: 27.00

Aceptar

Figura 56: Figura del módulo de facturación de pedidos.

Fuente: Creación propia.

5.19. Módulo de mantenimiento de clientes

En el módulo de mantenimiento de clientes se podrá registrar a los clientes, se debe ingresar como campo único el número de documento, además deberá registrar el nombre o razón social del cliente.

En el módulo se deberá ingresar los siguientes campos:

- Nombre o razón social.
- Apellido paterno.
- Apellido materno.
- Tipo de documento.
- Número de documento.
- Fecha de nacimiento.

El módulo tendrá un botón buscar, para ello se deberá ingresar el documento del cliente y el sistema lo buscará en el sistema retornando las características de este.

5.19.1. Validación

El número de documento no puede estar registrado en el sistema para poder realizar la creación del mismo, además no puede estar en blanco.

5.19.2. Procedimientos usados.

- **SP_BuscarCliente:** Realiza la búsqueda del cliente con el ingreso del documento de identidad.
- **SP_AgregarCliente:** Registra al nuevo cliente en el sistema, además tiene la función de actualización.

En la figura 57, se muestra el módulo de mantenimiento de clientes.



Figura 57: Figura del módulo de mantenimiento de clientes.

Fuente: Creación propia.

5.20. Módulo de reporte de almacén del día siguiente

El módulo de reporte de almacén del día siguiente realizará un cálculo predictivo de acuerdo al histórico de ventas para poder calcular la cantidad de insumos necesarios que se utilizarán el día siguiente; dicho módulo llama al reporte alojado en el servidor de reportes con el nombre ReportAlmacenSiguiete.

El módulo mostrará el reporte el cual podrá ser exportado en el formato pdf y Excel.

En la figura 58, se muestra el módulo del reporte del almacén promedio del día siguiente.

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesion

Almacén Promedio del Día Siguiente

14 3 L of 2 7 3 11 End | Next



Fecha: 18/06/2017

Ingrediente	Cantidad Promedio	Cantidad Max	Cantidad Min
Ajo	0,74	1,30	0,20
Albaja	0,25	0,60	0,10
Arroz	5,05	8,35	2,25
Camaron	3,01	7,20	1,00
Camote	2,07	4,00	0,25
Carne Molina	2,54	5,20	0,60
Cebolla	6,51	10,00	3,95
Cerveza Arequipeña	7,63	11,00	2,00
Champiñon	0,51	1,60	0,20
Chorda	3,90	8,00	1,00

Figura 58: Figura del reporte del almacén promedio del día siguiente.

Fuente: Creación propia.

5.21. Módulo de reporte de ventas diarias

El reporte obtendrá la cantidad de los productos vendidos en el día, dicho reporte se encuentra alojado en el servidor de reportes con el nombre ReportVentasDiarias.

El módulo mostrará el reporte el cual podrá ser exportado en el formato pdf y Excel.

En la figura 59, se muestra el módulo del reporte de las ventas del día.

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Reporte Ventas del Día

MI RESTAURANTE EXPERTO

Fecha : 18/06/2017

Plato	Cantidad Vendida	Monto de Ventas
Arroz de Camarón	3	\$/135,00
Asado Mechado con Pure	5	\$/125,00
Caucho de Camarón	2	\$/20,00
Cerveza Arequipaña	11	\$/11,00
Cerveza Cuzqueña	0	\$/50,00
Cerveza Pilsen	5	\$/47,00
Chicharrón de Chancha	3	\$/77,00
Chicharrón de Pollo	2	\$/54,00
Chuleta de Chancha	3	\$/75,00
Chupe de Camarones	1	\$/45,00

Figura 59: Figura del reporte de las ventas del día.

Fuente: Creación propia.

5.2.2. Módulo de reporte de almacén actual

El reporte obtendrá la cantidad de insumos registrados en el almacén, dicho reporte se encuentra alojado en el servidor de reportes con el nombre ReportAlmacenActual.

El módulo mostrará el reporte el cual podrá ser exportado en el formato pdf y Excel.

En la figura 60, se muestra el módulo del reporte del almacén actual.



Figura 60: Figura del reporte del almacén actual.

Fuente: Creación propia.

5.23. Módulo de reporte de tiempo de espera por día

El reporte analizará la data histórica para poder calcular el tiempo de estadía de un cliente en el restaurante, este análisis se determinará de acuerdo al día de la semana, dicho reporte se encuentra alojado en el servidor de reportes con el nombre ReportTiempoEspera.

El módulo mostrará el reporte el cual podrá ser exportado en el formato pdf y Excel.

En la figura 61, se muestra el módulo del reporte del tiempo de espera por día de semana.



Figura 61: Figura del reporte del tiempo de espera por día de semana.

Fuente: Creación propia.

5.24. Módulo de reporte de histórico de ventas por producto

El reporte analizará la data histórica para poder calcular la cantidad de ventas de un determinado producto, dicho reporte se analizará de acuerdo a los meses, dicho reporte se encuentra alojado en el servidor de reportes con el nombre Report1.

Para la obtención del reporte se deberá seleccionar un producto, en el que el reporte mostrará el histórico de ventas del producto seleccionado; dicho reporte podrá ser exportado en el formato pdf y Excel.

En la figura 62, se muestra el módulo del reporte del histórico de ventas por producto.



Figura 62: Figura del reporte del histórico de ventas por producto.

Fuente: Creación propia.

5.25. Módulo de reporte de hora de demanda

El reporte analizará la data histórica para poder calcular las horas de más demanda en el restaurante, para el análisis se realizará filtrando el día de la semana, dicho reporte se encuentra alojado en el servidor de reportes con el nombre ReportHoraDemanda.

Para la obtención del reporte se deberá seleccionar el día de semana, el cual el reporte mostrará el histórico de la hora de demanda del día seleccionado, dicho reporte podrá ser exportado en el formato pdf y Excel.

En la figura 63, se muestra el módulo del reporte del histórico de hora de demanda.



Figura 63: Figura del reporte del histórico de hora de demanda.

Fuente: Creación propia.

5.26. Módulo de reporte de histórico de ventas por meses

El reporte analizará la data histórica para poder calcular la cantidad de ventas en los últimos 12 meses, dicho reporte se encuentra alojado en el servidor de reportes con el nombre ReportVentas.

El módulo mostrará el reporte el cual podrá ser exportado en el formato pdf y Excel.

En la figura 64, se muestra el módulo de del reporte del histórico de monto de ventas por meses.



Figura 64: Figura del reporte del histórico de monto de ventas por meses.

Fuente: Creación propia.

5.27. Módulo de reporte de histórico del promedio de ventas por día de semana

El reporte analizará la data histórica para poder calcular el promedio de las ventas en un determinado día de la semana, dicho reporte se encuentra alojado en el servidor de reportes con el nombre ReportVentasDiaria.

El módulo mostrará el reporte, el cual podrá ser exportado en el formato pdf y Excel.

En la figura 65, se muestra el módulo del reporte del promedio de ventas por día de semana.



Figura 65: Figura del reporte del promedio de ventas por día de semana.

Fuente: Creación propia.

5.28. Módulo de promociones sugeridas

El módulo de promociones sugeridas realizará un cálculo interno de la base de conocimientos para la obtención de las promociones sugeridas.

El reporte mostrará una grilla de las cinco promociones que según el análisis tendrá un mayor impacto en las ventas.

5.28.1. Procedimientos usados

- **SP_ReglaPromocion:** Lista las 5 promociones sugeridas del sistema, esta información se obtiene de la base de conocimientos del sistema.

En la figura 66, se muestra el módulo del reporte de promociones sugeridas.



Nombre de Plato	Precio Actual	Costo de Producción	Ganancia Actual	Descuento %	Costo Nuevo Plato	% Nueva Ganancia
Pollo a la Iltawana	30,00	10,10	190,15 %	30	21,00	107,31 %
Pollo con Champiñones	27,00	11,15	142,15 %	30	18,50	69,51 %
Pollo a lo Hore	27,00	9,28	190,95 %	30	18,50	103,60 %
Febre de Lomcs	15,00	6,97	116,34 %	30	10,50	52,64 %
Sopa a la Minuta	15,00	4,18	234,82 %	30	10,50	101,30 %

Figura 66: Figura del reporte de promociones sugeridas.

Fuente: Creación propia.

5.29. Módulo de relación de productos

El reporte obtiene la información de la base de conocimientos del sistema, el cual realiza los cálculos para poder determinar la relación de las ventas de un producto en un determinado día de la semana, el resultado nos dará una matriz de relación del producto seleccionado, con el resto. Dicho módulo llama al reporte alojado en el servidor de reportes con el nombre ReportProductoPreferente.

Para la obtención del reporte se deberá ingresar el día de la semana y el plato seleccionado, el módulo mostrará el reporte el cual podrá ser exportado en el formato pdf y Excel.

En la figura 67, nos muestra el módulo del reporte de relación de productos.



Figura 67: Figura del reporte de relación de productos.

Fuente: Creación propia.

5.30. Módulo de hechos automáticos

Es el módulo que permite observar las actividades realizadas por el sistema experto.

El módulo mostrará una grilla, en la cual mostrará las acciones que tomó el sistema experto.

En la figura 68, muestra el módulo del reporte de hechos automáticos.



Código Ejecución	Descripción Ejecución	Fecha De Creación
0	Hocoto Helleno 2017-04-30 11:23:00:00	2017/04/30 12:25
1	Piqueo de Pollo 2017-04-25 12:10:00:00	2017/04/25 09:49

© 2017 - MII Restaurante Experto

Figura 68: Figura del reporte de hechos automáticos.

Fuente: Creación propia.

5.31. Población del data warehouse y cierre del sistema

La población del data warehouse se realiza en el cierre del sistema, la población consiste en poblar las dimensiones del data warehouse y la base de hechos (ver Anexo B).

5.32. Sistema de reportería

Se utilizará el sistema de reportería del reporting services, el cual se encontrará en el servidor de la aplicación. En la figura 69 se puede observar la estructura web del reporting services, usado para el desarrollo de los reportes en el sistema:

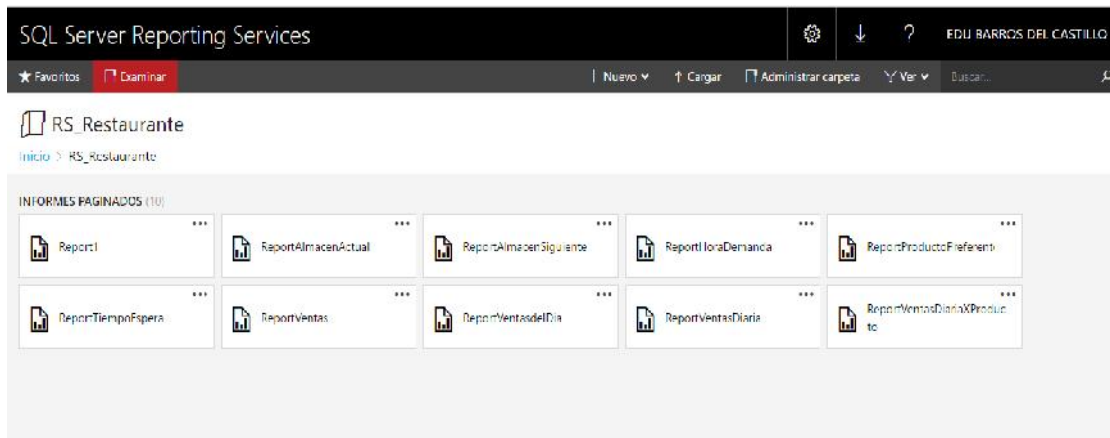


Figura 69: Figura del sistema de reportería.

Fuente: Creación propia.

5.33. Árbol de decisión del sistema experto

Se analizó las decisiones programadas que tiene que realizar el sistema experto, en conjunto con las personas involucradas en el proceso de ventas del restaurante, llegando a la conclusión que el sistema se basaría en las reglas de acuerdo a las circunstancias que se presentarán en el día.

En la figura 70, se muestra el diseño del árbol de decisión luego del análisis por los expertos del proceso de ventas.

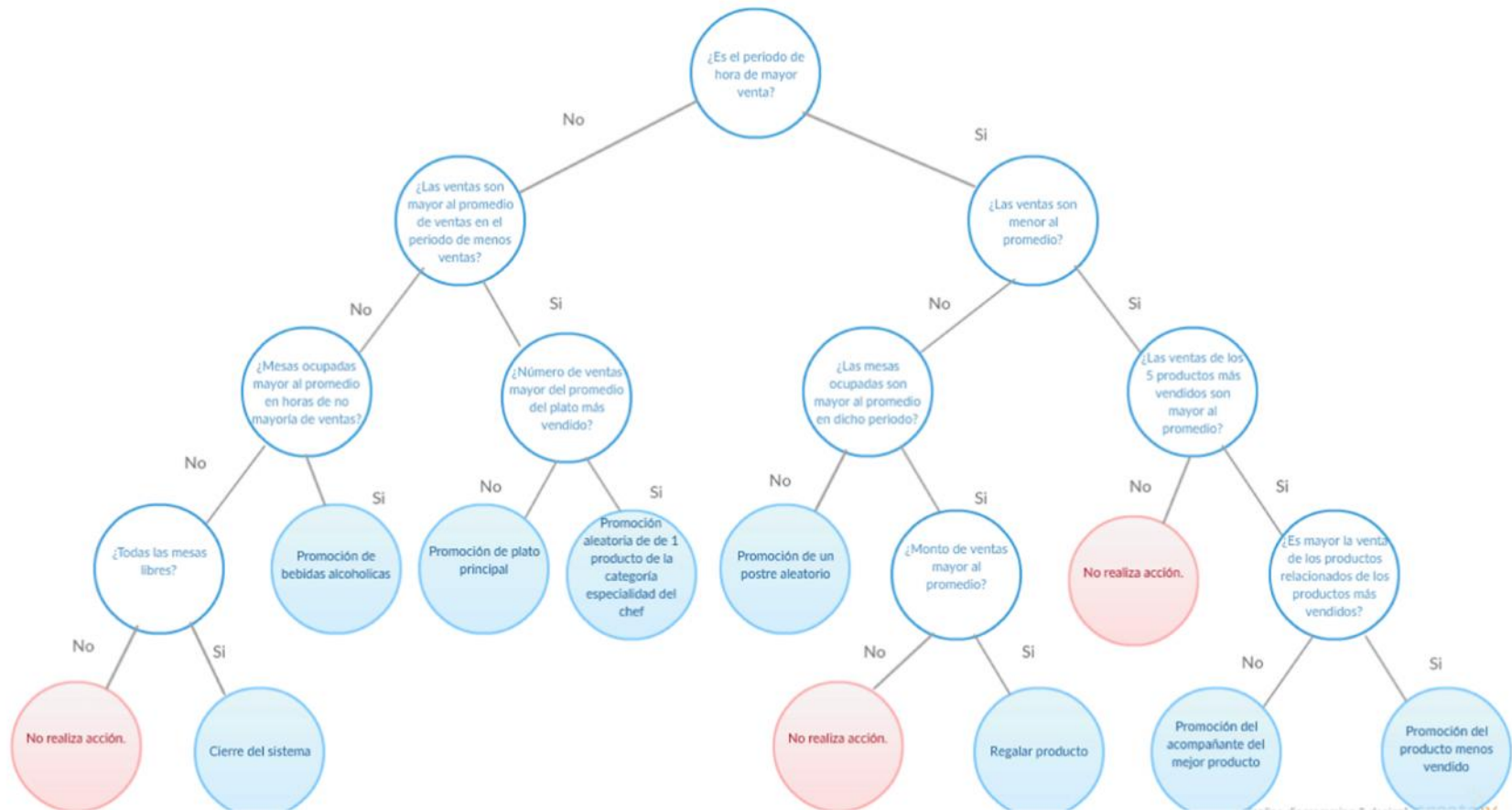


Figura 70: Figura de la funcionalidad del árbol del sistema experto.

Fuente: Creación propia.

Para poder responder las preguntas del árbol de decisión se creó un procedimiento el cuál se ejecutará cada 2 horas por medio de la creación de una tarea programada.

El procedimiento de acuerdo a las circunstancias tomará decisiones en el día a día del restaurante. (Anexo B).

5.34. Programación del Job

Para una programación automatizada se creó una tarea programada, el cuál ejecutará el proceso del cálculo del árbol de decisión.

En la figura 71, se observa el programador de tareas con la función de un programa que llame al procedimiento el análisis del árbol de decisión del sistema experto.

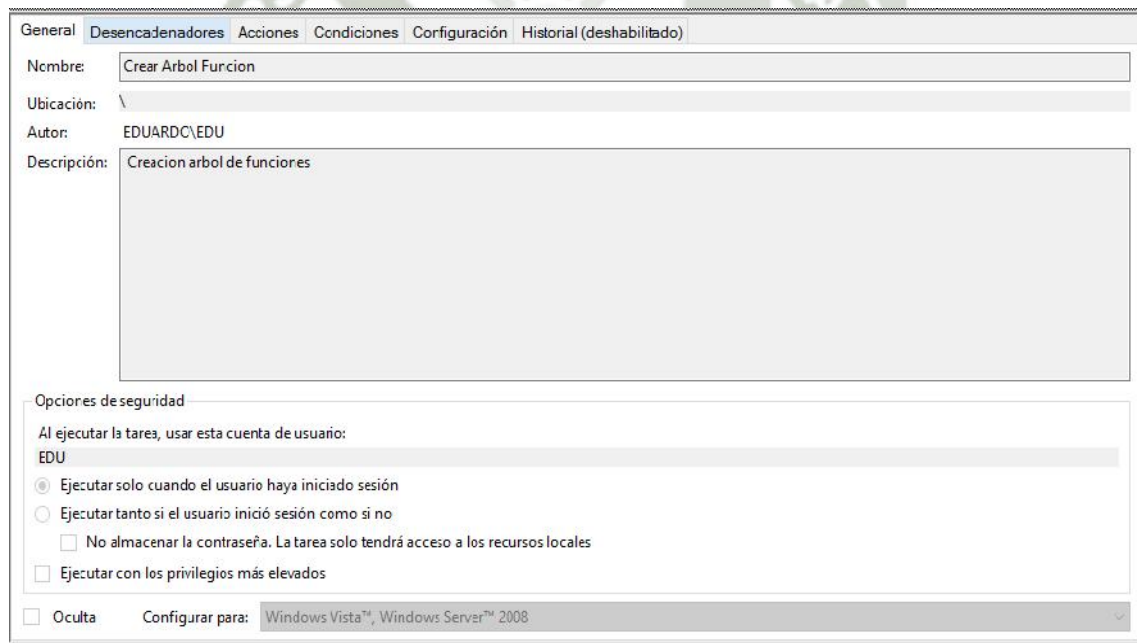


Figura 71: Figura de la programación de la ejecución del árbol de decisiones del sistema experto.

Fuente: Creación propia.

5.35. Instalación de la aplicación

Para poder realizar el despliegue del sistema se publicará el sistema y se alojará la aplicación en el IIS 10 del servidor.

En la figura 72, se observa el IIS donde estará alojada el sistema web.

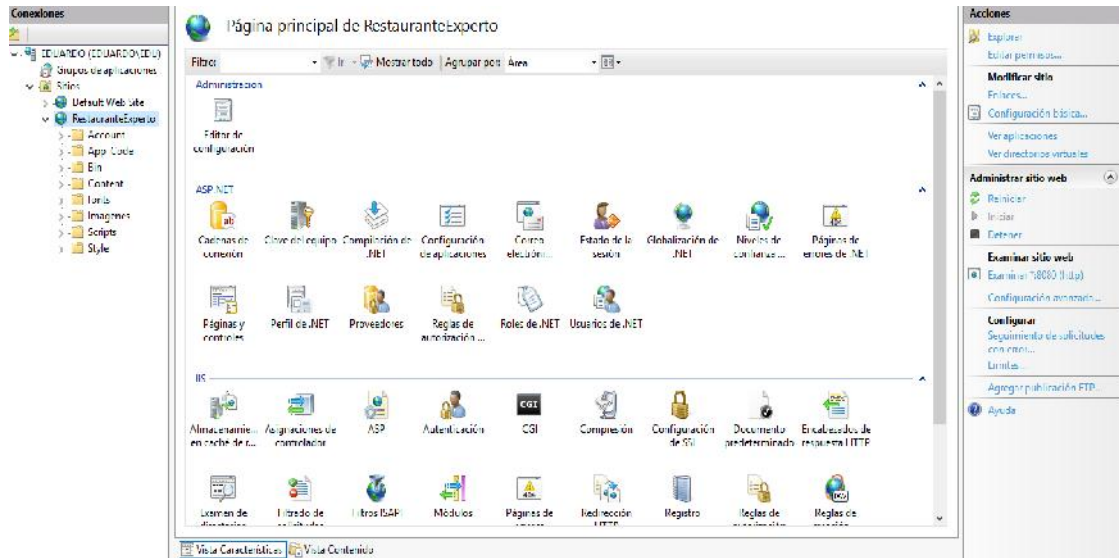


Figura 72: Figura donde se encuentra el IIS del sistema.

Fuente: Creación propia.

El sistema IIS llamará a la aplicación en la ubicación donde tengamos desplegado el sistema.

En la figura 73, se observa la ruta física donde estará la aplicación web y la ruta a la que será accedida por el IIS.

te equipo > Windows8_OS (C:) > inetpub > wwwroot > RestauranteExperto

Nombre	Fecha de modifica...	Tipo	Tamaño
Account	10/03/2017 21:16	Carpeta de archivos	
App_Code	10/03/2017 21:16	Carpeta de archivos	
Bin	10/03/2017 21:16	Carpeta de archivos	
Content	10/03/2017 21:16	Carpeta de archivos	
fonts	10/03/2017 21:16	Carpeta de archivos	
Imágenes	10/03/2017 21:16	Carpeta de archivos	
Scripts	10/03/2017 21:16	Carpeta de archivos	
Style	10/03/2017 21:16	Carpeta de archivos	
About	17/09/2016 23:24	ASP.NET Server Pa...	1 KB
About.aspx.cs	17/09/2016 23:24	Visual C# Source f...	1 KB
Alimentos	07/11/2016 22:04	ASP.NET Server Pa...	8 KB
Alimentos.aspx.cs	07/11/2016 21:53	Visual C# Source f...	7 KB
Almacen	09/01/2017 22:05	ASP.NET Server Pa...	8 KB
Almacen.aspx.cs	09/01/2017 22:11	Visual C# Source f...	4 KB
AsignarActividades	07/11/2016 22:04	ASP.NET Server Pa...	5 KB
AsignarActividades.aspx.cs	07/11/2016 21:44	Visual C# Source f...	4 KB
AsignarPromocionPlato	07/11/2016 22:04	ASP.NET Server Pa...	6 KB
AsignarPromocionPlato.aspx.cs	07/11/2016 21:54	Visual C# Source f...	5 KB
Bundle	03/11/2015 12:22	XML Configuratio...	1 KB
Categorialngrediente	07/11/2016 22:04	ASP.NET Server Pa...	6 KB
Categorialngrediente.aspx.cs	07/11/2016 21:55	Visual C# Source f...	4 KB
CategoriaPlato	07/11/2016 22:04	ASP.NET Server Pa...	6 KB
CategoriaPlato.aspx.cs	07/11/2016 21:56	Visual C# Source f...	4 KB
CerrarSession	26/02/2017 12:32	ASP.NET Server Pa...	1 KB

Figura 73: Figura donde se encuentra los archivos del sistema.

Fuente: Creación propia.

5.36. Vista del Sistema en diferentes dispositivos

El sistema se puede acceder desde diferentes dispositivos por medio de su cualidad responsiva; generando un mayor funcionamiento y practicidad en el uso del mismo.

- **Vista del Ordenador**

En la figura 74, se observa la vista del sistema desde una computadora.



Figura 74: Figura de la vista del sistema desde un ordenador.

Fuente: Creación propia.

- **Vista desde una Tableta.**

En la figura 75, se observa la vista del sistema desde una tableta.



Figura 75: Figura de la vista del sistema desde una tableta.

Fuente: Creación propia.

- **Vista desde un Smartphone**

En la figura 76, se observa la vista del sistema desde una Smartphone.

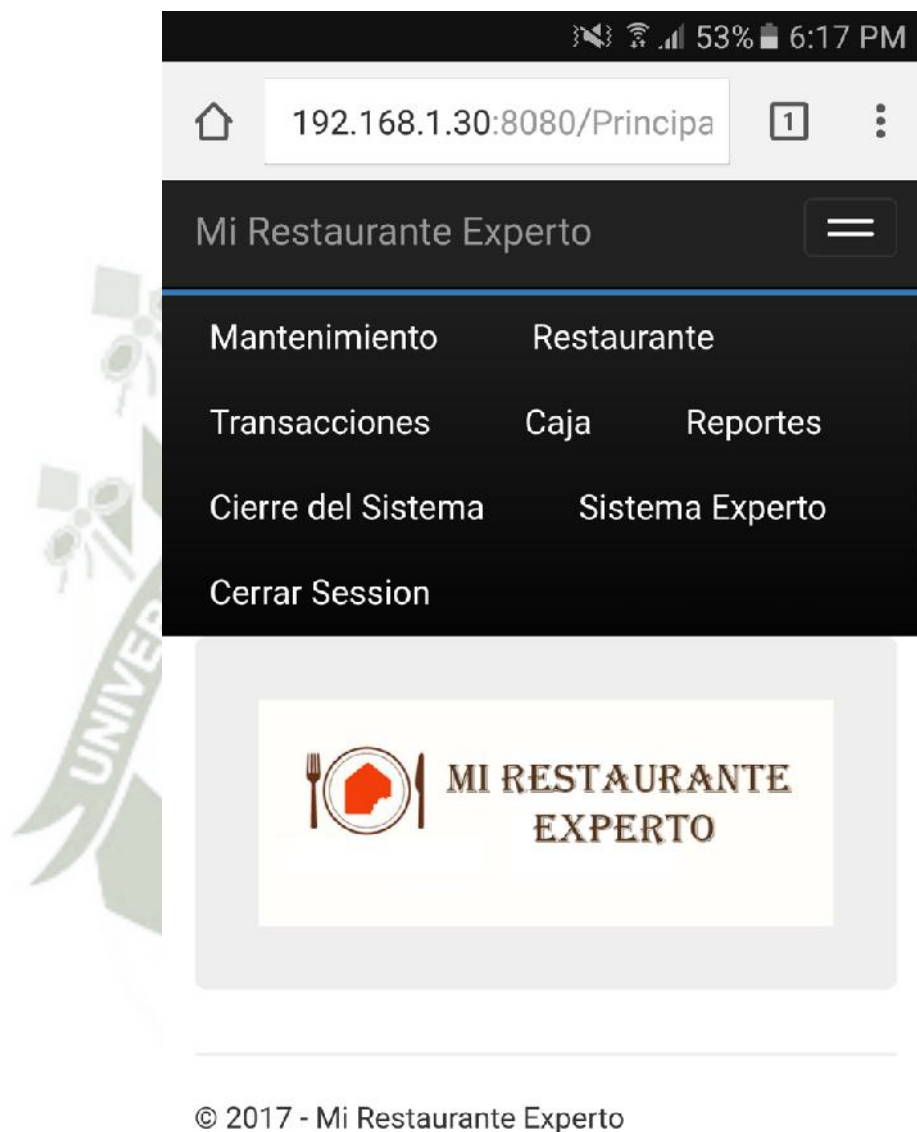


Figura 76: Figura de la vista del sistema desde un Smartphone.

Fuente: Creación propia.

5.37. Pruebas del sistema

En el siguiente apartado se analizarán las pruebas del sistema (ver Anexo D).

5.38. Manuales de usuario

En el presente apartado se mostrará el manual de usuario del sistema experto “Mi Restaurante Experto”. Ver Anexo C.



CAPÍTULO VI

RESULTADOS

En el presente capítulo se analizarán los resultados obtenidos, realizando la comparación de los tiempos de atención a la clientela y las ganancias de las ventas realizadas en el restaurante antes y después y de realizar la implantación del sistema.

El restaurante antes de la implementación presentaba los siguientes tiempos, de acuerdo al día de la semana; los tiempos registrados son los siguientes.

Tabla 3: Tiempos de atención antes de la implantación del sistema experto.

Fuente: Creación propia

Día semana	Tiempo
Lunes	20
Martes	17
Miércoles	18
Jueves	20
Viernes	22
Sábado	23
Domingo	25

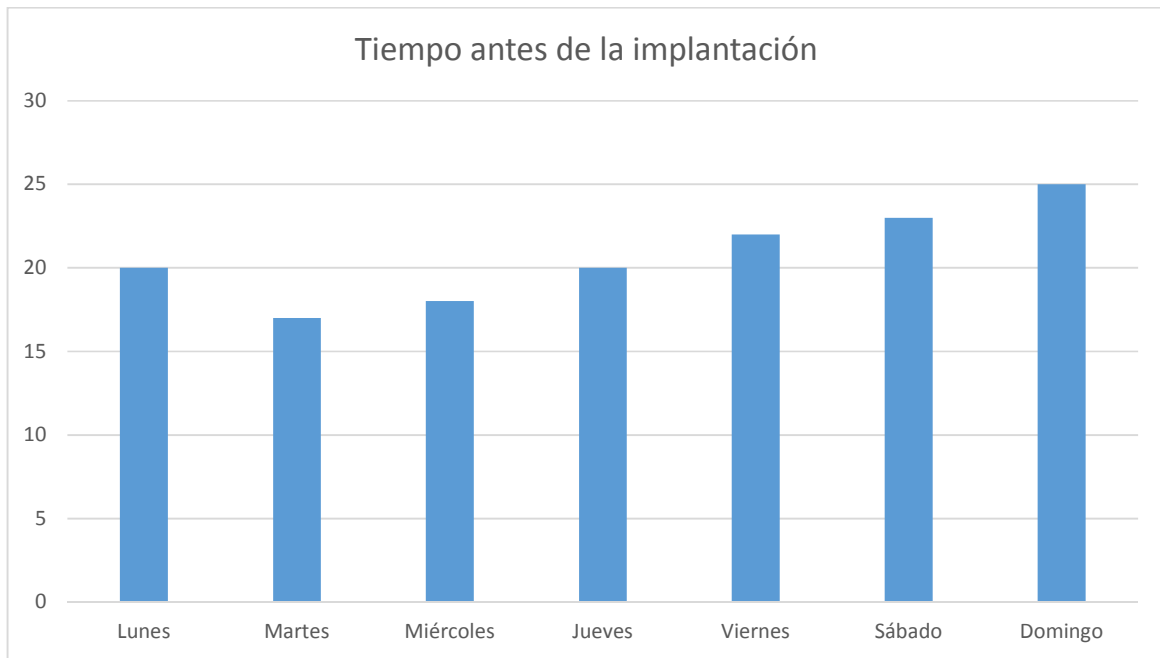


Figura 77: Figura de la atención antes de la Implantación del sistema experto.

Fuente: Creación propia.

El restaurante después de la Implantación presentaba los siguientes tiempos, de acuerdo al día de la semana, los tiempos registrados son los siguientes.

Tabla 4: Tiempos de atención después de la Implantación del sistema experto.

Fuente: Creación propia

Día semana	Tiempo
Lunes	15
Martes	13
Miércoles	14
Jueves	16
Viernes	17
Sábado	17
Domingo	18

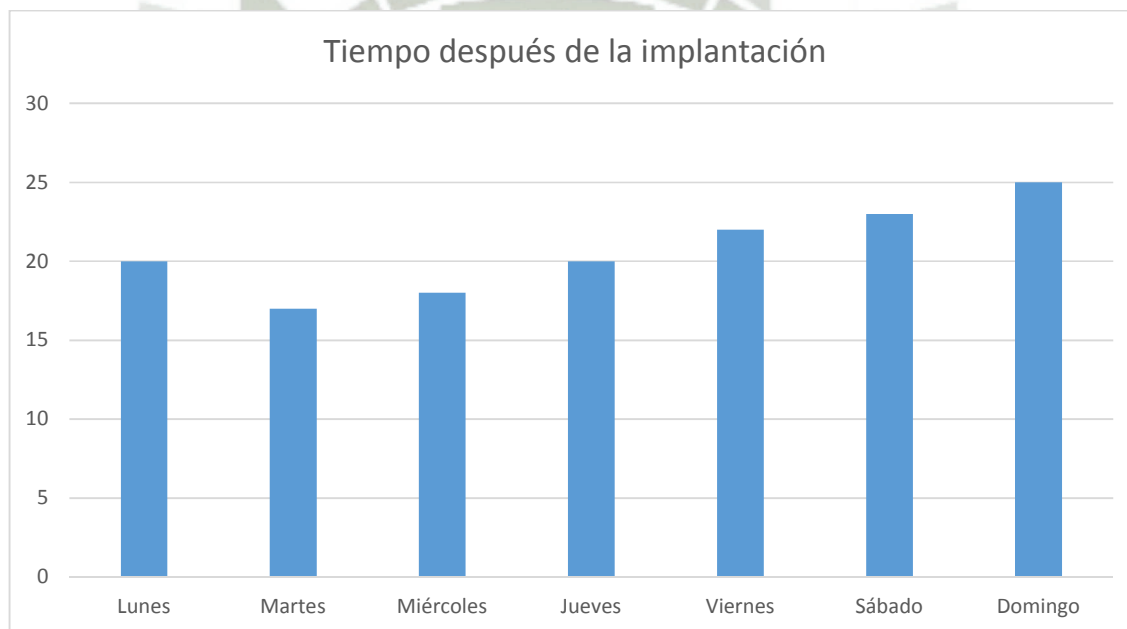


Figura 78: Figura de la atención antes de la implantación del sistema experto.

Fuente: Creación propia.

Sacando los promedios podemos obtener que se redujo en un 25% el tiempo de atención de los clientes, llegando a la conclusión que se mejoró la eficiencia en el proceso de ventas del restaurante.

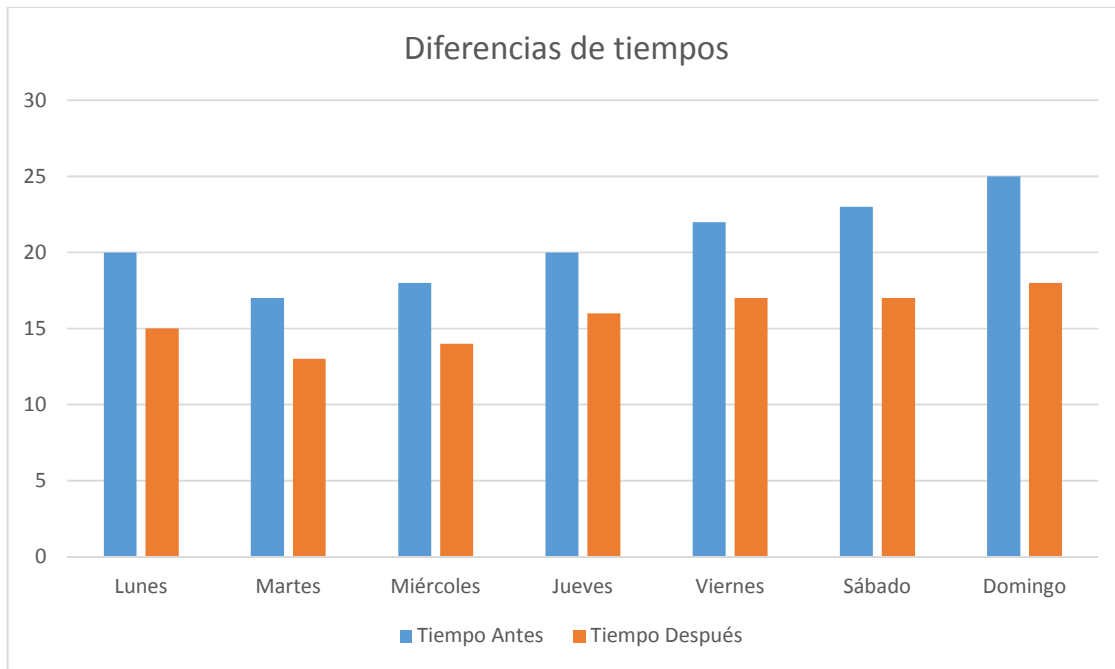


Figura 79: Figura de la diferencia de tiempos de atención de clientes.

Fuente: Creación propia.

La ganancia promedio del restaurante era de 5000 soles mensuales, luego de la implementación se aumentaron las ganancias en un periodo de tres meses.

Tabla 5: Ventas del restaurante.

Fuente: Creación propia.

Etapa	Ventas
Promedio	S/. 5000
Primer Mes	S/. 5400
Segundo Mes	S/. 5800
Tercer Mes	S/. 6100

Se puede apreciar este crecimiento en la siguiente gráfica.



Figura 80: Figura de la evolución de las ventas en el restaurante.

Fuente: Creación propia.

CONCLUSIONES

Primera.- En primer lugar, se desarrolló el sistema experto con las funcionalidades requeridas por las personas involucradas en el proceso de ventas; con este sistema el tiempo de entrega de los productos se redujo considerablemente.

Segunda.- En segundo lugar, se identificó que el punto más crítico del proceso, era el almacenamiento de los datos y la organización que tenían las personas involucradas que desarrollaban el proceso de ventas.

Tercera.- En tercer lugar, se desarrolló los módulos de acuerdo a los requerimientos de los involucrados en el proceso de ventas.

Cuarta.- En cuarto lugar, se desarrolló el sistema experto en un ambiente web, lo que permitió el acceso al sistema desde diferentes dispositivos que se encuentren en su red local.

Quinta.- En quinto lugar, se desarrolló un data warehouse el cual permite el análisis de forma más eficientes de los datos históricos del negocio; a su vez los reportes consumen los datos de esta base de datos, generando que no exista un sobreconsumo de recursos en el sistema transaccional.

Sexta.- En sexto lugar, se desarrolló un sistema experto, el cual toma decisiones en línea de acuerdo a las reglas creadas por los encargados del proceso de ventas.

Sétima- En sétimo lugar, se mejoró la eficiencia del proceso de ventas, disminuyendo el tiempo de la entrega de los productos a la clientela y aumentando las ventas del restaurante.

RECOMENDACIONES

Primera.- Como trabajos futuros el sistema debería enviar alertas como mensajes de textos o mails de alertas de los cambios realizados por el sistema experto.

Segunda.- Se debería crear un algoritmo de análisis de las preferencias de los clientes.

Tercera.- Los pedidos de insumos deberían ser de forma automatizada.



REFERENCIAS BIBLIOGRÁFICAS

- Ayde, A. (2016). *Self learning computer troubleshooting expert system*. Jimma: International Journal of Artificial Intelligence .
- Bernuy, A. (2014). La gestión del conocimiento en el siglo XXI. *Revista digital de la Carrera de Ingeniería de Sistemas*, 131-150.
- Castillo, E. (2016). *Sistemas Expertos y Modelos de Redes Probabilísticas*. Universidad de Cornell.
- Crespo, G. (2013). *Sistema Experto para asistir la decisión de promoción de ventas*. Guayaquil: Escuela superior politécnica del litoral.
- Dresner. (2 de 4 de 2017). *Gartner*. Obtenido de Gartner: <http://www.gartner.com/it-glossary/business-intelligence-bi>
- Gallegos, D. (2016). *Creación de un sistema de gestión automatizada de inventarios para la empresa Liquor's Store*. Quito: Universidad Tecnológica Israel.
- Giraldo, M. (2015). *Diseño de herramienta de Inteligencia de negocios para apoyar la toma de decisiones del área de ventas de un restaurante móvil de sushi*. Bogotá.
- Pavlovic, S. (2015). An ergonomic expert system for risk assessment of work-related musculo-skeletal disorders. *Cornell University*, 130-139.
- Ramirez, A. (2016). *SmartRestaurant*. Castilla: Universidad de Castilla-La Mancha.

Romero, M. (2016). *Aplicación del sistema de costos basado en actividades en los restaurantes dedicados a la elaboración de pollo rostizado del municipio de San Salvador*. San Salvador: Universidad de El Salvador.

Schmal, R. F. (2014). *Optimización del Proceso de Atención al Cliente en un Restaurante durante Períodos de Alta Demanda*. Talca-Chile: Información Tecnológica.

Seco, J. A. (2001). *El lenguaje de programación C#*.



APÉNDICES

Anexo A

En el anexo se encontrarán los diccionarios de datos de las bases de datos utilizadas para la creación del sistema experto:

DB_Restaurante:

Entidad tb_Plato			
Maestra de platos			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoPlato	Código de plato	int	Not null - Autogenerado
vNombrePlato	Nombre del plato	varchar(100)	
iCategoriaPlato	Código de categoría	int	FK
iMoneda	Código de moneda	int	
fCosto	Costo del plato	decimal(10, 2)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Indicador de estado	int	
Campos Clave			
- iCodigoPlato			
Relaciones:			
- iCategoriaPlato - tb_CategoriaPlato			

Entidad tb_Cliente			
Maestra de Clientes			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoCliente	Código del cliente	int	Not null - Autogenerado
vNombreCliente	Nombre del cliente	varchar(70)	
vApellidoMaterno	Apellido materno	varchar(100)	
vApellidoPaterno	Apellido paterno	varchar(100)	
vRazonSocial	Razon social	varchar(100)	
iTipoDocumento	Tipo de documento	int	
vNumeroDocumento	Número de documento	varchar(15)	
dFechaNacimiento	Fecha de nacimiento	date	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
- iCodigoCliente			

Entidad tb_Ingrediente			
Maestra de Ingredientes			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoIngrediente	Código de ingrediente	int	Not null - Autogenerado
vNombreIngrediente	Nombre de ingrediente	varchar(70)	
iCodigoMedida	Código de medida	int	FK
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
iCodigoCategoria	Código de categoría de ingrediente	int	
fcosto	Coso	decimal(10, 2)	
fCantidadMaxima	Cantidad máxima en almacén	decimal(10, 2)	
iTiempoCaducidad	Tiempo de caducidad	int	
fCantidadMinimo	Cantidad mínima	decimal(10, 2)	
fCantidadPromedio	Cantidad promedio usada	decimal(10, 2)	
iTiempoEspera	Tiempo de envío	int	
Campos Clave			
- iCodigoIngrediente			
Relaciones:			
-iCodigoMedida - tb_Medida			

Entidad tb_almacen			
Maestra de Almacen			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoAlmacen	Código de almacén	int	Not null - Autogenerado
iCodigoIngrediente	Código de ingrediente	int	FK
iCantidad	Cantidad	decimal(10, 2)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
- iCodigoAlmacen			
Relaciones:			
-iCodigoIngrediente - tb_Ingrediente			

Entidad tb_Actividad			
Tabla de Actividades del menú de opciones			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoActividad	Código de actividad	int	Not null – Autogenerado
iCodigoPadre	Código del padre de la actividad	int	
vNombreActividad	Nombre de la actividad	varchar(50)	
vlink	Link del formulario	varchar(50)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
iNivel	Nivel en menú de opciones	int	
Campos Clave			
- iCodigoActividad			

Entidad tb_CategorialIngrediente			
Tabla de las categorías de los ingredientes			
Atributo	Descripción del atributo	Tipo de Dato	
iCategorialIngrediente	Código de categoría de ingrediente	int	Not null - Autogenerado
vNombreCategoria	Nombre de categoría	varchar(100)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
- iCategorialIngrediente			

Entidad tb_CategoriaPlato			
Tabla de las categorías de platos			
Atributo	Descripción del atributo	Tipo de Dato	
iCategoriaPlato	Código de categoría de platos	int	Not null - Autogenerado
vNombreCategoria	Nombre de categoría	varchar(70)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
- iCategoriaPlato			

Entidad tb_Empresa			
Tabla de las empresas o sucursales			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoEmpresa	Código de empresa	int	Not null - Autogenerado
vNombreEmpresa	Nombre de empresa	varchar(100)	
vRuc	Ruc de la empresa	varchar(15)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
- iCodigoEmpresa			

Entidad tb_Factura			
Maestra de Facturas			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoFactura	Código de factura	int	Not null - Autogenerado
iCodigoCliente	Código de cliente	int	FK
fMontoTotal	Monto total	decimal(10, 2)	
fMontoIgv	Monto de IGV	decimal(10, 2)	
iMoneda	Moneda	int	
fMontoFinalLocal	Monto final	decimal(10, 2)	
dFecha	Fecha de creación	datetime	
iUsuarioCajero	Usuario de cobro	varchar(15)	
iCodigoLocal	Código de la empresa	int	FK
Campos Clave			
- iCodigoFactura			
Relaciones:			
- iCodigoCliente - tb_Cliente			
-iCodigoLocal – tb_Empresa			

Entidad tb_Fecha			
Tabla de la fecha del sistema			
Atributo	Descripción del atributo	Tipo de Dato	
dFechaActual	Fecha actual	date	
dFechaAnterior	Fecha del día anterior	date	
dFechaSiguiete	Fecha del día siguiente	date	

Entidad tb_Medida			
Tabla de medidas de los ingredientes			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoMedida	Código de medida	int	Not null - Autogenerado
vNombreMedida	Nombre de medida	varchar(50)	
vAbreviatura	Abreviatura de medida	char(3)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
- iCodigoMedida			

Entidad tb_Menu			
Tabla de productos tipo menu o combos de productos			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoMenu	Código del menú	int	Not null - Autogenerado
vNombreMenu	Nombre del menú	varchar(50)	
iMoneda	Moneda	int	
fCosto	Costo	decimal(10, 2)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iCodigoMenu			

Entidad tb_Mesa			
Tabla de las mesas del restaurante			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoMesa	Código de mesa	int	Not null - Autogenerado
vNombreMesa	Nombre de mesa	varchar(70)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creacion	varchar(15)	
vUsuarioModificacion	Usuario de modificacion	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iCodigoMesa			

Entidad tb_Parametro			
Tabla de parámetros			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoTabla	Código de tabla	int	Not null
iCodigoParametro	Código del parámetro	int	Not null
vNombreParametro	Nombre del parámetro	varchar(100)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iCodigoTabla - iCodigoParametro			

Entidad tb_Pedido			
Tabla de pedidos			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoPedido	Código del pedido	int	Not null - Autogenerado
iCodigoMesa	Código de mesa	int	FK
iCodigoPlato	Código de plato	int	FK
iCantidad	Cantidad	int	
iObservacion	Observación	varchar(1000)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iCodigoPedido			
Relaciones:			
- iCodigoMesa – tb_Mesa			
- iCodigoPlato – tb_Plato			

Entidad tb_Rol			
Tabla de roles de usuarios			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoRol	Código de rol	int	Not null - Autogenerado
vNombreRol	Nombre de rol	varchar(70)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iCodigoRol			

Entidad tb_TablaParametro			
Tabla de descripción general de parámetros del sistema			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoTabla	Código de tabla	int	
vNombreTabla	Nombre de tabla	varchar(70)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	

Entidad tb_Ubigeo			
Tabla de ubicaciones			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoUbigeo	Código de ubigeo	int	Not null - Autogenerado
vDepartamento	Departamento	varchar(50)	
vProvincia	Provincia	varchar(70)	
vDistrito	Distrito	varchar(70)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iCodigoUbigeo			



Entidad tb_Usuario			
Maestra de usuarios			
Atributo	Descripción del atributo	Tipo de Dato	
vCodigoUsuario	Código de usuario	varchar(15)	Not null - Autogenerado
vContraseña	Contraseña	varchar(20)	
vNombreUsuario	Nombre de usuario	varchar(50)	
vApellidoMaterno	Apellido materno	varchar(100)	
vApellidoPaterno	Apellido paterno	varchar(50)	
iCodigoRol	Código de rol	int	FK
vTelefono	Teléfono	varchar(10)	
vCelular	Celular	varchar(10)	
iCodigoUbigeo	Código de ubigeo	int	FK
iTipoDocumento	Tipo de documento	int	
vDocumento	Número de documento	varchar(12)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-vCodigoUsuario			
Relaciones:			
-iCodigoRol - tb_Rol			
- iCodigoUbigeo – tb_Ubigeo			

Entidad tb_Promocion			
Maestra de promociones de productos			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoPromocion	Código de promoción	int	Not null - Autogenerado
vNombrePromocion	Nombre de promoción	varchar(70)	
iTipoDescuento	Tipo de descuento	int	
fPorcentaje	Porcentaje de descuento	decimal(10, 2)	
fMonto	Monto de descuento	decimal(10, 2)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
dFechaInicio	Fecha de inicio	datetime	
dFechaFin	Fecha de término	datetime	
Campos Clave			
-iCodigoPromocion			

Entidad tb_CategoriaPromocion			
Tabla de categorías de promociones			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoCategoria	Código de categoría de promoción	int	Not null - Autogenerado
vNombreCategoria	Nombre de categoría	varchar(70)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iCodigoCategoria			

Entidad tb_DetalleFactura			
Tabla de detalle de factura			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoFactura	Código de factura	int	FK
iCodigoMenu	Código de menú	int	FK
iCodigoProducto	Código de producto	int	FK
iMoneda	Moneda	int	
dMontoLocal	Monto	decimal(10, 2)	
iCantidad	Cantidad	int	
iPedido	Código de pedido	int	FK
Relaciones:			
- iCodigoFactura - tb_Factura - iCodigoMenu – tb_Menu - iCodigoProducto – tb_Producto - iPedido – tb_Pedido			

Entidad tb_Restriccion			
Tabla de restricciones de promoción			
Atributo	Descripción del atributo	Tipo de Dato	
iRestriccion	Código de restricción	int	Not null - Autogenerado
iCodigoPromocion	Código de promoción	int	FK
iTipoDescuento	Tipo de descuento	int	
dNumeroCondicion	Cantidad de descuento	decimal(10, 2)	
iCondicion	Código de condición	int	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iRestriccion			
Relaciones:			
-iCodigoPromocion - tb_Promocion			

Entidad tb_PMenuPlato			
Puente entre menu y plato			
Atributo	Descripción del atributo	Tipo de Dato	
iMenuPlato	Código de puente	int	Not null - Autogenerado
iCodigoMenu	Código de menú	int	FK
iCodigoPlato	Código de plato	int	FK
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iMenuPlato			
Relaciones:			
-iCodigoMenu - tb_Menu			
-iCodigoPlato - tb_Plato			

Entidad tb_PPlatoIngrediente			
Puente entre plato e ingredientes			
Atributo	Descripción del atributo	Tipo de Dato	
iPlatoIngrediente	Código de puente	int	Not null - Autogenerado
iCodigoPlato	Código de plato	int	FK
iCodigoIngrediente	Código de ingrediente	int	FK
fCantidad	Cantidad	decimal(10, 2)	
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iPlatoIngrediente			
Relaciones:			
-iCodigoPlato - tb_Plato			
-iCodigoIngrediente - tb_Ingrediente			

Entidad Tb_PPromoPlato			
Puente entre promoción y platos			
Atributo	Descripción del atributo	Tipo de Dato	
iPromoPlato	Código de puente	int	Not null - Autogenerado
iCodigoPromocion	Código de promoción	int	FK
iCodigoPlato	Código de plato	int	FK
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iPromoPlato			
Relaciones:			
-iCodigoPromocion - tb_Promocion			
- iCodigoPlato – tb_Plato			

Entidad tb_PRolActividad			
Puente entre roles y actividades			
Atributo	Descripción del atributo	Tipo de Dato	
iRolActividad	Código de puente	int	Not null - Autogenerado
iCodigoRol	Código de rol	int	FK
iCodigoActividad	Código de actividad	int	FK
dCreacion	Fecha de creación	datetime	
dModificacion	Fecha de modificación	datetime	
vUsuarioCreacion	Usuario de creación	varchar(15)	
vUsuarioModificacion	Usuario de modificación	varchar(15)	
iEstado	Estado	int	
Campos Clave			
-iRolActividad			
Relaciones:			
-iCodigoRol - tb_Rol			
- iCodigoActividad – tb_Actividad			

DM_Restaurante:

Entidad di_Ingrediente			
Dimensión de tipo de documento			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCategorialIngrediente	Código de ingrediente	int	PK
vNombreCategoria	Nombre de categoría	varchar(100)	
Campos Clave			
-dFecha - iCategorialIngrediente			

Entidad di_Moneda			
Dimensión de tipo de documento			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoMoneda	Código de moneda	int	PK
vNombreMoneda	Descripción de moneda	varchar(100)	
Campos Clave			
- dFecha - iCodigoMoneda			

Entidad di_CategoriaPlato			
Dimensión de Categoría de plato			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
icategoriaPlato	Categoría del plato	int	PK
vNombreCategoria	Nombre Categoría	varchar(70)	
Campos Clave			
-dFecha			

Entidad di_Cliente			
Dimensión de clientes			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoCliente	Código de cliente	int	PK
vnombreCliente	Nombre del cliente	varchar(70)	
vApellidoMaterno	Apellido materno	varchar(100)	
vApellidoPaterno	Apellido paterno	varchar(100)	
vRazonSocial	Razón social	varchar(100)	
vNumeroDocumento	Número de documento	varchar(15)	
dFechaNacimiento	Fecha de nacimiento	date	
iedad	Edad	int	
Campos Clave			
-dFecha			
- iCodigoCliente			

Entidad di_CategoriaPlato			
Dimensión de categorías de platos			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
icategoriaPlato	Categoría plato	int	PK
vNombreCategoria	Nombre Categoría	varchar(70)	
Campos Clave			
-dFecha			
- icategoriaPlato			

Entidad di_Promocion			
Dimensión de promoción			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoPromocion	Código de promoción	int	PK
vnombreCategoria	Nombre de categoría	varchar(100)	
vnombrePromocion	Nombre de promoción	varchar(70)	
fmonto	Monto	decimal(10, 2)	
dFechaI	Fecha de inicio	datetime	
dFechaF	Fecha de fin	datetime	
Campos Clave			
-dFecha			

Entidad di_Usuario			
Dimensión de usuarios			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
vcodigoUsuario	Código de usuario	varchar(15)	PK
vcontraseña	Contraseña	varchar(20)	
vnombreUsuario	Nombre de usuario	varchar(50)	
vApellidoPaterno	Apellido paterno	varchar(50)	
vApellidoMaterno	Apellido materno	varchar(100)	
vTelefono	Teléfono	varchar(10)	
vCelular	Celular	varchar(10)	
iTipoDocumento	Tipo de documento	int	
vDocumento	Documento	varchar(12)	
Campos Clave			
-dFecha			
-vcodigoUsuario			

Entidad di_CategorialIngrediente			
Dimensión de categorías de ingredientes			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCategorialIngrediente	Código categoría de ingrediente	int	PK
vNombreCategoria	Nombre de categoría	varchar(100)	
Campos Clave			
-dFecha			
-iCategorialIngrediente			

Entidad di_Empresa			
Dimensión de empresas			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoEmpresa	Código de empresa	int	PK
vNombreEmpresa	Nombre de empresa	varchar(100)	
vRuc	Ruc	varchar(15)	
Campos Clave			
-dFecha			
-iCodigoEmpresa			

Entidad di_Ingrediente			
Dimensión de Ingredientes			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoIngrediente	Código de ingrediente	int	PK
vNombreIngrediente	Nombre de ingrediente	varchar(70)	
Campos Clave			
-dFecha			
- iCodigoIngrediente			

Entidad di_Medida			
Dimensión de Medida			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoMedida	Código de medida	int	PK
vNombreMedida	Nombre de medida	varchar(50)	
vAbreviatura	Abreviatura	char(3)	
Campos Clave			
-dFecha			

Entidad di_Menu			
Dimensión de Menu			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoMenu	Código de menú	int	PK
vNombreMenu	Nombre de menú	varchar(50)	
Campos Clave			
-dFecha			
- iCodigoMenu			

Entidad di_Mesa			
Dimensión de Mesas			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoMesa	Código de mesa	int	PK
vNombreMesa	Nombre de mesa	varchar(70)	
Campos Clave			
-dFecha			
- iCodigoMesa			

Entidad di_Plato			
Dimensión de Platos			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoPlato	Código de plato	int	PK
vNombrePlato	Nombre de plato	varchar(100)	
Campos Clave			
-dFecha			
- iCodigoPlato			

Entidad di_Rol			
Dimensión de Roles			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoRol	Código de rol	int	PK
vnombrerol	Nombre del rol	varchar(70)	
Campos Clave			
-dFecha			
- iCodigoRol			

Entidad di_Ubigeo			
Dimensión de Ubigeo			
Atributo	Descripción del atributo	Tipo de Dato	
dFecha	Fecha	date	PK
iCodigoUbigeo	Código de ubigeo	int	PK
vdepartamento	Departamento	varchar(50)	
vprovincia	Provincia	varchar(70)	
vDistrito	Distrito	varchar(70)	
Campos Clave			
-dFecha			
- iCodigoUbigeo			

Entidad di_fecha			
Dimensión de Fecha			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoFecha	Código de fecha	int	PK
dFecha	Fecha	date	
iAño	Año	int	
iMes	Mes número	int	
iDia	Día	int	
iTrimestre	Trimestre	nvarchar(30)	
vMes	Mes descripción	nvarchar(30)	
iDiaAño	Día del año	nvarchar(30)	
iSemana	Semana	nvarchar(30)	
iDiaSemana	Día de semana	int	
vNombreDia	Nombre del día	nvarchar(30)	
Campos Clave			
- iCodigoFecha			

Entidad di_Hora			
Dimensión de Hora			
Atributo	Descripción del atributo	Tipo de Dato	
iHora	Icódigo de hora	int	PK
vHora	Hora descripción	varchar(30)	
iNumeroHora	Hora	int	
iMinuto	Minuto	int	
Campos Clave			
-iHora			

Tabla tb_HechosPagos			
Tabla de hechos de pagos			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoFecha	Código de fecha	int	FK
iHora	Código de hora	int	FK
vUsuarioPedido	Usuario pedido	varchar(15)	FK
vUsuarioCajero	Usuario Cajero	varchar(15)	FK
icodigoCliente	Código del cliente	int	FK
iRolPedido	Código de rol	int	FK
iRolFactura	Código de factura	int	FK
iTipoDocumento	Código tipo de documento	int	FK
iEmpresa	Código de empresa	int	FK
iUbigeoPedido	Código de ubigeo pedido	int	FK
iUbigeoCajero	Código de ubigeo cajero	int	FK
iCodigoMesa	Código de mesa	int	FK
iCategoriaPlato	Código categoría plato	int	FK
iCodigoPlato	Código plato	int	FK
iCodigoMenu	Código de menú	int	FK
iCodigoPromocion	Código de promoción	int	FK
iMoneda	Código de moneda	int	
fMonto	Monto	decimal(10, 2)	
fDescuento	Descuento	decimal(10, 2)	
iCantidad	Cantidad	int	
fMontoFinal	Monto final	decimal(10, 2)	
fMontoIgv	Monto IGV	decimal(12, 2)	
fMontoTotal	Monto total	decimal(11, 2)	
iTiempoPedido	Tiempo pedido	int	
iCodigoPedido	Código del pedido	int	FK
Relaciones:			
-iCodigoRol - tb_Rol -iCodigoFecha – di_fecha -iHora – di_Hora -vUsuarioPedido – di_Usuario -vUsuarioCajero – di_Usuario -icodigoCliente – di_Cliente -iRolPedido – di_Rol -iRolFactura – di_Rol -iTipoDocumento – di_TipoDocumento -iEmpresa – di_Empresa -iUbigeoPedido – di_Ubigeo -iUbigeoCajero – di_Ubigeo -iCodigoMesa – di_Mesa -iCategoriaPlato –di_CategoriaPlato -iCodigoPlato – di_Plato -iCodigoMenu –di_Menu -iCodigoPromocion – di_Promocion -iMoneda – di_Moneda			

Tabla tb_HechosIngredientes			
Tabla de hechos de Almacén e ingredientes			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoFecha	Código de fecha	int	FK
iEmpresa	Código de empresa	int	FK
iCategorialIngrediente	Código de categoría de ingrediente	int	FK
iCodigoIngrediente	Código de ingrediente	int	FK
iCodigoMedida	Código de medida	int	FK
fCosto	Costo	decimal(10, 2)	
fCantidadMaxima	Cantidad Máxima	decimal(10, 2)	
iTiempoCaducidad	Tiempo de caducidad	int	
fCantidadMinimo	Cantidad mínima	decimal(10, 2)	
fCantidadPromedio	Cantidad promedio	decimal(10, 2)	
iTiempoEspera	Tiempo de espera	int	
CantidadGastada	Cantidad gastada	decimal(38, 2)	
MontoGastado	Monto Gastado	decimal(10, 2)	
Relaciones:			
<ul style="list-style-type: none"> - iCodigoFecha – di_Fecha - iEmpresa – di_Empresa -iCategorialIngrediente – di_CategorialIngrediente - iCodigoIngrediente- di_Ingrediente - iCodigoMedida – di_Medida 			

KB_Restaurante:

Entidad tb_Cantidades			
Tabla de la relación entre venta de platos			
Atributo	Descripción del atributo	Tipo de Dato	
iDiaSemana	Día de la semana	int	
iplato1	Plato 1	int	
iplato2	Plato 2	int	
iCantidad	Cantidad	int	
iEstado	Estado	int	

Entidad tb_Hcantidades			
Tabla del histórico de la relacion entre venta de platos			
Atributo	Descripción del atributo	Tipo de Dato	
Dia	Fecha de ejecución	date	PK
iDiaSemana	Día de semana	int	
iplato1	Plato 1	int	
iplato2	Plato 2	int	
iCantidad	Cantidad	int	
iEstado	Estado	int	
Campos Clave			
-Dia			

Entidad tb_Arbol			
Tabla de la relación de los nodos del árbol de decisiones			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoNodo	Código del nodo	int	FK
iHijoSi	Nodo respuesta Si	int	FK
iHijoNo	Nodo respuesta No	int	FK
vUsuarioCreacion	Usuario de Creación	varchar(12)	
dCreacion	Fecha de creación	datetime	
Relaciones:			
- iCodigoNodo – tb_Nodo - iHijoSi – tb_Nodo - iHijoNo – tb_Nodo			

Entidad tb_Nodo			
Tabla de nodos			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoNodo	Código nodo	int	
vNombreNodo	Nombre del nodo	varchar(200)	
iTipoNodo	Tipo de nodo	int	
vUsuarioCreacion	Usuario de creación	varchar(12)	
dCreacion	Fecha de creación	datetime	
Campos Clave			
-iCodigoNodo			

Entidad tb_Hora			
Tabla de horas			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoHora	Código de hora	int	Not null - Autogenerado
iHorainicio	Hora de inicio	int	
iHoraFin	Hora fin	int	
iHoraPico	Indicador hora principal	int	
Campos Clave			
-iCodigoHora			

Entidad tb_ResultadoLinea			
Tabla temporal de resultados en línea			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigoNodo	Código de nodo	int	FK
iRespuesta	Indicador de respuesta	int	
Relaciones:			
- iCodigoNodo – tb_Nodo			

Entidad tb_LogSistemaExperto			
Log del sistema experto			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigo	Código ingreso	int	Not null - Autogenerado
vNombre	Nombre o descripción	varchar(100)	
iCategoria	Cateogoría	int	
iEstado	Estado	int	
dCreacion	Fecha de creación	datetime	
Campos Clave			
-iCodigo			

Entidad tb_LogCierre			
Log del sistema experto			
Atributo	Descripción del atributo	Tipo de Dato	
iCodigo	Código del cierre	int	Not null - Autogenerado
dFechaC	Fecha cierre sistema	date	
dFechaCrecion	Fecha de ejecución	datetime	
Campos Clave			
-iCodigo			



Anexo B

Código de la funcionalidad del sistema experto

Para el desarrollo de la funcionalidad del sistema experto se utilizará el data warehouse del sistema, ya que nos servirá de primera fuente para la generación de los reportes.

El script para la generación de los datos al data warehouse es el siguiente.

```
USE [master]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

--exec SP_Cierre
CREATE Procedure [dbo].[SP_Cierre]
as
begin
Declare @Fecha date
Declare @BackInicio varchar(1000)

select @Fecha = dFechaActual from DB_Restaurante.dbo.tb_Fecha

select @BackInicio=format(@fecha, 'yyyyMMdd')
select @BackInicio
select @BackInicio='C:\Backup Tesis 2\DB_Restaurante_'+@BackInicio+'.bak'

BACKUP DATABASE [DB_Restaurante] TO DISK = @BackInicio
WITH NOFORMAT, NOINIT, NAME = N'DB_Restaurante-Full Database Backup',
SKIP, NOREWIND, NOUNLOAD, STATS = 10

--di_Cliente
insert into DM_Restaurante.dbo.di_Cliente
Select
@Fecha as dFecha,
cli.iCodigoCliente,
cli.vnombreCliente,
cli.vApellidoMaterno,
cli.vApellidoPaterno,
cli.vRazonSocial,
cli.vNumeroDocumento,
cli.dFechaNacimiento,
datediff(year,cli.dFechaNacimiento,@Fecha) as iedad
from DB_Restaurante.dbo.tb_Cliente cli
where cli.iestado=1

insert into DM_Restaurante.dbo.di_TipoDocumento
```

```

select
@Fecha as dFecha,
iCodigoParametro as iCodigoDocumento,
vnombreParametro as vNombreDocumento
from
DB_Restaurante.dbo.tb_Parametro
where
iCodigoTabla=1 and iEstado=1

insert into DM_Restaurante.dbo.di_Empresa
select
@Fecha as dFecha,
iCodigoEmpresa,
vNombreEmpresa,
vRuc
from DB_Restaurante.dbo.tb_Empresa
where iEstado=1

insert into DM_Restaurante.dbo.di_Medida
select
@Fecha as dFecha,
iCodigoMedida,
vNombreMedida,
vAbreviatura
from DB_Restaurante.dbo.tb_Medida
where iEstado=1

insert into DM_Restaurante.dbo.di_CategoriaIngrediente
select
@Fecha as dFecha,
iCategoriaIngrediente,
vNombreCategoria
from DB_Restaurante.dbo.tb_CategoriaIngrediente
where iEstado=1

insert into DM_Restaurante.dbo.di_Ingrediente
select
@Fecha as dFecha,
ing.iCodigoIngrediente,
ing.vNombreIngrediente
from DB_Restaurante.dbo.tb_Ingrediente ing
where ing.iEstado=1

insert into DM_Restaurante.dbo.di_Menu
select
@Fecha as dFecha,
iCodigoMenu,
vNombreMenu
from DB_Restaurante.dbo.tb_Menu men
where men.iEstado=1

insert into DM_Restaurante.dbo.di_Moneda
select
@Fecha as dFecha,
iCodigoParametro as iCodigoMoneda,
vNombreParametro as vNombreMoneda
from DB_Restaurante.dbo.tb_Parametro
where
iEstado=1 and
iCodigoTabla=4

```

```

insert into DM_Restaurante.dbo.di_Mesa
select
@Fecha as dFecha,
iCodigoMesa,
vNombreMesa
from DB_Restaurante.dbo.tb_Mesa
where iEstado=1

insert into DM_Restaurante.dbo.di_Plato
select
@Fecha as dFecha,
pla.iCodigoPlato,
pla.vNombrePlato
from DB_Restaurante.dbo.tb_Plato pla
where pla.iEstado=1

insert into DM_Restaurante.dbo.di_CategoriaPlato
select *
from
((select @Fecha as dFecha,0 as icategoriaPlato,'Menu' as vNombreCategoria)
union(
select
@Fecha as dFecha,
iCategoriaPlato,
vNombreCategoria
from
DB_Restaurante.dbo.tb_CategoriaPlato
where iestado=1
)) a

insert into DM_Restaurante.dbo.di_Promocion
select
@Fecha as dFecha,
pro.iCodigoPromocion,
tip.vnombreparametro as vnombreCategoria,
pro.vnombrePromocion,
case when pro.iTipoDescuento=1 then pro.fMonto else pro.fPorcentaje end as
fmonto,
pro.dFechaInicio,
pro.dFechaFin
from DB_Restaurante.dbo.tb_Promocion pro
inner join DB_Restaurante.dbo.tb_Parametro tip
on tip.iCodigoTabla= 5 and pro.iTipoDescuento = tip.iCodigoParametro and
tip.iestado=1
where pro.iEstado=1

insert into DM_Restaurante.dbo.di_Ubigeo
select
@Fecha as dFecha,
iCodigoUbigeo,
vdepartamento,
vprovincia,
vDistrito
from DB_Restaurante.dbo.tb_Ubigeo
where iEstado=1

insert into DM_Restaurante.dbo.di_Usuario

```

```

select
@Fecha as dFecha,
usu.vcodigoUsuario,
usu.vcontraseña,
usu.vnombreUsuario,
usu.vApellidoPaterno,
usu.vApellidoMaterno,
usu.vTelefono,
usu.vCelular,
usu.iTipoDocumento,
usu.vDocumento
from DB_Restaurante.dbo.tb_Usuario usu
where usu.iEstado=1

insert into DM_Restaurante.dbo.di_Rol
select
@Fecha as dFecha,
iCodigoRol,
vnombrerol
from DB_Restaurante.dbo.tb_Rol
where iEstado=1

insert into DM_Restaurante.dbo.tb_HechosPagos
select
fec.iCodigoFecha,
hor.iHora,
ped.vUsuarioModificacion as vUsuarioPedido,
fac.iUsuarioCajero as vUsuarioCajero,
fac.icodigoCliente,
upe.iCodigoRol as iRolPedido,
ufa.iCodigoRol as iRolFactura,
cli.iTipoDocumento,
1 as iEmpresa,
upe.iCodigoUbigeo as iUbigeoPedido,
ufa.iCodigoUbigeo as iUbigeoCajero,
ped.iCodigoMesa,
case when det.iCodigoMenu is not null then 0 else pla.iCategoriaPlato end
as iCategoriaPlato,
pla.iCodigoPlato,
case when det.iCodigoMenu is not null then det.iCodigoMenu else null end
as iCodigoMenu,
pro.iCodigoPromocion,
fac.iMoneda,
det.dMontoLocal as fMonto,
cast(det.icantidad*(case when pro.itipoDescuento=1 then pro.fmonto when
pro.itipoDescuento=2 then cast(pro.fPorcentaje*pla.fCosto/100 as
decimal(10,2)) else 0 end) as decimal(10,2)) as fDescuento,
det.iCantidad,

cast((det.dMontoLocal -
cast(det.icantidad*(case when pro.itipoDescuento=1 then pro.fmonto when
pro.itipoDescuento=2 then cast(pro.fPorcentaje*pla.fCosto/100 as
decimal(10,2)) else 0 end) as decimal(10,2)))*0.87 as decimal(10,2))as
fMontoFinal,
(
det.dMontoLocal -
cast(det.icantidad*(case when pro.itipoDescuento=1 then pro.fmonto when
pro.itipoDescuento=2 then cast(pro.fPorcentaje*pla.fCosto/100 as
decimal(10,2)) else 0 end) as decimal(10,2))
)
-
(

```

```

cast((det.dMontoLocal -
cast(det.icantidad*(case when pro.itipoDescuento=1 then pro.fmonto when
pro.itipoDescuento=2 then cast(pro.fPorcentaje*pla.fCosto/100 as
decimal(10,2)) else 0 end) as decimal(10,2)) )*0.87 as decimal(10,2))
) as fMontoIgv,
det.dMontoLocal -
cast(det.icantidad*(case when pro.itipoDescuento=1 then pro.fmonto when
pro.itipoDescuento=2 then cast(pro.fPorcentaje*pla.fCosto/100 as
decimal(10,2)) else 0 end) as decimal(10,2)) as fMontoTotal,
DATEDIFF(minute,ped.dCreacion,fac.dFecha) as iTiempoPedido,
ped.iCodigoPedido
from DB_Restaurante.dbo.tb_factura fac
inner join DB_Restaurante.dbo.tb_detallefactura det
on det.iCodigoFactura=fac.iCodigoFactura
inner join DB_Restaurante.dbo.tb_pedido ped
on ped.iCodigoPedido=det.iPedido
inner join DB_Restaurante.dbo.tb_Usuario upe
on upe.vCodigoUsuario=ped.vUsuarioModificacion and upe.iEstado=1
inner join DB_Restaurante.dbo.tb_Usuario ufa
on ufa.vCodigoUsuario=fac.iusuarioCajero and ufa.iEstado=1
inner join DB_Restaurante.dbo.tb_Cliente cli
on cli.iCodigoCliente= fac.iCodigoCliente and cli.iEstado=1
left join DB_Restaurante.dbo.tb_Plato pla
on pla.iCodigoPlato= det.iCodigoProducto and pla.iEstado=1
left join DB_Restaurante.dbo.tb_Menu men
on men.icodigomenu = det.iCodigoMenu and men.iEstado=1
left join DB_Restaurante.dbo.Tb_PPromoPlato ppro
on ppro.iCodigoPlato=pla.iCodigoPlato and ppro.iestado=1
left join DB_Restaurante.dbo.tb_Promocion pro
on pro.iCodigoPromocion = ppro.icodigopromocion and pro.iestado=1
left join DM_Restaurante.dbo.di_Fecha fec
on fec.dFecha = cast(fac.dFecha as date)
left join DM_Restaurante.dbo.di_Hora hor
on CONVERT(varchar, cast(fac.dFecha as datetime), 108) = hor.vHora
where cast(fac.dfecha as date)= @Fecha

insert into DM_Restaurante.dbo.tb_HechosIngredientes
select
fec.iCodigoFecha,
1 as iEmpresa,
ing.iCodigoCategoria as iCategoriaIngrediente,
ing.iCodigoIngrediente,
ing.iCodigoMedida,
ing.fCosto,
ing.fCantidadMaxima,
ing.iTiempoCaducidad,
ing.fCantidadMinimo,
ing.fCantidadPromedio,
iTiempoEspera,
sum(ppi.fCantidad) as CantidadGastada,
cast(sum(ppi.fCantidad)*ing.fCosto as decimal(10,2)) as MontoGastado

from DB_Restaurante.dbo.tb_Factura fac
inner join DB_Restaurante.dbo.tb_DetalleFactura det
on fac.iCodigoFactura = det.iCodigoFactura
inner join DB_Restaurante.dbo.tb_Plato pla
on pla.iCodigoPlato = det.iCodigoProducto and pla.iEstado=1
inner join DB_Restaurante.dbo.tb_PPlatoIngrediente ppi
on ppi.iCodigoPlato = pla.iCodigoPlato and pla.iEstado=1
inner join DB_Restaurante.dbo.tb_Ingrediente ing
on ing.iCodigoIngrediente = ppi.iCodigoIngrediente and ing.iEstado=1

```

```
inner join DB_Restaurante.dbo.tb_Almacen alm
on alm.iCodigoIngrediente = ing.iCodigoIngrediente and alm.iEstado=1
inner join DB_Restaurante.dbo.tb_Medida med
on med.iCodigoMedida = ing.iCodigoMedida
left join DM_Restaurante.dbo.di_Fecha fec
on fec.dFecha = cast(fac.dFecha as date)
where cast(fac.dfecha as date)= @Fecha
group by
fec.iCodigoFecha,
ing.iCodigoCategoria,
ing.iCodigoIngrediente,
ing.iCodigoMedida,
ing.fCantidadMaxima,
ing.iTiempoCaducidad,
ing.fCantidadMinimo,
ing.fCantidadPromedio,
iTiempoEspera,
ing.fCosto
```

```
Select @BackInicio=@Fecha
Select @BackInicio='C:\Backup Tesis
2\DM_Backup\DM_Restaurante_'+@BackInicio+'.bak'
BACKUP DATABASE [DB_Restaurante] TO DISK = @BackInicio WITH NOFORMAT,
NOINIT, NAME = N'DB_Restaurante-Full Database Backup', SKIP, NOREWIND,
NOUNLOAD, STATS = 10
set @Fecha=DATEADD(day,1,@Fecha)
select @Fecha
```

```
update DB_Restaurante.dbo.tb_Fecha
set
dfechaActual = DATEADD(day,1,dfechaActual),
dfechaAnterior = DATEADD(day,1,dfechaAnterior),
dfechaSiguiente = DATEADD(day,1,dfechaSiguiente)

end
GO
```

Procedimiento de ejecución del árbol de decisión

```
Create procedure [dbo].[SP_FuncionalArbol]
as
begin
SET NOCOUNT ON
update tb_Resultadolinea
set iRespuesta = null

Declare @Respuesta int

exec SP_CalHoraPico

select @Respuesta = iRespuesta from tb_Resultadolinea where iCodigoNodo =
1

if @Respuesta =1
begin

exec SP_CalVentaMenorPromedio
select @Respuesta = iRespuesta from tb_Resultadolinea where
iCodigoNodo = 2
```

```

        if @Respuesta =1
        begin
            exec SP_CalMayorVentas5Productos
            select @Respuesta = iRespuesta from tb_Resultadolinea where
iCodigoNodo = 7

            if @Respuesta =1
            begin
                exec SP_CalVentaMayorProRelacionados
                select @Respuesta = iRespuesta from tb_Resultadolinea
where iCodigoNodo = 10

                if @Respuesta =1
                begin
                    end
                    exec SP_17ProductoMalo
                    end

                else
                begin
                    end
                    exec SP_18ProductoAcompañante
                    end

                end

            else
            begin
                insert into tb_LogSistemaExperto
(vNombre,iCategoria,iEstado,dCreacion) values ('Nada',2,1,getdate())
                end

            end

        else
        begin
            exec SP_CalMayorMesasOcupadasMayorVentas
            select @Respuesta = iRespuesta from tb_Resultadolinea where
iCodigoNodo = 6
            if @Respuesta =1
            begin

                exec SP_CalMayorVentasPromedio
                select @Respuesta = iRespuesta from
tb_Resultadolinea where iCodigoNodo = 9
                if @Respuesta =1
                begin
                    exec SP_15RegalarProducto
                    end

                else
                begin
                    insert into tb_LogSistemaExperto
(vNombre,iCategoria,iEstado,dCreacion) values ('Nada',2,1,getdate())
                    end

                end

            else
            begin
                exec SP_18Postres
                end

            end

        end

    end
end

```

```

else
begin
    exec SP_CalVentaMayorPromedioNoPico
    select @Respuesta = iRespuesta from tb_Resultadolinea where
iCodigoNodo = 3
        if @Respuesta =1
            begin
                exec SP_CalNumeroMayorVentasPlatoPrinicpal
                select @Respuesta = iRespuesta from
tb_Resultadolinea where iCodigoNodo = 5
                    if @Respuesta =1
                        begin
                            exec SP_14EspecialidadChef
                            end
                        else
                            begin
                                exec SP_13ProductoPrincipal
                                end
                            end
                    end
                else
                    begin
                        exec SP_CalMAyorMesasOcupadas
                        select @Respuesta = iRespuesta from
tb_Resultadolinea where iCodigoNodo = 5
                            if @Respuesta =1
                                begin
                                    exec SP_12ProductoBebida
                                    end
                                else
                                    begin
                                        exec SP_TodasMesasLibres
                                        select @Respuesta = iRespuesta
from tb_Resultadolinea where iCodigoNodo = 8
                                            if @Respuesta =1
                                                begin
                                                    exec SP_CierreJob
                                                    end
                                                else
                                                    begin
                                                        insert into
tb_LogSistemaExperto (vNombre,iCategoria,iEstado,dCreacion) values
('Nada',2,1,getdate())
                                                        end
                                                    end
                                                end
                                            end
                                        end
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
end

```

Los procedimientos para la toma de decisiones son las siguientes.

- **Procedimiento para calcular la hora de mayoría de ventas.**

```

Create procedure [dbo].[SP_CalHoraPico]
as
begin
Declare @Hora int
Declare @Indicador int

```

```
select @Hora = datepart(hour,getdate() )

select @Indicador = iHoraPico
  from tb_Hora
where iHoraInicio = @Hora
     or iHoraFin = @Hora

  update tb_ResultadoLinea
  set irespuesta = @Indicador
  where iCodigoNodo = 1
end
```

- ¿Las ventas son menor al promedio?

```
Create Procedure [dbo].[SP_CalVentaMenorPromedio]
as
begin
Declare @HoraAnterior int
Declare @Promedio int
Declare @PedidoActual int
select @HoraAnterior = case when datepart(hour,getdate()) = 0 then 23
else datepart(hour,getdate())-1 end

select cast(dCreacion as date) as dia,
       sum(iCantidad ) as Cantidad
  into #tbTemporal
  from tb_Pedido
where datepart(WEEKDAY,dCreacion) = datepart(WEEKDAY,GETDATE())
     and datepart(hour,dCreacion) = @HoraAnterior
     and dCreacion < getdate()
group by cast(dCreacion as date)

select @Promedio = avg(Cantidad)
from #tbTemporal

Select @PedidoActual = Count(*) from tb_Pedido
where cast(dCreacion as date) = cast(getdate() as date)
     and datepart(hour,dCreacion)-1 = @HoraAnterior

if @Promedio > @PedidoActual
begin
  update tb_ResultadoLinea
  set iRespuesta = 1
  where iCodigoNodo = 2
end
else
begin
  update tb_ResultadoLinea
  set iRespuesta = 0
  where iCodigoNodo = 2
end
drop table #tbTemporal

end
```

- ¿Las ventas de los 5 productos más vendidos son mayor al promedio?

```

Create procedure [dbo].[SP_CalMayorVentas5Productos]
as
begin
Declare @Hora int
Declare @CantidadA int
Declare @CantidadP int

select @Hora =case when datepart(hour,getdate()) = 0 then 23 else
datepart(hour,getdate()) -1 end

select cast(ped.dCreacion as date) as fecha,
ped.iCodigoPlato,
sum(iCantidad) as Cantidad
into #temp
from tb_Pedido ped
inner join tb_Platos pla
on pla.iCodigoPlato = ped.iCodigoPlato
where pla.iCategoriaPlato not in (9,10)

group by ped.iCodigoPlato,cast(ped.dCreacion as date)
order by sum(iCantidad) desc

select top 5 iCodigoPlato
into #PlatosPrincipales
from #temp
group by iCodigoPlato
order by avg(Cantidad) desc

select top 5 @CantidadP=avg(Cantidad)
from #temp
group by iCodigoPlato
order by avg(Cantidad) desc

select @CantidadA = sum(iCantidad) from tb_Pedido
where cast(dCreacion as date) = cast(getdate() as date)
and iCodigoPlato in (
select * from #PlatosPrincipales
)

if @CantidadA > @CantidadP
begin
update tb_ResultadoLinea
set iRespuesta = 1
where iCodigoNodo = 7
end
else
begin
update tb_ResultadoLinea
set iRespuesta = 0
where iCodigoNodo = 7
end

drop table #temp
drop table #PlatosPrincipales

end
    
```

- ¿Es mayor la venta de los productos relacionados de los productos más vendidos?

```

Create Procedure [dbo].[SP_CalVentaMayorProRelacionados]
as
begin
Declare @Hora int
Declare @MontoP decimal (10,2)
Declare @MontoA decimal (10,2)
Select @Hora = case when datepart(hour,getdate()) = 0 then 23 else
datepart(hour,getdate())-1 end

select top 5
iCodigoPlato,
sum(iCantidad ) as Cantidad
into #temp
from tb_Pedido
group by iCodigoPlato
order by 2 desc

select
rank() over (partition by iplato1 order by avg(iCantidad),iplato2) as
rango,
iplato1,iplato2,avg(iCantidad) as Cantidad
into #PlatoRelacionado
from tb_Cantidades can
inner join tb_Platos pla
on pla.iCodigoPlato = can.iplato2
where iplato1 in (select iCodigoPlato from #temp)
and pla.iCategoriaPlato not in (9,10)
group by iplato1,iplato2

delete #PlatoRelacionado
where rango <>1

select @MontoP = avg(Cantidad) from (
select sum(iCantidad) as Cantidad from tb_Pedido
where iCodigoPlato in (
select iplato2
from #PlatoRelacionado
)
and datepart(hour,dCreacion) =@Hora
group by cast(dCreacion as date)
) a

select @MontoA = sum(iCantidad) from tb_Pedido
where iCodigoPlato in (
select iplato2
from #PlatoRelacionado
)
and Cast(dCreacion as date) = cast(getdate() as date)
and datepart(hour,dCreacion) = @Hora

if @MontoA > @MontoP
begin
update tb_ResultadoLinea
set iRespuesta =1

```

```

        where iCodigoNodo = 10
    end

    else
    begin
        update tb_ResultadoLinea
        set iRespuesta =0
        where iCodigoNodo = 10
    end
    drop table #temp
    drop table #PlatoRelacionado

    end

```

- ¿Las mesas ocupadas son mayor al promedio en dicho

periodo?

```

Create Procedure [dbo].[SP_CalMAYorMesasOcupadasMayorVentas]
as
begin

Declare @Hora int
Declare @Promedio int
Declare @Actual int
select @Hora=datepart(hour,getdate())

select cast(dCreacion as date) as Dia,
        count(distinct fac.iCodigoFactura) as Cantidad
        into #Temporal
        from tb_Pedido ped
left join tb_DetalleFactura det
on ped.iCodigoPedido = det.iPedido
left join tb_Factura fac
on fac.iCodigoFactura = det.iCodigoFactura
where datepart(hour,ped.dCreacion) < @Hora
and datepart(hour,fac.dFecha) > @Hora
group by cast(dCreacion as date)
order by 1 desc

select @Promedio=avg(Cantidad)
from #Temporal

select @Actual = count(distinct iCodigoMesa)
from tb_Pedido ped
left join tb_DetalleFactura fac
on ped.iCodigoPedido = fac.iPedido
where cast(ped.dCreacion as date) = cast(getdate() as date)
and fac.iCodigoFactura = null

if @Actual > @Promedio
begin
update tb_ResultadoLinea
set iRespuesta = 1
where iCodigoNodo =6
end
else
begin
update tb_ResultadoLinea
set iRespuesta = 0
where iCodigoNodo =6
end

```

end

drop table #Temporal

end

- **¿Monto de ventas mayor al promedio?**

```
Create procedure [dbo].[SP_CalMayorVentasPromedio]
as
begin
```

```
Declare @Hora int
```

```
Declare @MontoP decimal (10,2)
```

```
Declare @MontoA decimal (10,2)
```

```
Select @Hora = case when datepart(hour,getdate()) = 0 then 23 else
datepart(hour,getdate())-1 end
```

```
select cast(ped.dCreacion as date) as dia,
sum(pla.fCosto*ped.iCantidad) as Monto
into #temp
from tb_Pedido ped
inner join tb_Platos pla
on ped.iCodigoPlato = pla.iCodigoPlato
where datepart(hour,ped.dCreacion) = @Hora
group by cast(ped.dCreacion as date)
```

```
select @MontoP = avg(Monto) from #temp
```

```
select @MontoA = sum(pla.fCosto*ped.iCantidad)
from tb_Pedido ped
inner join tb_Platos pla
on ped.iCodigoPlato = pla.iCodigoPlato
where datepart(hour,ped.dCreacion) = @Hora
and cast(ped.dCreacion as date) = cast(getdate() as date)
```

```
if @MontoA > @MontoP
begin
update tb_ResultadoLinea
set iRespuesta = 1
where iCodigoNodo = 9
end
```

```
else
begin
update tb_ResultadoLinea
set iRespuesta = 0
where iCodigoNodo = 9
end
```

```
drop table #temp
end
```

- **¿Las ventas son mayor al promedio de ventas en el periodo**

de menos ventas?

```
Create Procedure [dbo].[SP_CalVentaMayorPromedioNoPico]
as
begin
Declare @HoraAnterior int
Declare @Promedio int
```

```
Declare @PedidoActual int
select @HoraAnterior = case when datepart(hour,getdate()) = 0 then 23
else datepart(hour,getdate())-1 end
```

```
select cast(dCreacion as date) as dia,
       sum(iCantidad ) as Cantidad
into #tbTemporal
from tb_Pedido
where datepart(WEEKDAY,dCreacion) = datepart(WEEKDAY,GETDATE())
and datepart(hour,dCreacion) = @HoraAnterior
and dCreacion< getdate()
group by cast(dCreacion as date)
```

```
select @Promedio = avg(Cantidad)
from #tbTemporal
```

```
Select @PedidoActual = Count(*) from tb_Pedido
where cast(dCreacion as date) = cast(getdate() as date)
and datepart(hour,dCreacion)-1 = @HoraAnterior
```

```
if @Promedio<@PedidoActual
begin
update tb_ResultadoLinea
set iRespuesta = 1
where iCodigoNodo = 3
```

```
end
else
begin
update tb_ResultadoLinea
set iRespuesta = 0
where iCodigoNodo = 3
end
drop table #tbTemporal
```

end

- ¿Número de ventas mayor del promedio del plato más vendido?

```
Create procedure [dbo].[SP_CalNumeroMayorVentasPlatoPrinicpal]
as
begin
Declare @PlatoPrincipal int
Declare @Hora int
Declare @PromedioVentas int
Declare @VentasActual int
```

```
Select @Hora = case when datepart(hour,getdate()) = 0 then 23 else
datepart(hour,getdate()) -1 end
select @PlatoPrincipal = (select top 1 ped.iCodigoPlato as cantidad
from tb_Pedido ped
inner join tb_Platos pla
on ped.iCodigoPlato = pla.iCodigoPlato
inner join tb_CategoriaPlato cat
on cat.iCategoriaPlato = pla.iCategoriaPlato
```

```

where cat.iCategoriaPlato not in (9,10)
group by ped.iCodigoPlato
order by count(*) desc
)

```

```

select cast(dCreacion as date) as fecha,
sum(iCantidad) as Cantidad
into #Temporal
from tb_Pedido
where datepart(hour,dCreacion) = @Hora
and iCodigoPlato = @PlatoPrincipal
group by cast(dCreacion as date)

```

```

select @PromedioVentas = avg(Cantidad)
from #Temporal

```

```

select @VentasActual=sum(iCantidad) from tb_Pedido
where datepart(hour,dCreacion) = @Hora
and iCodigoPlato = @PlatoPrincipal
and cast(dCreacion as date) = cast(getdate() as date)

```

```

if @VentasActual > @PromedioVentas
begin
update tb_ResultadoLinea
set iRespuesta = 1
where iCodigoNodo = 5
end
else
begin
update tb_ResultadoLinea
set iRespuesta = 0
where iCodigoNodo = 5
end

```

```
drop table #Temporal
```

```
end
```

- ¿Mesas ocupadas mayor al promedio en horas de no mayoría de ventas?

```

Create Procedure [dbo].[SP_CalMayorMesasOcupadas]
as
begin

```

```

Declare @Hora int
Declare @Promedio int
Declare @Actual int
select @Hora=datepart(hour,getdate())

```

```

select cast(dCreacion as date) as Dia,
count(distinct fac.iCodigoFactura) as Cantidad
into #Temporal
from tb_Pedido ped
left join tb_DetalleFactura det
on ped.iCodigoPedido = det.iPedido
left join tb_Factura fac
on fac.iCodigoFactura = det.iCodigoFactura
where datepart(hour,ped.dCreacion) < @Hora

```

```

and datepart(hour,fac.dFecha) > @Hora
group by cast(dCreacion as date)
order by 1 desc

select @Promedio=avg(Cantidad)
from #Temporal

select @Actual = count(distinct iCodigoMesa)
from tb_Pedido ped
left join tb_DetalleFactura fac
on ped.iCodigoPedido = fac.iPedido
where cast(ped.dCreacion as date) = cast(getdate() as date)
and fac.iCodigoFactura = null

if @Actual > @Promedio
begin
update tb_ResultadoLinea
set iRespuesta = 1
where iCodigoNodo =4
end
else
begin
update tb_ResultadoLinea
set iRespuesta = 0
where iCodigoNodo =4
end

drop table #Temporal
end

```

- ¿Todas las mesas libres?

```

Create Procedure [dbo].[SP_TodasMesasLibres]
as
begin
Declare @Pedido int

select @Pedido = count(*) from tb_Pedido
where iCodigoPedido not in (
select iPedido from tb_DetalleFactura
)

if @Pedido = 0
begin
update tb_ResultadoLinea
set iRespuesta =1
where iCodigoNodo = 8
end

else

begin
update tb_ResultadoLinea
set iRespuesta =0
where iCodigoNodo = 8
end
end
end

```

Luego del análisis nos dará nodos finales, las que tomarán las decisiones sobre el sistema.

- **Promoción del producto menos vendido**

```
Create procedure [dbo].[SP_17ProductoMalo]
as
begin
Declare @Hora int
Declare @Horai varchar(19)
Declare @Horaf varchar(19)
Declare @NombrePlato varchar(100)
Declare @Plato int
Declare @Promocion int

select @Hora=datepart(hour,getdate())

select @Horai =case when iHoraInicio = @Hora
then
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHorainicio<10 then
'0'+cast(iHorainicio as char(1)) else cast(iHoraInicio as varchar(2))
end +':00:00'
else
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10 then
'0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2)) end
+':00:00'
end ,
@Horaf = Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10
then '0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2))
end +':59:59'

from tb_Hora
where iHoraInicio = @Hora or
iHoraFin = @Hora

select @Plato = (
select top 1 iCodigoPlato
from tb_Pedido
group by iCodigoPlato
order by sum(icantidad) asc)

select @NombrePlato = vNombrePlato from tb_Platos where iCodigoPlato =
@Plato
select @NombrePlato = @NombrePlato + ' '+@Horai

exec SP_CrearPromocion @Nombre=@NombrePlato
,@Tipo=2,@Monto=20,@Usuario=N'Admin',@FechaI=@Horai,@FechaF=@Horaf
select @Promocion = max(icodigoPromocion) from tb_Promocion
exec SP_AgregarPromoPlato
@Promo=@Promocion,@Plato=@Plato,@Usuario=N'Admin'
insert into tb_LogSistemaExperto
(vNombre,iCategoria,iEstado,dCreacion) values
(@NombrePlato,1,1,getdate())
end
```

- Promoción del acompañante del mejor producto

```

Create procedure [dbo].[SP_18ProductoAcompañante]
as
begin
Declare @Hora int
Declare @Horai varchar(19)
Declare @Horaf varchar(19)
Declare @NombrePlato varchar(100)
Declare @Plato int
Declare @Promocion int

select @Hora=datepart(hour,getdate())

select @Horai =case when iHoraInicio = @Hora
then
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraInicio<10 then
'0'+cast(iHoraInicio as char(1)) else cast(iHoraInicio as varchar(2))
end +':00:00'
else
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10 then
'0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2)) end
+':00:00'
end ,
@Horaf = Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10
then '0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2))
end +':59:59'
from tb_Hora
where iHoraInicio = @Hora or
iHoraFin = @Hora

select top 5
iCodigoPlato,
sum(iCantidad ) as Cantidad
into #temp
from tb_Pedido
group by iCodigoPlato
order by 2 desc

select
rank() over (partition by iplato1 order by avg(iCantidad),iplato2) as
rango,
iplato1,iplato2,avg(iCantidad) as Cantidad
into #PlatoRelacionado
from tb_Cantidades can
inner join tb_Platos pla
on pla.iCodigoPlato = can.iplato2
where iplato1 in (select iCodigoPlato from #temp)
and pla.iCategoriaPlato not in (9,10)
group by iplato1,iplato2

select @Plato = (
select top 1 iplato2 from #PlatoRelacionado
group by iplato2
order by sum(cantidad) desc
)

select @NombrePlato = vNombrePlato from tb_Platos where iCodigoPlato =
@Plato
select @NombrePlato = @NombrePlato + ' '+@Horai

```

```
exec SP_CrearPromocion @Nombre=@NombrePlato
,@Tipo=2,@Monto=20,@Usuario=N'Admin',@FechaI=@Horai,@FechaF=@Horaf
select @Promocion = max(icodigoPromocion) from tb_Promocion
exec SP_AgregarPromoPlato
@Promo=@Promocion,@Plato=@Plato,@Usuario=N'Admin'
insert into tb_LogSistemaExperto (vNombre,iCategoria,iEstado,dCreacion)
values (@NombrePlato,1,1,getdate())
end
```

- **Regalar producto**

```
Create Procedure [dbo].[SP_15RegalarProducto]
as
begin
Declare @Nombre varchar(100)

Declare @Mesa int

select @Mesa = (
select top 1 iCodigoMesa from tb_Pedido ped
left join tb_DetalleFactura fac
on ped.iCodigoPlato = fac.iCodigoFactura
where fac.iCodigoFactura is null
order by NEWID())

update tb_Almacen
set iCantidad = iCantidad -1
where iCodigoIngrediente = 18

set @Nombre = 'Regalar a Mesa: ' + cast(@Mesa as varchar(3))+' 1
Gaseosas'

insert into tb_LogSistemaExperto (vNombre,iCategoria,iEstado,dCreacion)
values (@Nombre,2,1,getdate())

end
```

- **Promoción de un postre aleatorio**

```
Create procedure [dbo].[SP_18Postres]
as
begin
Declare @Hora int
Declare @Horai varchar(19)
Declare @Horaf varchar(19)
Declare @NombrePlato varchar(100)
Declare @Plato int
Declare @Promocion int

select @Hora=datepart(hour,getdate())
select @Horai =case when iHoraInicio = @Hora
then
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHorainicio<10 then
'0'+cast(iHorainicio as char(1)) else cast(iHoraInicio as varchar(2))
end +':00:00'
else
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10 then
'0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2)) end
+':00:00'
end ,
@Horaf = Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10
then '0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2))
end +':59:59'
```

```

from tb_Hora
where iHoraInicio = @Hora or
      iHoraFin = @Hora

      select @Plato = (
select top 1 iCodigoPlato
from tb_Platos
where iCategoriaPlato = 9
ORDER BY NEWID()
)

select @Horai,@Horaf
select @NombrePlato = vNombrePlato from tb_Platos where iCodigoPlato =
@Plato
select @NombrePlato = @NombrePlato + ' '+@Horai
select @Plato
exec SP_CrearPromocion @Nombre=@NombrePlato
,@Tipo=2,@Monto=20,@Usuario=N'Admin',@FechaI=@Horai,@FechaF=@Horaf
select @Promocion = max(iCodigoPromocion) from tb_Promocion
exec SP_AgregarPromoPlato
@Promo=@Promocion,@Plato=@Plato,@Usuario=N'Admin'
insert into tb_LogSistemaExperto (vNombre,iCategoria,iEstado,dCreacion)
values (@NombrePlato,1,1,getdate())
end

```

- **Promoción aleatoria de 1 producto de la categoría
especialidad del chef**

```

Create procedure [dbo].[SP_14EspecialidadChef]
as
begin
Declare @Hora int
Declare @Horai varchar(19)
Declare @Horaf varchar(19)
Declare @NombrePlato varchar(100)
Declare @Plato int
Declare @Promocion int

select @Hora=datepart(hour,getdate())
select @Horai =case when iHoraInicio = @Hora
then
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHorainicio<10 then
'0'+cast(iHorainicio as char(1)) else cast(iHoraInicio as varchar(2))
end +':00:00'
else
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10 then
'0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2)) end
+':00:00'
end ,
@Horaf = Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10
then '0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2))
end +':59:59'
from tb_Hora
where iHoraInicio = @Hora or
      iHoraFin = @Hora
select @Plato = (
select top 1 iCodigoPlato
from tb_Platos
where iCategoriaPlato = 6
ORDER BY NEWID()
)

```

```

select @Horai,@Horaf
select @NombrePlato = vNombrePlato from tb_Platos where iCodigoPlato =
@Plato
select @NombrePlato = @NombrePlato + ' '+@Horai
select @Plato
exec SP_CrearPromocion @Nombre=@NombrePlato
,@Tipo=2,@Monto=20,@Usuario=N'Admin',@FechaI=@Horai,@FechaF=@Horaf
select @Promocion = max(icodigoPromocion) from tb_Promocion
exec SP_AgregarPromoPlato
@Promo=@Promocion,@Plato=@Plato,@Usuario=N'Admin'
insert into tb_LogSistemaExperto (vNombre,iCategoria,iEstado,dCreacion)
values (@NombrePlato,1,1,getdate())
end

```

- **Promoción de plato principal**

```

Create procedure [dbo].[SP_13ProductoPrincipal]
as
begin
Declare @Hora int
Declare @Horai varchar(19)
Declare @Horaf varchar(19)
Declare @NombrePlato varchar(100)
Declare @Plato int
Declare @Promocion int

select @Hora=datepart(hour,getdate())
select @Horai =case when iHoraInicio = @Hora
then
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHorainicio<10 then
'0'+cast(iHorainicio as char(1)) else cast(iHoraInicio as varchar(2))
end +':00:00'
else
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10 then
'0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2)) end
+':00:00'
end ,
@Horaf = Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10
then '0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2))
end +':59:59'
from tb_Hora
where iHoraInicio = @Hora or
iHoraFin = @Hora
select @Plato = (
select top 1 ped.iCodigoPlato
from tb_Pedido ped
inner join tb_Platos pla
on ped.iCodigoPlato = pla.iCodigoPlato
where pla.iCategoriaPlato not in (9,10)
group by ped.iCodigoPlato
order by sum(icantidad) desc)

select @NombrePlato = vNombrePlato from tb_Platos where iCodigoPlato =
@Plato
select @NombrePlato = @NombrePlato + ' '+@Horai
exec SP_CrearPromocion @Nombre=@NombrePlato
,@Tipo=2,@Monto=20,@Usuario=N'Admin',@FechaI=@Horai,@FechaF=@Horaf
select @Promocion = max(icodigoPromocion) from tb_Promocion
exec SP_AgregarPromoPlato
@Promo=@Promocion,@Plato=@Plato,@Usuario=N'Admin'
insert into tb_LogSistemaExperto (vNombre,iCategoria,iEstado,dCreacion)
values (@NombrePlato,1,1,getdate())end

```

- **Promoción de bebidas alcohólicas**

```

Create procedure [dbo].[SP_12ProductoBebida]
as
begin
Declare @Hora int
Declare @Horai varchar(19)
Declare @Horaf varchar(19)
Declare @NombrePlato varchar(100)
Declare @Plato int
Declare @Promocion int

select @Hora=datepart(hour,getdate())
select @Horai =case when iHoraInicio = @Hora
then
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHorainicio<10 then
'0'+cast(iHorainicio as char(1)) else cast(iHoraInicio as varchar(2))
end +':00:00'
else
Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10 then
'0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2)) end
+':00:00'
end ,
@Horaf = Format(GetDate(), N'yyyy-MM-dd')+ 'T'+case when iHoraFin<10
then '0'+cast(iHoraFin as char(1)) else cast(iHoraFin as varchar(2))
end +':59:59'
from tb_Hora
where iHoraInicio = @Hora or
iHoraFin = @Hora
select @Plato = (
select top 1 iCodigoPlato
from tb_Platos
where iCategoriaPlato = 10
ORDER BY NEWID()
)
select @Horai,@Horaf
select @NombrePlato = vNombrePlato from tb_Platos where iCodigoPlato =
@Plato
select @NombrePlato = @NombrePlato + ' '+@Horai
select @Plato
exec SP_CrearPromocion @Nombre=@NombrePlato
,@Tipo=2,@Monto=20,@Usuario=N'Admin',@FechaI=@Horai,@FechaF=@Horaf
select @Promocion = max(icodigoPromocion) from tb_Promocion
exec SP_AgregarPromoPlato
@Promo=@Promocion,@Plato=@Plato,@Usuario=N'Admin'
insert into tb_LogSistemaExperto (vNombre,iCategoria,iEstado,dCreacion)
values (@NombrePlato,1,1,GETDATE())
end
    
```

- **Cierre del sistema**

El procedimiento cerrará el sistema el cuál realizará los siguientes pasos

- Población del data warehouse.
- Cambio de la fecha del sistema.
- Cálculo de la relación de ventas de productos.



Anexo C

En el anexo se encontrará el manual de usuario del sistema experto:

1. Pantalla de inicio del Sistema “Mi Restaurante Experto”

En la figura se puede observar la pantalla principal o de inicio de la aplicación.



2. Login del sistema

En la figura, se muestra el ingreso o logeo en el sistema, para ingresar a dicho módulo se deberá realizar click en el botón iniciar sesión.



3. Módulo de usuarios

El módulo de usuarios tiene la funcionalidad de agregar, actualizar y eliminar a los usuarios del sistema.

El módulo permite la opción de búsqueda a través del botón que está representado de una lupa.

Para poder agregar al usuario se tendrá que insertar los campos de usuario, la contraseña, el nombre, apellido paterno, apellido materno, el teléfono y/o celular, el rol, el tipo y número de documento, el estado y el ubigeo del usuario elegido.

En el campo de roles se desplegará una lista de los roles creados en el sistema.

El campo de ubigeo se carga de forma de automática para que sea posible el llenado de los campos de departamento, provincia y distrito.

En la figura, se puede observar el módulo de registro de usuarios, luego de realizar click en el botón agregar usuario.

Código Usuario	Nombre Usuario	Apellido Paterno	Apellido Materno	Teléfono	Celular	Rol	Documento	Departa	Provincia	Distrito	Estado	Editar	Eliminar
000001	ALFONSO	CASTAÑO	LOPEZ	946217575	9555894430	Administrador de Sistema	71450700	Ancash	Ancash	La Joya	Activo	Editar	Eliminar
000002	CARLOS	RIVERA	MOSQUERA	221040	90405213	Personal de Atención	23055-12	Ancash	Ancash	Ancash	Activo	Editar	Eliminar
000003	PORCEY	CHALCO	BLANCO	205514	955599995	Personal de Atención	4510009	Ancash	Ancash	San Antonio	Activo	Editar	Eliminar
000004	JIMMY	SOLÍS	LOPEZ	205514	955599995	Personal de Atención	45024979	Ancash	Ancash	Saizaca	Activo	Editar	Eliminar

4. Módulo de mantenimiento de roles

El módulo de mantenimiento de roles permite la creación de los roles del sistema.

Para crear un nuevo rol se tiene que ingresar el nombre del rol, teniendo la validación del sistema que no se repita el mismo nombre.

Cuando se hace click en el botón editar, se carga el nombre del rol en el textbox y cambian los botones por los de editar y cancelar.

En la primera figura, se observa la pantalla del módulo de mantenimiento de roles al inicio del ingreso al módulo, mientras que en la segunda figura, se observar luego de realizar click en el botón editar.



Codigo Rol	Nombre Rol	Fecha Modificación	Editar
4	Encargado de Ventas	17/03/2017	Editar
3	Personal de Atención	8/10/2016	Editar
2	Cajero	8/10/2016	Editar
1	Administrador	16/06/2016	Editar



Codigo Rol	Nombre Rol	Fecha Modificación	Editar
4	Encargado de Ventas	17/03/2017	Editar
3	Personal de Atención	8/10/2016	Editar
2	Cajero	8/10/2016	Editar
1	Administrador	10/03/2016	Editar

5. Módulo de asignar actividades a rol

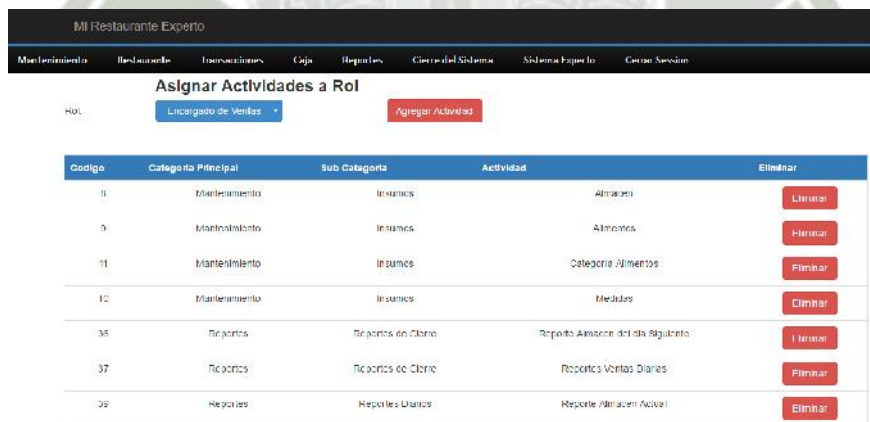
El módulo de asignar actividades a rol permite la asignación de actividades u opciones del menú al rol seleccionado.

El listbox del módulo se carga con los roles del sistema y se muestra en la grilla las actividades que le pertenecen al rol en selección.

Cuando se hace click en el botón agregar actividad, aparece un nuevo listbox con las actividades que el rol no cuenta para que puedan agregarse al rol.

En la grilla se tendrá el botón eliminar el cual elimina la actividad del rol asignado.

En la primera figura, se observa el módulo de asignar actividades a un rol al inicio, mientras que en la segunda figura, se observa luego de realizar click en el botón agregar actividad.



Codigo	Categoria Principal	Sub Categoría	Actividad	Eliminar
8	Mantenimiento	Insumos	Alimentos	Eliminar
9	Mantenimiento	Insumos	Alimentos	Eliminar
11	Mantenimiento	Insumos	Categoría Alimentos	Eliminar
10	Mantenimiento	Insumos	Medidas	Eliminar
38	Reportes	Reportes de Clientes	Reporte Análisis del día siguiente	Eliminar
37	Reportes	Reportes de Clientes	Reportes Ventas Diarias	Eliminar
39	Reportes	Reportes Diarios	Reporte Análisis Actual	Eliminar

Mi Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Asignar Actividades a Rol

Rol:

Actividad:

Código	Categoría Principal	Sub Categoría	Actividad	Eliminar
8	Mantenimiento	Insumos	Almacén	<input type="button" value="Eliminar"/>
9	Mantenimiento	Insumos	Almacén	<input type="button" value="Eliminar"/>
18	Mantenimiento	Insumos	Categoría Almacén	<input type="button" value="Eliminar"/>
19	Mantenimiento	Insumos	Modas	<input type="button" value="Eliminar"/>
36	Reportes	Reportes de Cliente	Reporte Almacén del día Siguiente	<input type="button" value="Eliminar"/>
37	Reportes	Reportes de Cliente	Reportes Ventas Diarias	<input type="button" value="Eliminar"/>
39	Reportes	Reportes Diarios	Reporte Almacén Actual	<input type="button" value="Eliminar"/>

6. Módulo de mantenimiento de almacén

Cómo se observa en la figura 86, el módulo de mantenimiento de almacén permite la actualización del mismo, en ella se actualiza los campos de cantidad que tiene el almacén, además se ingresa la cantidad máxima permitida por cada tipo de ingrediente y el tiempo de caducidad del mismo.

En la grilla se muestra el nombre de la categoría del ingrediente, el nombre del ingrediente, la cantidad, el monto que se calcula de la multiplicación de la cantidad por el costo del ingrediente, la cantidad máxima, el tiempo de caducidad y la cantidad mínima que se calcula de los cantidad promedio de ventas diarias y el tiempo de caducidad del ingrediente seleccionado.

En cada registro de la grilla se encuentra el botón editar el cual permite modificar la cantidad, la cantidad máxima y el tiempo de caducidad del ingrediente seleccionado.

Además el módulo permite la opción de búsqueda por ingrediente por medio del botón que tiene una representación de lupa.

Código Almacén	Nombre Categoría	Nombre Ingrediente	Cantidad	Unidad	Cantidad Máxima	Tiempo Caducidad	Cantidad Mínimo	Opciones
0	Alimentos	Aceite	50.75	litros	50.00	30.00	1.00	Editar
1	Alimentos	Harina	11.00	kg	10.00	3.00	0.00	Editar
2	Alimentos	Leche	20.75	litros	20.00	10.00	2.00	Editar
3	Alimentos	Carne	50.75	kg	50.00	3.00	1.00	Editar
4	Alimentos	Sal	10.75	kg	10.00	3.00	0.00	Editar
5	Alimentos	Azúcar	10.75	kg	10.00	3.00	0.00	Editar

7. Módulo de mantenimiento de alimentos

Cómo se observa en la figura, el módulo de mantenimiento de alimentos permite la creación y actualización de nuevos ingredientes.

El módulo presenta cuatro campos los cuales son un listbox que despliega una lista de las categorías de alimentos, un campo del nombre de alimento, un listbox que carga una lista de las medidas del sistema y el precio que se coloca por unidad de medida del ingrediente.

En el módulo mostrará una grilla que se actualizará de acuerdo a la categoría del ingrediente, en dicha grilla se mostrarán el nombre de la categoría, la medida, el nombre del ingrediente, el costo y un botón editar que permitirá la actualización de los atributos del ingrediente.



Código Alimento	Nombre Categoría	Medida	Alimento	Costo	Editar
1	Aves	Pieza	Pierna de Pollo	3.50	Editar
2	Aves	Kilogramos	Pechuga de Pollo	12.00	Editar
3	Aves	Kilogramos	Pechuga de Pavo	8.00	Editar
21	Aves	Kilogramos	Gallina	9.00	Editar

9. Módulo de mantenimiento de medidas

El módulo de mantenimiento de medidas permite la creación y actualización de las medidas de los ingredientes del sistema.

En el módulo se tendrán dos textbox los cuales son, el nombre de la medida y la abreviación de la misma.

En el módulo aparecerá una grilla la cual mostrará los campos de código de medida, el nombre de la medida, la abreviatura de la medida y un botón de editar el cual permitirá la modificación de la misma.

Cuando se realice click en el botón editar este cargará los datos a los campos del módulo y aparecerán dos nuevos botones los cuales serán editar y cancelar.

En la primera figura, se observa el módulo del mantenimiento de medidas al ingresar, mientras que en la segunda figura, se observa luego de realizar click en el botón editar.

Mi Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Mantenimiento de Medidas

Medida:

Abreviatura:

Código Medida	Nombre Medida	Abreviatura	Editar
1	Kilogramos	Kg	<input type="button" value="Editar"/>
2	Litros	L	<input type="button" value="Editar"/>
3	Pieza	Pz	<input type="button" value="Editar"/>
4	Bolsa	Bt	<input type="button" value="Editar"/>

Mi Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Mantenimiento de Medidas

Medida:

Abreviatura:

Código Medida	Nombre Medida	Abreviatura	Editar
1	Kilogramos	Kg	<input type="button" value="Editar"/>
2	Litros	L	<input type="button" value="Editar"/>
3	Pieza	Pz	<input type="button" value="Editar"/>
4	Bolsa	Bt	<input type="button" value="Editar"/>

10. Módulo de mantenimiento de categoría ingrediente

En el módulo de mantenimiento de categoría ingrediente se podrá crear y actualizar las categorías de ingredientes.

En el módulo tiene un checkbox para llenar el nombre de la categoría, y una grilla que mostrará los campos de código de categoría,

nombre de categoría y un botón de editar el cual al hacer click mostrará el nombre de la categoría en el textbox, para su edición.

Cuando se agregue o actualice una categoría el sistema validará que no exista una con el mismo nombre.

En la primera figura, se observa el módulo de mantenimiento de categoría ingrediente, mientras que en la segunda figura, se observa luego de realizar click en el botón editar.




11. Módulo de mantenimiento de platos

En la figura se observa el módulo de mantenimiento de platos el que permite la creación edición y eliminación de los platos o productos de la empresa.

En este módulo se carga un listbox que contiene información de las categorías de platos, además tiene un textbox que permite el ingreso del nombre de plato, presenta otro listbox que carga el tipo de moneda y por último el un textbox que permite el ingreso del monto del plato. En el módulo se presenta una grilla que muestra el código de plato, la categoría, la moneda, el costo y dos botones uno que permite la edición del plato y el otro la eliminación del mismo.

Al realizar click en el botón editar carga sus características del plato seleccionado en los campos del módulo.



Mantenimiento de Platos

Categoría Plato: Aves

Nombre de Plato:

Moneda: \$/

Monto:

Agregar

Código Plato	Categoría	Plato	Moneda	Costo	Editar	Eliminar
3	Aves	Perú Inga Chucabata	\$/	25.00	Editar	Eliminar
4	Aves	Milanesa de Pollo	\$/	26.00	Editar	Eliminar
5	Aves	Pollo Chucabata	\$/	25.00	Editar	Eliminar
6	Aves	Chicharrón de Pollo	\$/	27.00	Editar	Eliminar
7	Aves	Tufo a la Havanana	\$/	30.00	Editar	Eliminar

13. Módulo de mantenimiento de menús

En la figura, se observa el módulo de mantenimiento de menús el que permite la creación de los nombre de los menús del sistema.

En este módulo aparecen tres campos los cuales son, el nombre del menú, una lista desplegable del tipo de moneda y el monto por el menú.

En el módulo se carga una grilla de los menús creados en el sistema, en ella se carga el código del menú, el nombre del menú, la moneda, el costo y dos botones editar y eliminar.

Cuando se realiza click en el botón editar carga los atributos del menú seleccionado en los campos del módulo.



Código Menú	Menú	Moneda	Costo	Editar	Eliminar
2	Menú infantil	S/.	20.00	Editar	Eliminar
3	Menú 1	S/.	25.00	Editar	Eliminar

14. Mantenimiento de las categorías de platos

El módulo de mantenimiento de las categorías de los platos, permite la creación de nuevas categorías de platos; es decir estas categorías agrupan los platos de acuerdo a ciertas características.

En el módulo se presenta una caja de texto que permite la creación de la categoría y una grilla que lista las categorías de los platos, además en la grilla aparece un botón de editar que permite la actualización de la

categoría, al realizar click se carga el nombre de la categoría al textbox del módulo para su edición.

En la primera figura, se observa el módulo de mantenimiento de categoría de platos al inicio, mientras que en la segunda figura, se observa luego de realizar click en el botón editar.

MI Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Session

Mantenimiento de Categoría Platos

Nombre del Categoría:

[Agregar](#)

Codigo Categoría	Nombre Categoría	Editar
1	Entrada	Editar
2	Piqueos	Editar
3	Caldos	Editar
4	Platos Típicos	Editar
5	Aves	Editar

MI Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Session

Mantenimiento de Categoría Platos

Nombre del Categoría:

[Editar](#) [Cancelar](#)

Codigo Categoría	Nombre Categoría	Editar
1	Entrada	Editar
2	Piqueos	Editar
3	Caldos	Editar
4	Platos Típicos	Editar
5	Aves	Editar
6	Especialidades del Chef	Editar

15. Módulo de ingredientes por plato

En la figura se observa el módulo de ingredientes por plato el que permite el ingreso de la cantidad de insumos para la el plato seleccionado. El módulo en primera instancia carga dos listas las cuales son la categoría del plato y otro que carga los nombres de los platos, estas listas están relacionadas de acuerdo a la categoría del plato.



Luego de hacer click en el botón buscar aparece una lista de las categorías de ingredientes, una lista de los ingredientes, un textbox de la cantidad de ingrediente que se utilizará para el plato, otro textbox que calcula el costo de producción del producto.

En el módulo aparece una grilla que lista los ingredientes que se agregaron a la creación del plato con dos botones uno que permite la edición del ingrediente y un botón eliminar que permite borrar el ingrediente necesario para la creación del plato.

En la figura, se observa el módulo luego de realizar click en el botón buscar.

Mi Restaurante Experto

Maneja el inventario Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Ingredientes por Plato

Categoría del Plato: Nombre del Plato:

Categoría de ingrediente: Ingrediente:

Cantidad: Kg Costo:

Código	Nombre Ingrediente	Medida	Cantidad	Costo	Editar	Eliminar
16	Pechuga de Pollo	Kg	0.30	0.30	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
17	Mozz	Kg	0.10	0.10	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
18	Papa	Kg	0.25	0.63	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
19	Lechuga	Lb	0.10	0.70	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

16. Módulo de asignación de platos al menú

En la figura, se observa el módulo de asignación de platos a menú el que permite la creación y actualización de los platos al menú o combos creados por el restaurante.

En este módulo se visualizan tres campos los cuales son el nombre del menú el cual carga una lista con los menús creados en el sistema, una lista de las categorías de los platos, el cual carga las categorías creados en el sistema y una lista de los platos del restaurante, está lista está relacionada con la categoría seleccionada ya que sólo mostrará los productos de la categoría seleccionada.

Además en el módulo aparece una grilla que lista el código del menú, la categoría del plato, el nombre del plato y un botón eliminar que elimina el plato seleccionado del menú.

MI Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Asignación de Platos a Menú

Nombre del Menú:

Categoría de Plato:

Nombre de Plato:

Codigo	Categoría	Plato	Eliminar
8	Aves	Chicharrón de Pollo	<input type="button" value="Eliminar"/>
9	Bebidas	Gaseosa de 1 Litro	<input type="button" value="Eliminar"/>

17. Módulo de mantenimiento de promociones

En la figura, se observa el módulo de mantenimiento de promociones el que permite la creación de promociones a los platos del restaurante.

El sistema cuenta con dos tipos de descuentos, el primero un descuento por monto y el segundo un descuento por porcentaje.

El módulo cuenta con un tres ingresos, un textbox que permite el ingreso del nombre de la promoción, una lista que permite escoger el tipo de descuento ya sea por monto o por porcentaje y un textbox que permite el ingreso del monto a descontar o el porcentaje de rebaja que tendrá el producto.

El sistema cargará una grilla que mostrará el código de la promoción, el nombre de la promoción, el tipo de descuento y el monto a descontar, también en la grilla se mostrará dos botones; un botón editar

que permite la modificación de la promoción y un botón eliminar que permite la anulación de la misma.

Mi Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Crear Sesión

Crear Promociones

Número Promoción:

Tipo Descuento: Monto

Monto:

Agregar

Código	Nombre Promoción	Tipo	Monto/Porcentaje	Editar	Eliminar
1	Descuento Millanesa	Porcentaje	10.00	Editar	Eliminar
4	Descuento Chuleta	Monto	5.00	Editar	Eliminar
5	Chataron de Choncho	Porcentaje	10.00	Editar	Eliminar
8	Descuento cuy	Monto	10.00	Editar	Eliminar

18. Módulo de asignación de platos a promoción

En la figura, se observa el módulo de asignar platos a una promoción el que permite agregar la promoción al producto seleccionado para poder aplicar el descuento de la misma.

El módulo cuenta con una lista que carga los nombres de las promociones, y otra lista que carga los nombres de los platos.

El módulo carga una grilla que se actualiza de acuerdo se cambie el nombre de la promoción en la lista de los nombres de las promociones.

En la grilla se carga la promoción por plato seleccionado y en ella se muestra los campos de código de asignación de la promoción, el nombre del plato y un botón eliminar que permite la anulación del de plato de la misma.



Codigo	Nombre Plato	Eliminar
10	Milanesa de Pollo	Eliminar

19. Módulo de mantenimiento de pedidos

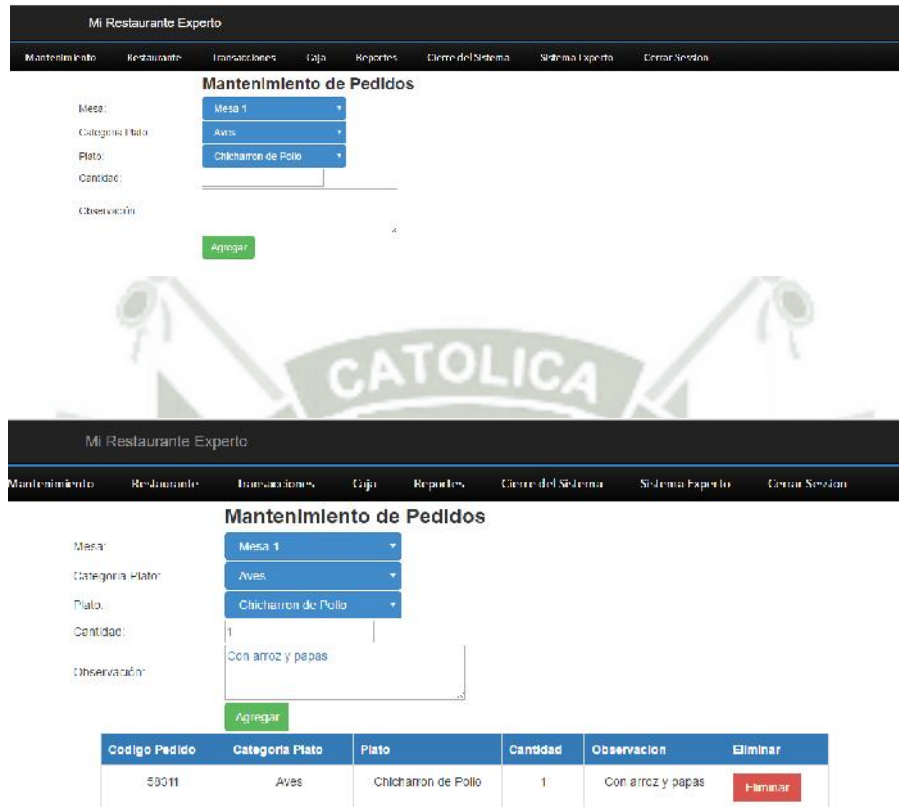
El módulo de mantenimiento de pedidos permite el registro de los pedidos de los clientes, cuando se realiza los pedidos cambia el estado de la mesa a ocupado.

Este módulo debe ser usado en un dispositivo móvil ya que en ese modo da más beneficios al negocio.

El módulo presenta una lista de las mesas, una lista de las categorías de los platos que se relaciona con la lista de los platos o productos del restaurante, además presenta un textbox de la cantidad deseada del producto y una observación del mismo.

Cuando se realiza un pedido se carga una grilla con los datos de los pedidos de la mesa seleccionada.

En la primera figura, se observa el módulo cuando la mesa seleccionada no cuenta con pedidos, mientras en la segunda figura, se observa cuando la mesa si cuentas con pedidos.



Mantenimiento de Pedidos

Mesa:

Categoría Plato:

Plato:

Cantidad:

Observación:

Código Pedido	Categoría Plato	Plato	Cantidad	Observación	Eliminar
50011	Aves	Chicharrón de Pollo	1	Con arroz y papas	<input type="button" value="Eliminar"/>

20. Módulo de mantenimiento de mesas

En la figura, se observa el módulo de mantenimiento de mesas el que permite la creación de nuevas mesas, además la eliminación de las mismas.

El módulo presenta un textbox que permite el ingreso del nombre de la mesa, este valida si está creado con anterioridad.

El módulo muestra una grilla de las mesas creadas, mostrando los campos del código de la mesa, el nombre de la mesa y un botón eliminar.

MI Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Mantenimiento de Mesas

Nombre de Mesa:

Agregar

Código	Nombre de Mesa	Eliminar
2	Mesa 1	Eliminar
3	Mesa 2	Eliminar
4	Mesa 3	Eliminar
5	Mesa 4	Eliminar
6	Mesa 5	Eliminar
7	Mesa 6	Eliminar
8	Mesa 7	Eliminar

1 2

21. Módulo de mesas disponibles

En la figura, se observa el módulo de mesas disponibles que es un reporte online que permite observar el estatus de las mesas este puede dos tipos de estados: Mesa libre o mesa ocupada, el estatus de la mesas cambia a mesa ocupada cuando se realiza un pedido en la mesa seleccionada y vuelve a cambiar de estado cuando se cancela la factura de la misma.

En el módulo se puede observar el código de la mesa, el nombre de la mesa y el estado de la mesa. La grilla cuenta una paginación que muestra ocho registros por página.

MI Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Mesas Disponibles

Código	Nombre de Mesa	Estado de Mesa
2	Mesa 1	Mesa Ocupada
3	Mesa 2	Mesa Libre
4	Mesa 3	Mesa Libre
5	Mesa 4	Mesa Libre
6	Mesa 5	Mesa Libre
7	Mesa 6	Mesa Libre
8	Mesa 7	Mesa Libre

1 >

22. Módulo de facturación de pedidos

En la figura se observa el módulo de facturación de pedidos que tiene como funcionalidad cancelar los pedidos de la mesa seleccionada.

El módulo lista las mesas del restaurante, la fecha del día, la búsqueda de los clientes por medio del número de documento que nos servirá para cargar el nombre del cliente.

El módulo muestra una grilla con los datos de la cantidad de los platos seleccionados, el nombre del plato, el monto por unidad del mismo, el descuento que aplica la promoción, el monto base que es el monto sin la aplicación del descuento, el monto del descuento y el coste final. La grilla va cambiando de acuerdo a la mesa seleccionada.

En la parte inferior realiza un cálculo de los tres tipos de montos, el primero el monto ruto, el segundo el cálculo del IGV, este campo se puede editar a través de la tabla de parámetros del sistema y el monto final.



The screenshot shows a web application interface for 'MI Restaurante Experto'. At the top, there is a navigation menu with options: 'Mantenimiento', 'Restaurante', 'Transacciones', 'Caja', 'Reportes', 'Cerrar del Sistema', 'Sistema Experto', and 'Cerrar Sesión'. The main content area is titled 'Facturación de Pedidos'. It includes a dropdown menu for 'Mesa' (currently set to 'Mesa 1'), a search field for 'DNI', and a 'Fecha' field (set to '2015-01-01'). Below this is a table with the following data:

Cantidad	Plato	Monto CUI	Promoción	Monto Base	Monto Descuento	Monto Final
1	Chicharrón de Pollo	27.00		27.00	0.00	27.00

Below the table, there are three input fields for calculations: 'Monto Base' (27.00), 'IGV' (3.00), and 'Monto Final' (27.00). A green 'Aceptar' button is located at the bottom of the form.

23. Módulo de mantenimiento de clientes

En la figura, se observa el módulo de mantenimiento de clientes el que permite la creación y actualización de los clientes.

El módulo requiere el ingreso de los datos específicos de los clientes ya que la creación del mismo se realiza de forma rápida y óptima.

El módulo permite el ingreso del nombre o la razón social de la empresa, el apellido paterno si es que lo tuviera, el apellido materno si es que lo tuviera, una lista que muestra los tipos de documentos, el número de documento que permite la búsqueda de los clientes para certificarnos si se tiene agregado al cliente en el sistema y la fecha de nacimiento del mismo.



24. Módulo de reporte de almacén del día siguiente

En la figura, se muestra el reporte de almacén del día siguiente permite mostrar el promedio de uso de gastado de acuerdo al día siguiente.

Para poder obtener este reporte se utilizan los gastos por ingrediente filtrando sólo las transacciones del día siguiente según corresponda; es decir si estamos el día lunes, sólo se sacará los datos del día martes para realizar el reporte.

El reporte muestra los datos del nombre del ingrediente, la cantidad promedio, la cantidad máxima y la cantidad mínima del día seleccionado.

El reporte tiene la funcionalidad de poder exportarse en formato PDF, XLS, Word, Power Point y como fuente de datos.



Ingrediente	Cantidad Promedio	Cantidad Máx	Cantidad MÍN
Asa	0.04	1.25	0.00
Alfalfa	11.75	0.00	0.30
Arroz	0.00	0.30	2.55
Comeron	3.81	7.20	1.00
Carne	2.02	4.90	0.00
Carne Molida	2.50	5.20	0.50
Calabaza	0.50	10.00	3.00
Chorizo Compañero	7.20	11.00	3.00
Champiñon	0.01	1.00	0.20
Choclo	2.00	2.00	1.00

25. Módulo de reporte de ventas del día

En la figura, se muestra el módulo de reporte de ventas del día que es un reporte de cierre del sistema, que permite obtener un listado de los productos vendidos en el día.

El reporte se obtiene de los datos de las transacciones realizadas del día anterior, es decir para obtener dicho reporte se tendrá que realizar el cierre del sistema.

El reporte de ventas del día permite visualizar el nombre del plato, la cantidad vendida y el monto de ventas, que es el resultado de la multiplicación de los productos por el costo del mismo aplicado algún descuento si es que lo tuviera.

El reporte tiene la funcionabilidad de poder exportarse en formato PDF, XLS, Word, Power Point y como fuente de datos.

Plato	Cantidad Vendida	Monto de Ventas
Arroz de Camarón	3	\$1,135.00
Arroz Mechado con Frito	2	\$1,740.00
Carne de res con papas	2	\$2,200.00
Comida Arquepacha	11	\$1,770.00
Comida Andaluza	6	\$2,200.00
Comida Chilena	6	\$2,200.00
Chicharrón de Chircho	3	\$1,740.00
Chicharrón de Trufa	2	\$2,240.00
Filetes de Salmón	1	\$2,175.00
Chupe de Camarones	1	\$1,400.00

26. Módulo de reporte de almacén actual

En la figura, se observa el módulo del reporte de almacén que permite poder visualizar el estado actual del almacén, es decir que cuando se realice algún pedido de algún cliente este reporte se actualizará.

El reporte visualizará un listado de los ingredientes, la cantidad que se encuentra en el almacén, la cantidad mínima vendida de acuerdo al día este dato se obtiene de un cálculo del cierre de mes, como también el dato de la cantidad promedio del mismo.

El reporte ayuda para que el restaurante no se quede sin stock.

El reporte tiene la funcionalidad de poder exportarse en formato PDF, XLS, Word, Power Point y como fuente de datos.

Ingrediente	Cantidad	Cantidad Mínima	Cantidad Promedio
AC	26,00	1,00	1,00
Aliso	1,70	1,00	0,30
Ajete	30,00	1,00	7,00
Camote	10,00	1,00	0,70
Camote	24,00	1,00	2,00
Carne Mielca	3,00	1,00	3,00
Cebolla	25,00	1,00	9,50
Concha Arrozado	98,00	24,00	12,40
Concha Ouzo	120,00	23,00	10,00

27. Módulo de reporte de tiempo de espera por día de semana

En la figura 110 se muestra el reporte de módulo de reporte de tiempo de espera por día de semana que tiene la funcionalidad de observar el comportamiento de los clientes, ya que muestra en promedio el tiempo de demora de un comensal de acuerdo al día de semana.

Esta información permite al restaurante ver la posibilidad de crecimiento a nivel de mesas ya que si se cuenta con una tasa alta de promedio, será necesario que se amplíe el restaurante, o que se realice una revisión del tiempo de demora en la producción de los platos.

El reporte tiene la funcionalidad de poder exportarse en formato PDF, XLS, Word, Power Point y como fuente de datos.



28. Módulo de histórico de ventas por producto

En la figura, se observa el módulo de histórico de ventas por producto que permite visualizar los datos de las ventas filtrando por algún producto en específico.

El módulo muestra un listado de los productos que al seleccionarlo y hacer click en el botón de vista del reporte se visualiza una línea que muestra la evolución de las ventas en el último año.

Este reporte es útil ya que se ve la evolución de las ventas del mismo.

El reporte tiene la funcionabilidad de poder exportarse en formato PDF, XLS, Word, Power Point y como fuente de datos.



29. Módulo de reporte histórico de hora de demanda

En la figura, se observa el módulo de histórico de hora de demanda, que es un reporte que permite visualizar la cantidad de pedidos de acuerdo a las ventas por productos por la hora de la facturación.

El reporte permite el ingreso de una fecha que permite el filtro de las ventas de acuerdo al día de la semana.

El reporte sirve para identificar la cantidad de demanda que se tiene en diversos horarios del restaurante.

El reporte tiene la funcionalidad de poder exportarse en formato PDF, XLS, Word, Power Point y como fuente de datos.



30. Módulo de reporte del histórico de ventas por meses

En la figura, se observa el módulo de reporte del histórico de ventas por meses que nos permite observar la evolución general del restaurante, es decir que agrupa las ventas por el mes para poder obtener el reporte.

El reporte traza una línea del monto de las ventas de acuerdo a los últimos doce meses en actividad de la empresa.

Este reporte se obtiene de la data histórica es decir del data warehouse que se crear por medio de las transacciones históricas del sistema.

El reporte tiene la funcionalidad de poder exportarse en formato PDF, XLS, Word, Power Point y como fuente de datos.



31. Módulo del reporte de promedio de ventas por día de semana

En la figura, se observa el reporte del promedio de ventas por día de semana que permite identificar los días donde el restaurante presenta mayor actividad.

Este reporte es de principal necesidad ya que le permite al restaurante aumentar el personal si es necesario en los días donde se tiene una mayor demanda en el establecimiento.

Para la obtención del reporte se utiliza como fuente de datos el data warehouse de las ventas por día de semana y se obtiene el promedio de la misma.

El reporte tiene la funcionalidad de poder exportarse en formato PDF, XLS, Word, Power Point y como fuente de datos.



32. Módulo del Cierre del sistema

En la figura, se observa el módulo del cierre del sistema que tiene como funcionalidad la protección y población del data warehouse. Estos procesos son transparentes al usuario ya que sólo con un click en el botón cierre del sistema se ejecuta los procesos de forma interna.



33. Módulo de promociones sugeridas

En la figura, se observa el módulo de promociones sugeridas que es un módulo perteneciente al sistema experto.

Para obtener el reporte se necesita se realiza un pre procesamiento, el cual consiste en tomar a los 5 productos menos vendidos del restaurante y proponer promociones, según lo conversado por los expertos una promoción máxima no debería ser más alta que el del sesenta por ciento por producto.

El reporte nos mostrará una grilla con el nombre del plato, el precio actual que está a la venta, el costo de producción la ganancia actual que se tiene por plato vendido en la actualidad, el porcentaje de descuento propuesto por el sistema experto; este porcentaje va variando de acuerdo a la cantidad de ventas promedio, el reporte también muestra cual sería el posible costo del nuevo plato y el porcentaje que se tendría de la nueva ganancia si se aplicara el descuento.

Este módulo nos sirve para no tener productos estancados; es decir productos que no se consuman; para ello se insertarán promociones pero con la autorización del administrador del sistema, ya que él será el responsable de ingresar la promoción al sistema.

Mi Restaurante Experto								
Mantenimiento	Reservación	Transacciones	Caja	Reportes	Cantidad Sistema	Sistema Experto	Control Sistema	
Promociones Sugeridas								
Nombre de Plato	Precio Actual	Costo de Producción	Ganancia Actual	Descuento %	Costo Nuevo Plato	% Nueva Ganancia		
Pollo a la Havalana	33.00	10.13	135.15 %	30	21.00	107.31 %		
Pollo con Champañones	27.00	11.15	142.15 %	30	18.00	83.51 %		
Pollo a la Hogue	27.00	5.20	189.99 %	30	10.50	199.99 %		
Pequeño de Lomo	15.00	6.07	118.04 %	30	10.00	52.94 %		
Pepe a la minuta	15.00	4.48	234.82 %	30	10.00	134.38 %		

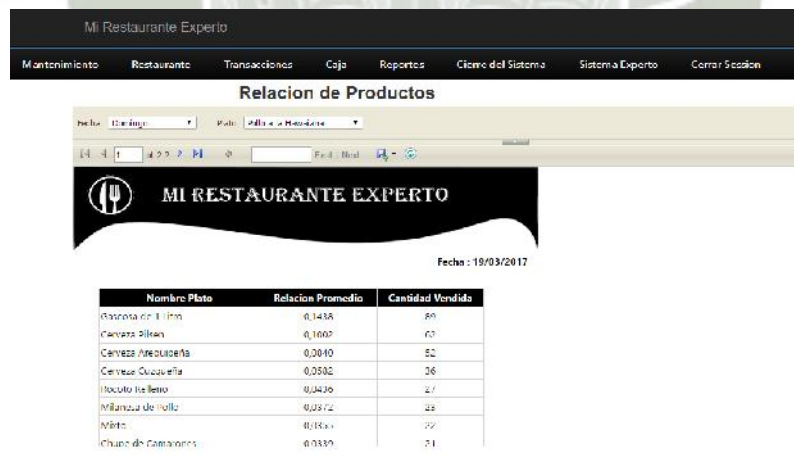
34. Módulo de reporte de la relación de productos

En la figura, se observa el módulo del reporte de la relación de productos que tiene la funcionalidad de poder identificar la relación que existe en la compra de algún producto con otro de acuerdo a algún día en específico.

Para la obtención de este reporte se realiza un pre procesamiento en el cierre diario del sistema el cual carga la base de conocimiento del sistema y nos entrega como resultado la relación de los productos.

Se necesita ingresar el día de la semana y el plato en gestión, al realizar click en la vista del reporte nos mostrará una grilla que mostrará el nombre del plato, la relación promedio expresado en cuatro decimales y la cantidad vendida repetida que se presenta.

El reporte tiene la funcionalidad de poder exportarse en formato PDF, XLS, Word, Power Point y como fuente de datos.



MI RESTAURANTE EXPERTO

Fecha: 19/03/2017

Nombre Plato	Relacion Promedio	Cantidad Vendida
Escudón de Limpio	0.1438	80
Cerveza Pilsen	0.1609	63
Cerveza Artesanal	0.2040	52
Cerveza Cusqueña	0.0502	16
Bocuto de Iteño	0.0426	27
Milanesa de pollo	0.0372	28
Milanesa	0.0354	29
Chupe de Camarones	0.0330	31

35. Módulo de hechos automáticos

En la figura, se observa el módulo de hechos automáticos mostrará las decisiones que generaron el sistema experto y la fecha de creación que se tomó dicha decisión.

MI Restaurante Experto

Mantenimiento Restaurante Transacciones Caja Reportes Cierre del Sistema Sistema Experto Cerrar Sesión

Hechos Automáticos

Código Ejecución	Descripción Ejecución	Fecha De Creación
8	Kocoto Helado 2017-04-30 11:00:00	2017/04/30 12:26
1	Piqueo de Pollo 2017-04-26 21:00:00	2017/04/26 03:40

© 2017 - Mi Restaurante Experto



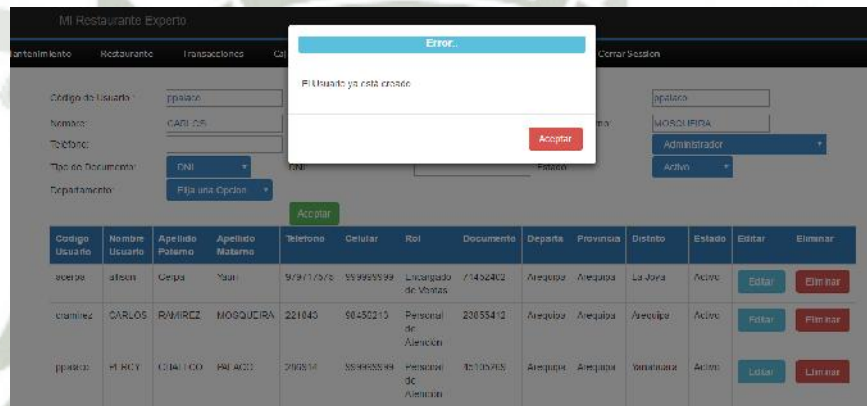
Anexo D

Pruebas de Validación y de sistema

En este tipo de pruebas se describirá el flujo de las operaciones en el sistema, las cuales nos servirán para poder identificar las alertas de validación de datos e identificar el correcto funcionamiento del sistema.

1. Creación de usuarios en el sistema.

Para la creación de un nuevo usuario se tendrá que comprobar que no existen usuarios con el mismo código ya que mostrará una alerta.



Cuando se desee eliminar a un usuario se pedirá una confirmación para poder deshabilitar al usuario seleccionado



2. Creación de roles

Para poder crear un nuevo rol en el sistema no se podrá ingresar un nombre de rol que ya está registrado en el sistema, ya que se dará una alerta de que ya existe en el sistema.

Código Rol	Nombre Rol	Fecha Modificación	Editar
6	Chef	24/06/2017	Editar

3. Asignar actividades al rol

Para asignar las actividades a un rol sólo listará las opciones en el sistema que no pertenezcan al rol seleccionado.

Sub Categoría	Actividad	Eliminar
Reservas	Usuarios	Eliminar

Sub Categoría	Actividad	Eliminar
Sistema	Usuarios	Eliminar
Inventos	Almacén	Eliminar

4. Mantenimiento de Almacén

Para el mantenimiento del almacén, se tendrá que ingresar las cantidades del insumo mayores a 0.



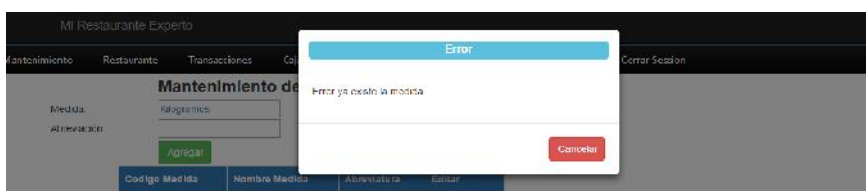
5. Mantenimiento de insumos

Para la creación de insumos, se deberá ingresar un nombre insumo que no se encuentre registrado en el sistema, ya que nos dará una alerta; a su vez el monto ingresado en el precio deberá ser mayor a 0.



6. Mantenimiento de medidas

Para la creación de una medida, se tendrá que ingresar el nombre y abreviatura de la medida que no esté registrado en el sistema, ya que nos dará una alerta que ya existe dicha medida.



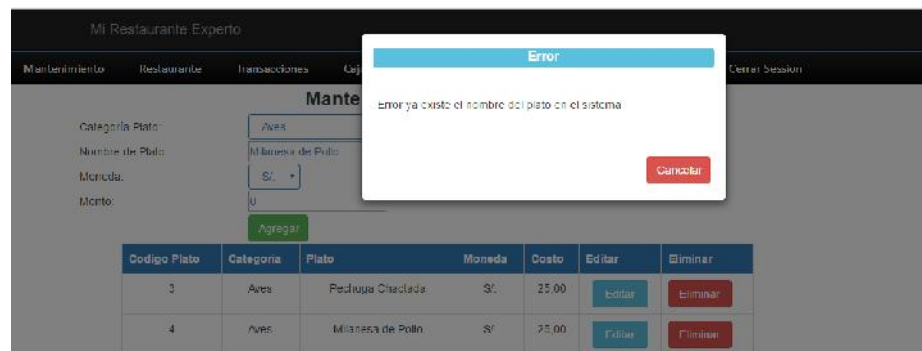
7. Mantenimiento de categorías de ingredientes

Para la creación de una nueva categoría de ingrediente, se tendrá que ingresar el nombre de la categoría de ingrediente que no se encuentre registrado en el sistema, ya que mostrará una alerta.



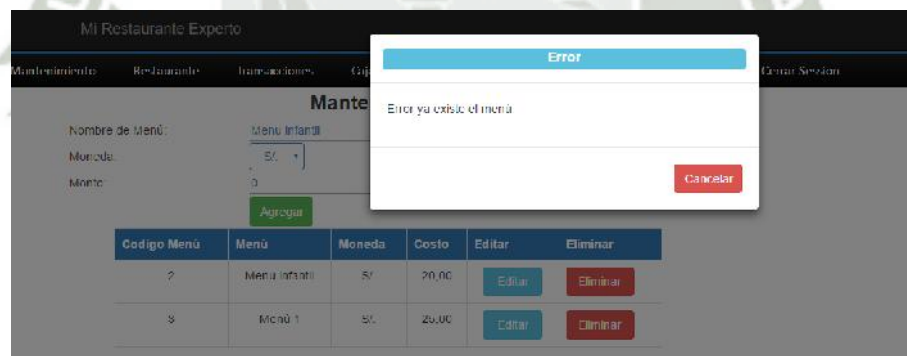
8. Mantenimiento de platos

Para la creación de un nuevo plato, se tendrá que ingresar un nombre del plato que no se encuentre registrado en el sistema, a su vez el precio deberá ser mayor a 0.



9. Mantenimiento de menús

Para la creación de un nuevo menú, se deberá ingresar un menú que no exista en el sistema y que el monto sea mayor a 0.



10. Mantenimiento de categoría de platos

Para la creación de una nueva categoría de platos, se deberá ingresar un nombre de categoría de platos que no se encuentre registrado en el sistema ya que mostrará una alerta.



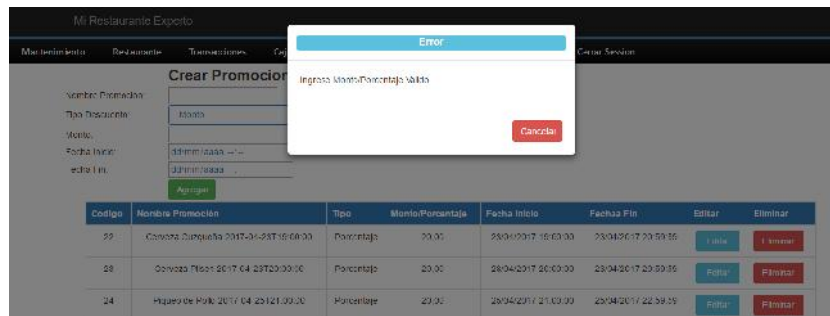
11. Preparación de platos

Para la preparación de los platos se deberá ingresar una cantidad mayor a 0 en el ingreso del ingrediente seleccionado.



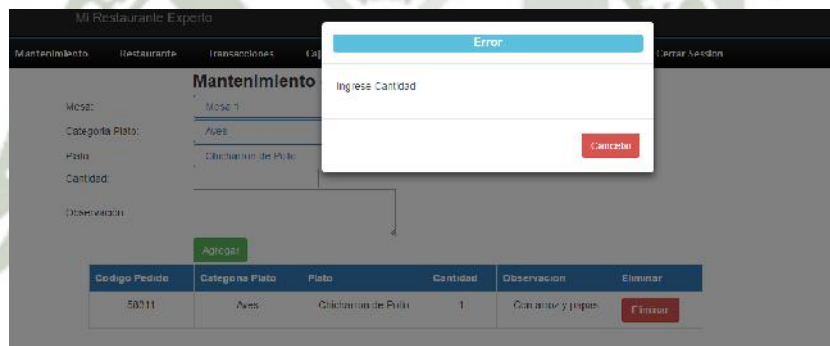
12. Mantenimiento de promociones

Para crear una promoción se deberá ingresar una cantidad mayor a 0.



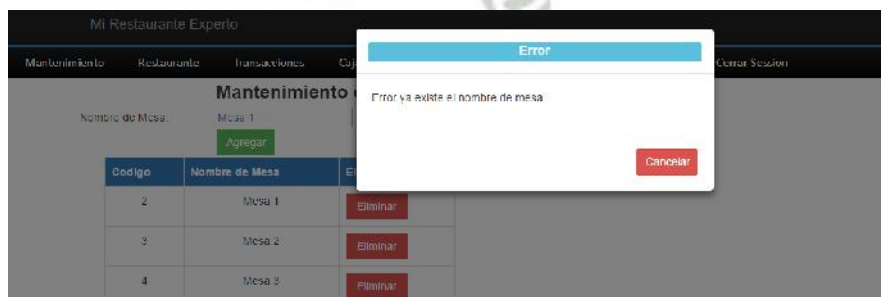
13. Mantenimiento de pedidos

Para el ingreso de un pedido se deberá ingresar en el campo de cantidades un valor mayor a 0 ya que saldrá una alerta.



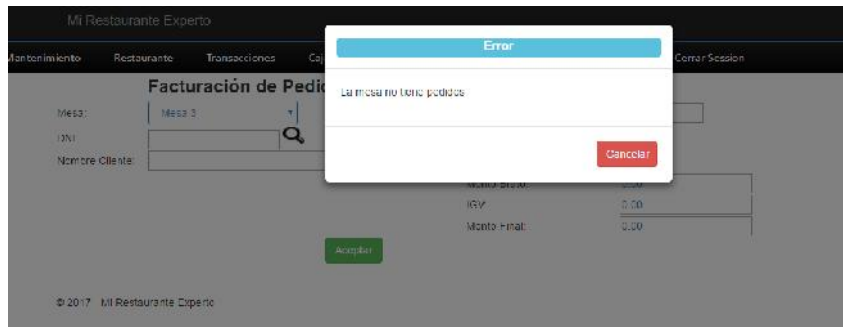
14. Mantenimiento de mesas

Para el ingreso de una nueva mesa, se deberá registrar un nombre de mesa que no exista en el sistema, ya que mostrará una alerta.



15. Facturación de pedidos

Para poder realizar una facturación, se deberá seleccionar una mesa que cuente con pedidos, si no mostrará una alerta.



Pruebas Unitarias

Se realizarán pruebas unitarias de los procedimientos de inserción a la base de datos.

Función: fn_CrearIngrediente

```

1 referencia
public void fn_CrearIngrediente(string Nombre,int Medida,int Categoria, double costo,
    string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_AgregarIngrediente";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", Nombre);
    cmd.Parameters.AddWithValue("Medida", Medida);
    cmd.Parameters.AddWithValue("Categoria", Categoria);
    cmd.Parameters.AddWithValue("Costo", costo);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
    
```

Procedimiento: SP_AgregarIngrediente

```

Create Procedure [dbo].[SP_AgregarIngrediente]
@Nombre varchar(70),
@Medida int,
@Categoria int,
@Costo decimal(10,2),
@Usuario varchar(15)
as
    
```

```
begin
INSERT INTO [dbo].[tb_Ingrediente]
    ([vNombreIngrediente]
    ,[iCodigoMedida]
    ,[dCreacion]
    ,[dModificacion]
    ,[vUsuarioCreacion]
    ,[vUsuarioModificacion]
    ,[iEstado]
    ,[iCodigoCategoria]
    ,[fcosto])
VALUES
    (@Nombre
    ,@Medida
    ,getdate()
    ,getdate()
    ,@Usuario
    ,@Usuario
    ,1
    ,@Categoria
    ,@Costo)
end
```

Prueba Unitaria: fn_CrearIngrediente

```
[TestMethod]
0 referencias
public void fn_CrearIngrediente()
{
    try
    {
        Alimento objAlimento = new Alimento();
        objAlimento.fn_CrearIngrediente("Prueba",1,1,5,"EBARROS");
    }
    catch (Exception e)
    {
    }
}
}
```

Resultado:

fn_CrearIngrediente

Origen: [UnitTest1.cs línea 31](#)

✔ Pruebas superadas - fn_CrearIngrediente

Tiempo transcurrido: 1 ms

Función: fn_CrearAlmacen

```

V Referencias
public void fn_CrearAlmacen(int Ingrediente, double Cantidad,
    double CantidadMaxima, double Caducidad, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_AgregarAlmacen";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Ingrediente", Ingrediente);
    cmd.Parameters.AddWithValue("Cantidad", Cantidad);
    cmd.Parameters.AddWithValue("CantidadMaxima", CantidadMaxima/100);
    cmd.Parameters.AddWithValue("Caducidad", Caducidad);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}

```

Procedimiento: SP_AgregarAlmacen

```

Create Procedure [dbo].[SP_AgregarAlmacen]
(
    @Ingrediente int,
    @Cantidad decimal(10,2),
    @CantidadMaxima decimal(10,2),
    @Caducidad decimal(10,2),
    @Usuario varchar(15)
)
as
begin
    Declare @Contador int
    select @Contador= count(*) from tb_Almacen where iCodigoIngrediente=@Ingrediente
    and iestado=1
    if @Contador = 0
    begin
        INSERT INTO [dbo].[tb_Almacen]
        ([iCodigoIngrediente],[iCantidad],[dCreacion],[dModificacion],[vUsuarioCreacion],
        [vUsuarioModificacion],[iEstado])
        VALUES
        (@Ingrediente,@Cantidad,getdate(),getdate(),@Usuario,@Usuario,1)
    end
    else
    begin
        update tb_Almacen
        set iCantidad=iCantidad,
        dModificacion=GETDATE(),
        vUsuarioModificacion=@Usuario
        where iCodigoIngrediente=@Ingrediente
    end
end

```

```

update tb_Ingrediente
set fcantidadMaxima=@CantidadMaxima,
  iTiempoCaducidad=@Caducidad
where iCodigoIngrediente=@Ingrediente
end

```

```

update tb_Almacen
set
iCantidad=0
where iCodigoIngrediente=@Ingrediente
and iCantidad<0
end

```

Prueba Unitaria: fn_CrearAlmacen

```

[TestMethod]
0 referencias
public void fn_CrearAlmacen()
{
    try
    {
        Almacenes objAlmacen = new Almacenes();
        objAlmacen.fn_CrearAlmacen(1, 5,100, 1, "SYSADM");
    }
    catch (Exception e)
    {
    }
}

```

Resultado:

fn_CrearAlmacen

Origen: UnitTest1.cs línea 45

✓ Pruebas superadas - fn_CrearAlmacen

Tiempo transcurrido: < 1 ms

Función: fn_CrearCategorial

0 referencias

```
public void fn_CrearCategorial(string Nombre, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_AgregarCategorial";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", Nombre);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
```

Procedimiento: SP_AgregarCategorial

```
Create procedure [dbo].[SP_AgregarCategorial]
(
@Nombre varchar(100),@Usuario varchar(15)
)
as
begin
INSERT INTO [dbo].[tb_CategoriaIngrediente]
([vNombreCategoria]
,[dCreacion]
,[dModificacion]
,[vUsuarioCreacion]
,[vUsuarioModificacion]
,[iEstado])
VALUES
(@Nombre
,[getdate()]
,[getdate()]
,@Usuario
,@Usuario
,1)
end
```

Prueba Unitaria: fn_CrearCategorial

```
[TestMethod]
0 referencias
public void fn_CrearCategorialI()
{
    try
    {
        CategoriaI objAlmacen = new CategoriaI();
        objAlmacen.fn_CrearCategorialI("Prueba", "SYSADM");
    }
    catch (Exception e)
    {
    }
}
}
```

Resultado:

fn_CrearCategorial

Origen: UnitTest1.cs línea 60

✔ Pruebas superadas - fn_CrearCategorial

Tiempo transcurrido: 65 ms

Función: fn_CrearCategorial

```
0 referencias
public void fn_CrearCategorial(string NombreCategorial, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_AgregarCategorialPlato";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", NombreCategorial);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
```

Procedimiento: SP_AgregarCategorialPlato

```
Create Procedure [dbo].[SP_AgregarCategorialPlato]
(
    @Nombre varchar(70),
    @Usuario varchar(15)
)
as
begin
INSERT INTO [dbo].[tb_CategorialPlato]
([vNombreCategorial]
,[dCreacion]
,[dModificacion]
```

```

        , [vUsuarioCreacion]
        , [vUsuarioModificacion]
        , [iEstado])
VALUES
    (@Nombre
    , getdate()
    , getdate()
    , @Usuario
    , @Usuario
    , 1)
end

```

Prueba Unitaria: fn_CrearCategoria

```

[TestMethod]
0 referencias
public void fn_CrearCategoria()
{
    try
    {
        CategoriaPlatos objCategoriaPlatos = new CategoriaPlatos();
        objCategoriaPlatos.fn_CrearCategoria("Prueba", "SYSADM");
    }
    catch (Exception e)
    {
    }
}
}

```

Resultado:

fn_CrearCategoria

Origen: UnitTest1.cs línea 75

✓ Pruebas superadas - fn_CrearCategoria

Tiempo transcurrido: 62 ms

Función: AgregarCliente

0 referencias

```
public void AgregarCliente(string Nombre, string Materno, string Paterno, int Tipo,
    string Documento, string Nacimiento, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_AgregarCliente";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", Nombre);
    cmd.Parameters.AddWithValue("Materno", Materno);
    cmd.Parameters.AddWithValue("Paterno", Paterno);
    cmd.Parameters.AddWithValue("Tipo", Tipo);
    cmd.Parameters.AddWithValue("Documento", Documento);
    cmd.Parameters.AddWithValue("Nacimiento", Nacimiento);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
```

Procedimiento: SP_AgregarCliente

```
Create Procedure [dbo].[SP_AgregarCliente]
(
    @Nombre varchar(70),
    @Materno varchar(100),
    @Paterno varchar(100),
    @Tipo int,
    @Documento varchar(15),
    @Nacimiento date,
    @Usuario varchar(15)
)
as
begin
    Declare @Validar int
    Select @Validar = count(*) from tb_Cliente
    where vnumeroDocumento=@Documento
    If @Tipo =2
    begin
        if @Validar = 0
        begin
            INSERT INTO [dbo].[tb_Cliente]
                ([vNombreCliente], [vApellidoMaterno], [vApellidoPaterno], [vRazonSocial]
                , [iTipoDocumento], [vNumeroDocumento], [dFechaNacimiento], [dCreacion]
                , [dModificacion], [vUsuarioCreacion], [vUsuarioModificacion]
                , [iEstado])
            VALUES
                (
                    ,
                    ,
                    ,
                    , @Nombre
                    , @Tipo
                )
        end
    end
end
```

```

        ,@Documento
        ,@Nacimiento
        ,GETDATE()
        ,getdate()
        ,@Usuario
        ,@Usuario
        ,1)
    end
else
begin
    update tb_Cliente set
    vRazonSocial=@Nombre,
    iTipoDocumento=@Tipo,
    dFechaNacimiento=@Nacimiento,
    dModificacion=getdate(),
    vUsuarioModificacion=@Usuario
    where vNumeroDocumento=@Documento
end
end
else
begin
if @Validar = 0
begin
    INSERT INTO [dbo].[tb_Cliente]
    ([vNombreCliente],[vApellidoMaterno],[vApellidoPaterno],[vRazonSocial]
    ,[iTipoDocumento],[vNumeroDocumento],[dFechaNacimiento],[dCreacion]
    ,[dModificacion],[vUsuarioCreacion],[vUsuarioModificacion]
    ,[iEstado])
    VALUES
    (@Nombre
    ,@Materno
    ,@Paterno
    ,
    ,
    ,@Tipo
    ,@Documento
    ,@Nacimiento
    ,GETDATE()
    ,getdate()
    ,@Usuario
    ,@Usuario
    ,1)
end
else
begin
    update tb_Cliente set
    vNombreCliente=@Nombre,
    vApellidoMaterno=@Materno,
    vApellidoPaterno=@Paterno,
    iTipoDocumento=@Tipo,
    dFechaNacimiento=@Nacimiento,
    dModificacion=getdate(),
    vUsuarioModificacion=@Usuario
    where vNumeroDocumento=@Documento
end
end
end
end

```

Prueba Unitaria: AgregarCliente

```
[TestMethod]
0 referencias
public void AgregarCliente()
{
    try
    {
        Client objCliente = new Client();
        objCliente.AgregarCliente("Prueba", "Prueba", "Prueba", 1, "71304628",
            "20170518", "SYSADM");
    }
    catch (Exception e)
    {
    }
}
}
```

Resultado:

AgregarCliente

Origen: [UnitTest1.cs línea 75](#)

✔ Pruebas superadas - AgregarCliente

Tiempo transcurrido: 71 ms

Función: CrearFactura

```
0 referencias
public void CrearFactura(string Documento, double Total, double Igv, double Final,
    string Usuario, string Fecha, int Mesa)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    string Hora= DateTime.Now.ToShortTimeString();
    cmd.CommandText = "SP_CrearFactura";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Documento", Documento);
    cmd.Parameters.AddWithValue("Total", Total);
    cmd.Parameters.AddWithValue("Igv", Igv);
    cmd.Parameters.AddWithValue("Final", Final);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Parameters.AddWithValue("Fecha", Fecha+ " "+ Hora);
    cmd.Parameters.AddWithValue("Mesa", Mesa);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
}
```

Procedimiento: SP_CrearFactura

```

Create Procedure [dbo].[SP_CrearFactura]
(
@Documento varchar(15),
@Total decimal(10,2),
@Igv decimal(10,2),
@Final decimal(10,2),
@Usuario varchar(15),
@Fecha datetime,
@Mesa int
)
as
begin
Declare @Cliente int
Declare @Factura int
Select @Cliente=iCodigoCliente from tb_Cliente
where vNumeroDocumento=@Documento

INSERT INTO [dbo].[tb_Factura]
([iCodigoCliente],[fMontoTotal],[fMontoIgv]
,[iMoneda],[fMontoFinalLocal],[dFecha]
,[iUsuarioCajero],[iCodigoLocal])
VALUES
(@Cliente
,@Total
,@Igv
,1
,@Final
,@Fecha
,@Usuario
,1)

Select @Factura=max(iCodigoFactura) from tb_Factura
where iCodigoCliente=@Cliente

insert into tb_DetalleFactura
select
@Factura,
null,
pla.iCodigoPlato,
1,
cast(count(*)*pla.fCosto as decimal(10,2)) -
cast(count(*)*(case when pro.itipoDescuento=1 then pro.fmonto when
pro.itipoDescuento=2 then cast(pro.fPorcentaje*pla.fCosto/100 as decimal(10,2))
else 0 end) as decimal(10,2)) MontoFinal,
count(*),
ped.iCodigoPedido
from tb_Pedido ped
inner join tb_Plato pla
on ped.iCodigoPlato=pla.iCodigoPlato and pla.iEstado=1
left join tb_PromoPlato ppp
on ppp.iCodigoPlato=pla.iCodigoPlato and ppp.iEstado=1
left join tb_Promocion pro
on pro.icodigopromocion= ppp.iCodigoPromocion and pro.iestado=1
where ped.iEstado=1
and ped.iCodigoMesa=@Mesa
group by
pla.vNombrePlato,
pla.fCosto,
isnull(pro.vNombrePromocion, '')
,pro.fPorcentaje,pla.fCosto,pro.iTipoDescuento,pro.fMonto,pla.iCodigoPlato,ped.iC
odigoPedido

```

```
update tb_Pedido
set iestado=3
where iCodigoPedido in
(select iCodigoPedido from tb_Pedido
where iCodigoMesa=@Mesa
and iEstado=1)
```

```
update tb_Mesa
set iEstado=1
where iCodigoMesa=@Mesa
end
```

Prueba Unitaria: CrearFactura

```
[TestMethod]
0 referencias
public void CrearFactura()
{
    try
    {
        Factu objFactura = new Factu();
        objFactura.CrearFactura("71304628", 100, 18, 118,
"SYSADM", "20170118 10:00", 1);
    }
    catch (Exception e)
    {
    }
}
}
```

Resultado:

CrearFactura

Origen: [UnitTest1.cs línea 92](#)

✔ Pruebas superadas - CrearFactura

Tiempo transcurrido: 93 ms

Función: fn_CrearMedida

0 referencias

```
public void fn_CrearMedida(string Nombre,string abreviacion, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_InsertarMedida";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", Nombre);
    cmd.Parameters.AddWithValue("Abreviacion", abreviacion);
    cmd.Parameters.AddWithValue("UsuarioCreador", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
```

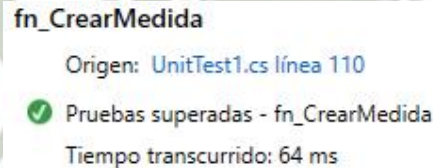
Procedimiento: SP_InsertarMedida

```
Create Procedure [dbo].[SP_InsertarMedida]
@Nombre Varchar(50),
@Abreviacion char(3),
@UsuarioCreador varchar(15)
as
Begin
INSERT INTO [dbo].[tb_Medida]
([vNombreMedida]
,[vAbreviatura]
,[dCreacion]
,[dModificacion]
,[vUsuarioCreacion]
,[vUsuarioModificacion]
,[iEstado])
VALUES
(@Nombre
,@Abreviacion
,getdate()
,getdate()
,@UsuarioCreador
,@UsuarioCreador
,1)
end
```

Prueba Unitaria: fn_CrearMedida

```
[TestMethod]
0 referencias
public void fn_CrearMedida()
{
    try
    {
        Medida objMedida = new Medida();
        objMedida.fn_CrearMedida("Prueba","PR", "SYSADM");
    }
    catch (Exception e)
    {
    }
}
}
```

Resultado:



fn_CrearMedida
 Origen: UnitTest1.cs línea 110
 ✓ Pruebas superadas - fn_CrearMedida
 Tiempo transcurrido: 64 ms

Función: fn_CrearMenu

```
0 referencias
public void fn_CrearMenu(string Nombre, int Moneda, double Costo, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_CrearMenu";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", Nombre);
    cmd.Parameters.AddWithValue("Moneda", Moneda);
    cmd.Parameters.AddWithValue("Costo", Costo);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
}
```

Procedimiento: SP_CrearMenu

```
Create Procedure [dbo].[SP_CrearMenu]
(
    @Nombre varchar(50),
    @Moneda int,
    @Costo decimal(10,2),
    @Usuario varchar(15)
```

```

)
as
begin
INSERT INTO [dbo].[tb_Menu]
    ([vNombreMenu]
    , [iMoneda]
    , [fCosto]
    , [dCreacion]
    , [dModificacion]
    , [vUsuarioCreacion]
    , [vUsuarioModificacion]
    , [iEstado])
VALUES
    (@Nombre
    , @Moneda
    , @Costo
    , getdate()
    , getdate()
    , @Usuario
    , @Usuario
    , 1)
end
    
```

Prueba Unitaria: fn_CrearMenu

```

[TestMethod]
0 referencias
public void fn_CrearMenu()
{
    try
    {
        Menu objMenu = new Menu();
        objMenu.fn_CrearMenu("Prueba", 1, 20, "SYSADM");
    }
    catch (Exception e)
    {
    }
}
    
```

Resultado:

fn_CrearMenu

Origen: [UnitTest1.cs](#) línea 125

✓ Pruebas superadas - fn_CrearMenu

Tiempo transcurrido: 65 ms

Función: fn_CrearMenuPlato

0 referencias

```
public void fn_CrearMenuPlato(int Menu, int Plato, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_InsertarMenuPlato";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Menu", Menu);
    cmd.Parameters.AddWithValue("Plato", Plato);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
```

Procedimiento: SP_InsertarMenuPlato

```
Create Procedure [dbo].[SP_InsertarMenuPlato]
(
    @Menu int,
    @Plato int,
    @Usuario varchar(15)
)
as
begin
    INSERT INTO [dbo].[tb_PMenuPlato]
    (
        [iCodigoMenu]
        , [iCodigoPlato]
        , [dCreacion]
        , [dModificacion]
        , [vUsuarioCreacion]
        , [vUsuarioModificacion]
        , [iEstado]
    )
    VALUES
    (
        @Menu
        , @Plato
        , getdate()
        , getdate()
        , @Usuario
        , @Usuario
        , 1
    )
end
```

Prueba Unitaria: fn_CrearMenuPlato

```
[TestMethod]
0 referencias
public void fn_CrearMenuPlato()
{
    try
    {
        Menus objMenu = new Menus();
        objMenu.fn_CrearMenuPlato(1, 1, "SYSADM");
    }
    catch (Exception e)
    {
    }
}
}
```

Resultado:

fn_CrearMenuPlato

Origen: UnitTest1.cs línea 140

✓ Pruebas superadas - fn_CrearMenuPlato

Tiempo transcurrido: 67 ms

Función: fn_AgregarMesa

```
0 referencias
public void fn_AgregarMesa(string Nombre, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_AgregarMesa";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", Nombre);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
}
```

Procedimiento: SP_AgregarMesa

```
Create Procedure [dbo].[SP_AgregarMesa]
(
    @Nombre varchar(70),
    @Usuario varchar(15)
)
as
begin
INSERT INTO [dbo].[tb_Mesa]
([vNombreMesa]
```

```

    , [dCreacion]
    , [dModificacion]
    , [vUsuarioCreacion]
    , [vUsuarioModificacion]
    , [iEstado])
VALUES
    (@Nombre
    , getdate()
    , getdate()
    , @Usuario
    , @Usuario
    , 1)
end

```

Prueba Unitaria: fn_AgregarMesa

```

[TestMethod]
0 referencias
public void fn_AgregarMesa()
{
    try
    {
        Mes objMesa = new Mes();
        objMesa.fn_AgregarMesa("Mesa Prueba", "SYSADM");
    }
    catch (Exception e)
    {
    }
}
}

```

Resultado:

fn_CrearMenuPlato

Origen: [UnitTest1.cs](#) línea 140

✓ Pruebas superadas - fn_CrearMenuPlato

Tiempo transcurrido: 67 ms

Función: AgregarPedido

```
public void AgregarPedido(int Mesa, int Plato,int Cantidad, string Observacion,
    string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_InsertarPedido";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Mesa", Mesa);
    cmd.Parameters.AddWithValue("Plato", Plato);
    cmd.Parameters.AddWithValue("Cantidad", Cantidad);
    cmd.Parameters.AddWithValue("Observacion", Observacion);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
```

Procedimiento: SP_InsertarPedido

```
Create Procedure [dbo].[SP_InsertarPedido]
(
    @Mesa int,
    @Plato int,
    @Cantidad int,
    @Observacion varchar(1000),
    @Usuario varchar(15)
)
as
begin
    update tb_mesa
    set iEstado=3
    where iCodigoMesa=@Mesa
    INSERT INTO [dbo].[tb_Pedido]
    ([iCodigoMesa]
    , [iCodigoPlato]
    , [iCantidad]
    , [iObservacion]
    , [dCreacion]
    , [dModificacion]
    , [vUsuarioCreacion]
    , [vUsuarioModificacion]
    , [iEstado])
    VALUES
    (@Mesa
    , @Plato
    , @Cantidad
    , @Observacion
    , getdate()
    , getdate()
    , @Usuario
    , @Usuario
    , 1)
    update alm
    set
    alm.iCantidad=alm.iCantidad - pue.fcantidad*@Cantidad
    from tb_PplatoIngrediente pue
    inner join tb_Plato pla
    on pla.iCodigoPlato=pue.iCodigoPlato and pla.iestado=1
```

```

inner join tb_Ingrediente ing
on ing.iCodigoIngrediente = pue.iCodigoIngrediente and ing.iEstado=1
inner join tb_Almacen alm
on alm.iCodigoIngrediente=ing.iCodigoIngrediente and alm.iEstado=1
where pue.iEstado=1
and pla.iCodigoPlato=@Plato

end

```

Prueba Unitaria: AgregarPedido

```

[TestMethod]
0 referencias
public void AgregarPedido()
{
    try
    {
        Pedid objPedido = new Pedid();
        objPedido.AgregarPedido(1, 1, 1, "Prueba","SYSADM");
    }
    catch (Exception e)
    {
    }
}
}

```

Resultado:

AgregarPedido

Origen: UnitTest1.cs línea 172

✓ Pruebas superadas - AgregarPedido

Tiempo transcurrido: 86 ms

Función: fn_CrearPlato

```

public void fn_CrearPlato(string Nombre,int Categoria,int Moneda, double Costo,
string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_AgregarPato";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", Nombre);
    cmd.Parameters.AddWithValue("Categoria", Categoria);
    cmd.Parameters.AddWithValue("Moneda", Moneda);
    cmd.Parameters.AddWithValue("Costo", Costo);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
}

```

Procedimiento: SP_AgregarPato

```

Create Procedure [dbo].[SP_AgregarPato]
(
@Nombre varchar(70),
@Categoria int,
@Moneda int,
@Costo decimal(10,2),
@Usuario varchar(15)
)
as
begin

INSERT INTO [dbo].[tb_Plato]
([vNombrePlato]
,[iCategoriaPlato]
,[iMoneda]
,[fCosto]
,[dCreacion]
,[dModificacion]
,[vUsuarioCreacion]
,[vUsuarioModificacion]
,[iEstado])
VALUES
(@Nombre
,@Categoria
,@Moneda
,@Costo
,getdate()
,getdate()
,@Usuario
,@Usuario
,1)

end
    
```

Prueba Unitaria: fn_CrearPlato

```

[TestMethod]
0 referencias
public void fn_CrearPlato()
{
    try
    {
        Plato objPlato = new Plato();
        objPlato.fn_CrearPlato("Prueba", 1, 1, 25, "SYSADM");
    }
    catch (Exception e)
    {
    }
}
    
```

Resultado:

fn_CrearPlato

Origen: [UnitTest1.cs](#) línea 187

✔ Pruebas superadas - fn_CrearPlato

Tiempo transcurrido: 67 ms

Función: fn_CrearPromocion

U referencias

```
public void fn_CrearPromocion(string Nombre, int Tipo, double Monto,
    string Usuario, string FechaI, string FechaF)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_CrearPromocion";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", Nombre);
    cmd.Parameters.AddWithValue("Tipo", Tipo);
    cmd.Parameters.AddWithValue("Monto", Monto);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Parameters.AddWithValue("FechaI", FechaI+":00");
    cmd.Parameters.AddWithValue("FechaF", FechaF + ":00");
    cmd.Connection = con;
    cmd.ExecuteReader();
}
```

Procedimiento: SP_CrearPromocion

```
Create Procedure [dbo].[SP_CrearPromocion]
(
    @Nombre varchar(70),
    @Tipo int,
    @Monto decimal(10,2),
    @Usuario varchar(15),
    @FechaI datetime,
    @FechaF datetime
)
as
begin
    if @Tipo=1
    begin

INSERT INTO [dbo].[tb_Promocion]
    ([vNombrePromocion]
    ,[iTipoDescuento]
    ,[fPorcentaje]
    ,[fMonto]
    ,[dCreacion]
    ,[dModificacion]
    ,[vUsuarioCreacion]
    ,[vUsuarioModificacion]
    ,[iEstado]
    ,dFechaInicio
```

```

        ,dFechaFin)
VALUES
    (@Nombre
    ,@Tipo
    ,0
    ,@Monto
    ,getdate()
    ,getdate()
    ,@Usuario
    ,@Usuario
    ,1,@FechaI,@FechaF)
end
else
begin
INSERT INTO [dbo].[tb_Promocion]
    ([vNombrePromocion]
    , [iTipoDescuento]
    , [fPorcentaje]
    , [fMonto]
    , [dCreacion]
    , [dModificacion]
    , [vUsuarioCreacion]
    , [vUsuarioModificacion]
    , [iEstado]
    , dFechaInicio
    , dFechaFin)
VALUES
    (@Nombre
    ,@Tipo
    ,@Monto
    ,0
    ,getdate()
    ,getdate()
    ,@Usuario
    ,@Usuario
    ,1,@FechaI,@FechaF)
end
end

```

Prueba Unitaria: fn_CrearPromocion

```

[TestMethod]
0 referencias
public void fn_CrearPromocion()
{
    try
    {
        Promocion objPromocion = new Promocion();
        objPromocion.fn_CrearPromocion("Prueba", 1, 10,"SYSADM",
            "20170101 10:00", "20170102 10:00");
    }
    catch (Exception e)
    {
    }
}
}

```

Resultado:

fn_CrearPromocion

Origen: [UnitTest1.cs](#) línea 202

✓ Pruebas superadas - fn_CrearPromocion

Tiempo transcurrido: 77 ms

Función: fn_CrearPlatoPromo

0 referencias

```
public void fn_CrearPlatoPromo(int Promo, int Plato, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_AgregarPromoPlato";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Promo", Promo);
    cmd.Parameters.AddWithValue("Plato", Plato);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
```

Procedimiento: SP_AgregarPromoPlato

```
Create Procedure [dbo].[SP_AgregarPromoPlato]
(
    @Promo int,
    @Plato int,
    @Usuario varchar(15)
)
as
begin
INSERT INTO [dbo].[Tb_PPromoPlato]
    ([iCodigoPromocion]
    ,[iCodigoPlato]
    ,[dCreacion]
    ,[dModificacion]
    ,[vUsuarioCreacion]
    ,[vUsuarioModificacion]
    ,[iEstado])
VALUES
    (@Promo
    ,@Plato
    ,getdate()
    ,getdate()
    ,@Usuario
    ,@Usuario
    ,1)
end
```

Prueba Unitaria: fn_CrearPlatoPromo

```
[TestMethod]
0 referencias
public void fn_CrearPlatoPromo()
{
    try
    {
        Promocion objPromocion = new Promocion();
        objPromocion.fn_CrearPlatoPromo(1, 1, "SYSADM");
    }
    catch (Exception e)
    {
    }
}
}
```

Resultado:

fn_CrearPlatoPromo

Origen: UnitTest1.cs línea 218

✓ Pruebas superadas - fn_CrearPlatoPromo

Tiempo transcurrido: 65 ms

Función: fn_CrearRol

```
0 referencias
public void fn_CrearRol(string NombreRol, string Usuario )
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_InsertarRol";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Nombre", NombreRol);
    cmd.Parameters.AddWithValue("UsuarioCreador", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
}
```

Procedimiento: SP_InsertarRol

```
Create Procedure [dbo].[SP_InsertarRol]
@Nombre Varchar(70),
@UsuarioCreador varchar(15)
as
Begin
INSERT INTO [dbo].[tb_Rol]
([vNombreRol]
,[dCreacion]
,[dModificacion]
,[vUsuarioCreacion]
```

```

        , [vUsuarioModificacion]
        , [iEstado])
VALUES
    (@Nombre
    , GETDATE()
    , GETDATE()
    , @UsuarioCreador
    , @UsuarioCreador
    , 1)
end
    
```

Prueba Unitaria: fn_CrearRol

```

[TestMethod]
0 referencias
public void fn_CrearRol()
{
    try
    {
        Rol objRol = new Rol();
        objRol.fn_CrearRol("Prueba", "SYSADM");
    }
    catch (Exception e)
    {
    }
}
    
```

Resultado:

fn_CrearRol

Origen: [UnitTest1.cs línea 234](#)

✔ Pruebas superadas - fn_CrearRol

Tiempo transcurrido: 65 ms

Función: Fn_CrearActRol

```

0 referencias
public void Fn_CrearActRol(int Rol, int Actividad, string Usuario)
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_CrearActRol";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("Rol", Rol);
    cmd.Parameters.AddWithValue("Actividad", Actividad);
    cmd.Parameters.AddWithValue("Usuario", Usuario);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
    
```

Procedimiento: SP_CrearActRol

```

Create Procedure [dbo].[SP_CrearActRol]
@Rol int,
@Actividad int,
@Usuario varchar(15)
as
begin
INSERT INTO [dbo].[tb_PRolActividad]
([iCodigoRol]
,[iCodigoActividad]
,[dCreacion]
,[dModificacion]
,[vUsuarioCreacion]
,[vUsuarioModificacion]
,[iEstado])
VALUES
(@Rol
,[Actividad]
,[getdate()]
,[getdate()]
,[@Usuario]
,[@Usuario]
,[1])
end
    
```

Prueba Unitaria: Fn_CrearActRol

```

[TestMethod]
0 referencias
public void Fn_CrearActRol()
{
    try
    {
        Rol objRol = new Rol();
        objRol.Fn_CrearActRol(10,1, "SYSADM");
    }
    catch (Exception e)
    {
    }
}
    
```

Resultado:

Fn_CrearActRol

Origen: UnitTest1.cs línea 249

✔ Pruebas superadas - Fn_CrearActRol

Tiempo transcurrido: 67 ms

Función: fn_CrearUsuario

```

public void fn_CrearUsuario(string CodigoUsuario,string Contraseña,string nombre,string materno,
    string paterno,int rol,string telefono,string celular,int ubigeo,int tipodoc,string documento,
    string usuario,int estado
    )
{
    Conexion objCon = new Conexion();
    SqlConnection con = new SqlConnection();
    con = objCon.Abrir();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SP_InsertarUsuario";
    cmd.CommandType = System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("CodigoUsuario", CodigoUsuario);
    cmd.Parameters.AddWithValue("Contraseña", Contraseña);
    cmd.Parameters.AddWithValue("NombreUsuario", nombre);
    cmd.Parameters.AddWithValue("ApellidoMaterno", materno);}
    cmd.Parameters.AddWithValue("ApellidoPaterno", paterno);
    cmd.Parameters.AddWithValue("CodigoRol", rol);
    cmd.Parameters.AddWithValue("Telefono", telefono);
    cmd.Parameters.AddWithValue("Celular", celular);
    cmd.Parameters.AddWithValue("CodigoUbigeo", ubigeo);
    cmd.Parameters.AddWithValue("TipoDocumento", tipodoc);
    cmd.Parameters.AddWithValue("Documento", documento);
    cmd.Parameters.AddWithValue("Usuario", usuario);
    cmd.Parameters.AddWithValue("Estado", estado);
    cmd.Connection = con;
    cmd.ExecuteReader();
}
}

```

Procedimiento: SP_InsertarUsuario

```

Create Procedure [dbo].[SP_InsertarUsuario]
@CodigoUsuario varchar(20),
@Contraseña varchar(20),
@NombreUsuario varchar(50),
@ApellidoMaterno varchar(50),
@ApellidoPaterno varchar(50),
@CodigoRol int,
@Telefono varchar(10),
@Celular varchar(10),
@CodigoUbigeo int,
@TipoDocumento int,
@Documento varchar(12),
@Usuario varchar(15),
@Estado int
as
begin

```

```

INSERT INTO [dbo].[tb_Usuario]
([vCodigoUsuario]
,[vContraseña]
,[vNombreUsuario]
,[vApellidoMaterno]
,[vApellidoPaterno]
,[iCodigoRol]
,[vTelefono]
,[vCelular]
,[iCodigoUbigeo]
,[iTipoDocumento]
,[vDocumento]
,[dCreacion]
,[dModificacion]
,[vUsuarioCreacion]

```

```

        , [vUsuarioModificacion]
        , [iEstado])
VALUES
    (@CodigoUsuario
    , @Contrasena
    , @NombreUsuario
    , @ApellidoMaterno
    , @ApellidoPaterno
    , @CodigoRol
    , @Telefono
    , @Celular
    , @CodigoUbigeo
    , @TipoDocumento
    , @Documento
    , getdate()
    , getdate()
    , @Usuario
    , @Usuario
    , @Estado)
end

```

Prueba Unitaria: fn_CrearUsuario

```

[TestMethod]
0 referencias
public void fn_CrearUsuario()
{
    try
    {
        Usuario objUsuario = new Usuario();
        objUsuario.fn_CrearUsuario("TUSUARIO", "1234", "Prueba", "materno",
        "paterno", 1, "112233", "996677826", 115522, 1, "77556688", "SYSADM", 1);
    }
    catch (Exception e)
    {
    }
}
}

```

Resultado:

fn_CrearUsuario

Origen: [UnitTest1.cs línea 265](#)

✓ Pruebas superadas - fn_CrearUsuario

Tiempo transcurrido: 68 ms

Pruebas de integración

Para las pruebas de integración se realizó las pruebas unitarias en forma grupal teniendo el siguiente resultado.

Ejecutar todas Ejecutar... Lista de reproducción: Todas las pruebas	
Pruebas: superadas (16)	
✓ AgregarCliente	< 1 ms
✓ AgregarPedido	< 1 ms
✓ CrearFactura	< 1 ms
✓ fn_AgregarMesa	< 1 ms
✓ fn_CrearActRol	< 1 ms
✓ fn_CrearAlmacen	1 ms
✓ fn_CrearCategorial	< 1 ms
✓ fn_CrearIngresante	73 ms
✓ fn_CrearMedida	< 1 ms
✓ fn_CrearMenu	< 1 ms
✓ fn_CrearMenuPlato	< 1 ms
✓ fn_CrearPlato	1 ms
✓ fn_CrearPlatoPromo	< 1 ms
✓ fn_CrearPromocion	< 1 ms
✓ fn_CrearRol	< 1 ms
✓ fn_CrearUsuario	< 1 ms

Resumen	
Última serie de pruebas superadas (Tiempo de ejecución total 0:00:05)	
✓ 16 pruebas superadas	



Anexo E

1. Proceso de registro de almacén

1.1. Caso de uso Mantenimiento de almacén

Precondiciones: Ingreso de los ingredientes.

Flujo principal: Se ingresan las cantidades de cada ingrediente al sistema.

1.2. Caso de uso Mantenimiento de insumos

Precondiciones: Registro de los insumos en el almacén.

Flujo principal: Se registran los insumos y el costo de los ingredientes en el sistema.

Flujo alternativo: Se registrarán los insumos en un archivo Excel para luego ser ingresados al sistema.

1.3. Caso de uso Mantenimiento de medidas

Precondiciones: No tiene.

Flujo principal: Se ingresará la medida y la abreviatura de la medida en el sistema.

1.4. Caso de uso Categoría de alimentos

Precondiciones: No tiene.

Flujo principal: Se ingresará las categorías de alimentos al sistema.

2. Proceso de Ventas

2.1. Caso de uso de Mantenimiento de usuarios

Precondiciones: Identificar las personas que intervendrán en el sistema.

Flujo principal: Se registrarán las características de las personas que interactuarán en el sistema.

2.2. Caso de uso de Mantenimiento de roles

Precondiciones: Registros de los módulos en el sistema.

Flujo principal: Se creará un nuevo rol y se asignará los módulos que no tengan el rol seleccionado.

2.3. Caso de uso de Mantenimiento de platos

Precondiciones: Realizar el listado de los productos que venderá el restaurante.

Flujo principal: Registrar los productos que venderá la empresa, además de ingresar el precio de venta de dicho producto.

2.4. Caso de uso de Mantenimiento de promociones

Precondiciones: Tener registrado los productos de la empresa en el sistema.

Flujo principal: Se listarán los productos y se elegirá el tipo de descuento el cual será aplicado al producto seleccionado.

2.5. Caso de uso de Mantenimiento de mesas

Precondiciones: Listado de las mesas

Flujo principal: Se ingresará el nombre de las mesas al sistema.

2.6. Caso de uso de Cierre del sistema

Precondiciones: Bases de datos del sistema.

Flujo principal: Se realizará una copia de las bases del sistema.

2.7. Caso de uso de Visualización de reportes

Precondiciones: Análisis de los datos del data warehouse

Flujo principal: Se obtendrá los reportes desde los módulos de reportería del sistema.

2.8. Caso de uso de Sugerencias del sistema experto

Precondiciones: Análisis en línea del estatus del restaurante.

Flujo principal: El sistema de forma automática por medio de un job realizará la evaluación del estatus del sistema, para que de acuerdo a ciertas circunstancias tome decisiones al sistema.

2.9. Caso de uso de Registro de mesas disponibles

Precondiciones: Realizar la evaluación del estatus de las mesas

Flujo principal: Mostrará un reporte del estatus de las mesas.

2.10. Caso de uso de Mantenimiento de caja

Precondiciones: Realizar los pedidos de los clientes.

Flujo principal: Los clientes registrarán los pedidos en el sistema, la persona encargada indicará el nombre de la mesa para proceder el cobro de dichos pedidos.

2.11. Caso de uso de Mantenimiento de los clientes

Precondiciones: Información de los clientes.

Flujo principal: El cliente indicará las características a la persona encargada para que pueda registrar dichas características en el sistema.

2.12. Caso de uso de Mantenimiento de preparación de platos

Precondiciones: Ingreso en el sistema el listado de productos e insumos.

Flujo principal: El usuario asignará las cantidades de los insumos necesarios para la elaboración de un determinado plato.

2.13. Caso de uso de Mantenimiento de pedidos

Precondiciones: Registro de los pedidos de los clientes.

Flujo principal: La persona encargada tomará el pedido indicado de los clientes para poder registrarlos en el sistema; además de indicar la cantidad de los productos que desee.

Flujo alternativo: Se registrarán los pedidos en una libreta de papel, para posteriormente registrarlo al sistema.