

**UNIVERSIDAD DE HUANUCO**  
**FACULTAD DE INGENIERÍA**  
**E.A.P. DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**



**TRABAJO DE SUFICIENCIA PROFESIONAL**

**“IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO DE CAJA Y  
ALMACEN PARA LA GESTIÓN DEL RECREO FALCON DE LA CIUDAD  
DE HUÁNUCO”**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE  
SISTEMAS E INFORMÁTICA**

**Presentado por el Bachiller:**

**SANTIAGO GONZALES, ABEL**

**Asesor:**

**MG. ZACARÍAS VENTURA, HÉCTOR RAÚL**

**HUÁNUCO - PERÚ**

**2016**



# UNIVERSIDAD DE HUANUCO

## Facultad de Ingeniería

E.A.P. DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

### ACTA DE SUSTENTACIÓN DE TRABAJO DE SUFICIENCIA PROFESIONAL PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO (A) DE SISTEMAS E INFORMÁTICA

En la ciudad de Huánuco, siendo las 10:30 horas del día 02 del mes de AGOSTO..... del año 2019., en el Auditorio de la Facultad de Ingeniería, en cumplimiento de lo señalado en el Reglamento de Grados y Títulos de la Universidad de Huánuco, se reunieron los Jurados Calificadores Nombrados mediante la Resolución N° 735-2019-D-FI-UDH integrado por los docentes:

Mg. BERTHA LUCILO CAMPOS RIOS (Presidente)  
Ing. ETHEL JHOVANI MORZANO LOZANO (Secretario)  
Ing. PSOLO EDUER SOLIS JORA (Vocal)

Para calificar el Trabajo de Suficiencia Profesional intitulada:

"IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO DE COJA Y ALMACÉN PARA LA GESTIÓN DEL RECREO FOLCLÓRICO DE LA CIUDAD DE HUANUCO"

....." presentado por el (la) Bachiller ABEL SANTIAGO GONZALES....., para optar el Título Profesional de Ingeniero (a) de Sistemas e Informática.

Dicho acto de sustentación se desarrolló en dos etapas: exposición y absolución de preguntas: procediéndose luego a la evaluación por parte de los miembros del Jurado.

Habiendo absuelto las objeciones que le fueron formuladas por los miembros del Jurado y de conformidad con las respectivas disposiciones reglamentarias, procedieron a deliberar y calificar, declarándolo (a) APROBADO..... por UNANIMIDAD..... con el calificativo cuantitativo de 14..... y cualitativo de SUFICIENTE

Siendo las 12:00 horas del día 02 del mes de AGOSTO del año 2019., los miembros del Jurado Calificador firman la presente Acta en señal de conformidad.

Bertha Lucilo Campos Rios  
Presidente

Ethel Jhovani Morzano Lozano  
Secretario

Psolo Eduer Solis Jora  
Vocal

**DEDICATORIA:**

A mi familia por su apoyo y guía incondicional.

### **AGRADECIMIENTO:**

A las personas que me apoyaron e hicieron que el trabajo se efectúe con éxito en especial a aquellos que me compartieron sus conocimientos.

## ÍNDICE

INTRODUCCIÓN .....	X
RESUMEN .....	XII
SUMARY .....	XIII
CAPÍTULO I.....	14
ASPECTOS DE LA ENTIDAD RECEPTORA .....	14
1.1    NOMBRE O RAZÓN SOCIAL .....	14
1.2    RUBRO.....	14
1.3    UBICACIÓN / DIRECCIÓN .....	14
1.4    RESEÑA.....	14
CAPITULO II.....	15
ASPECTOS DEL ÁREA O SECCIÓN .....	15
CAPITULO III.....	16
IDENTIFICACIÓN DE LA SITUACIÓN PROBLEMÁTICA.....	16
3.1.    PROBLEMA DE INVESTIGACIÓN.....	16
3.1.1. DESCRIPCIÓN DEL PROBLEMA .....	16
3.1.2. FORMULACIÓN DEL PROBLEMA .....	17
3.1.2.1. PROBLEMA PRINCIPAL .....	17
3.1.2.2. PROBLEMAS ESPECÍFICOS.....	17
3.1.3. OBJETIVOS.....	17
3.1.3.1. OBJETIVO GENERAL.....	17
3.1.3.2. OBJETIVOS ESPECÍFICOS .....	17

3.1.4. JUSTIFICACIÓN.....	17
3.1.5. HIPÓTESIS.....	18
3.1.6. VARIABLES.....	18
3.1.6.1. VARIABLE INDEPENDIENTE.....	18
3.1.6.2. VARIABLE DEPENDIENTE.....	18
3.1.7. OPERACIONALIZACIÓN DE VARIABLES.....	18
3.1.8. LIMITACIONES.....	18
3.1.9. VIABILIDAD .....	19
3.1.9.1. RECURSOS HUMANOS:.....	19
3.1.9.2. RECURSOS ECONÓMICOS: .....	19
3.1.9.3. RECURSOS TECNOLÓGICOS (MATERIALES).....	19
3.2. MARCO TEORICO.....	19
3.2.1. ANTECEDENTES DE LA INVESTIGACIÓN .....	19
3.2.1.1. INTERNACIONALES .....	19
3.2.1.2. NACIONALES .....	21
3.2.2. BASES TEÓRICAS .....	22
3.2.3. DEFINICIONES CONCEPTUALES.....	31
3.3. MATERIALES Y MÉTODOS.....	32
3.3.1. TIPO DE INVESTIGACIÓN.....	32
CAPITULO IV .....	33
APORTES PARA LA SOLUCIÓN DEL PROBLEMA.....	33
<b>4.1. ANÁLISIS DEL SISTEMA PROPUESTO .....</b>	<b>33</b>

4.2.	DISEÑO DE LA BASE DE DATOS.....	38
4.3.	MODELO DE CASOS DE USO .....	43
4.4.	DIAGRAMA DE ROBUSTEZ POR MÓDULOS .....	63
4.5.	DIAGRAMA DE SECUENCIA POR MÓDULOS .....	65
4.6.	CODIFICACIÓN .....	68
4.7.	PRUEBAS .....	87
4.8.	IMPLANTACIÓN DEL SISTEMA .....	95
	CONCLUSIONES .....	113
	RECOMENDACIONES .....	114
	REFERENCIAS BIBLIOGRÁFICAS .....	115
	ANEXOS.....	116
	MATRIZ DE CONSISTENCIA.....	116

## ÍNDICE DE TABLAS

Tabla 1 CONTROL DE CAJA Y EGRESOS .....	33
Tabla 2 FUNCIONALIDAD .....	34
Tabla 3 DESCRIPCIÓN DEL TIPO DE ACTOR .....	37
Tabla 4 EFINICION DE ACCESO AL SISTEMA.....	38
Tabla 5 Sección principal autenticar usuario .....	45
Tabla 6 Curso normal de los eventos autenticar usuarios .....	45
Tabla 7 Cursos alternos, autenticar usuarios .....	46
Tabla 8 Especificación de caso de uso de administración de usuarios .	47
Tabla 9 Curso normal de los eventos de administrador de usuarios.....	47
Tabla 10 Cursos alternos, Administrar usuario .....	48
Tabla 11 Seccion nuevo usuario (curso normal de los eventos).....	49
Tabla 12 Cursos alternos, seccion nuevo usuario .....	49
Tabla 13 Seccion eliminar usuario.....	50
Tabla 14 Cursos alternos, seccion eleiminar usuario.....	50
Tabla 15 Especificación de caso de uso de administrar mesas .....	51
Tabla 16 Curso normal de eventos de caso de uso de administrar ....	52
Tabla 17 Cursos alternativos .....	52
Tabla 18 Sección administrar nuevas mesas .....	53
Tabla 19 Cursos alternos .....	53
Tabla 20 Especificación de caso de uso consultar ventas .....	54
Tabla 21 Curso normal de eventos de caso de uso (CU) de consulta de ventas .....	54
Tabla 22 Cursos alternativos .....	55
Tabla 23 Especificación de caso de uso administrar inventario .....	56



Tabla 24 Curso normal de eventos de caso de uso (CU) administrar inventario.....	56
Tabla 25 Cursos alternativos de administrar inventario .....	57
Tabla 26 Curso norma de los eventos sección ingresar inventario. ....	57
Tabla 27 Cursos alternativos sección ingresar inventario .....	58
Tabla 28 Curso norma de los eventos sección ingresar categoría. ....	58
Tabla 29 Cursos alternativos sección ingresar categoría.....	59
Tabla 30 Sección principal administrar productos .....	60

## INDICE DE GRAFICOS

Gráfico 1 MODELO ENTIDAD RELACIÓN .....	41
Gráfico 2 MODELO RELACIONAL DE LA BASE DE DATOS .....	42
Gráfico 3 Subsistema de facturación .....	43
Gráfico 4 Subsistema de egresos y caja .....	43
Gráfico 5 Subsistema de Usuarios .....	44
Gráfico 6 Subsistema de Pedidos .....	44
Gráfico 7 Descripción de casos de uso de autenticar al usuario .....	45
Gráfico 8 Diagrama de caso de uso de administrador de usuarios .....	47
Gráfico 9 Diagrama de caso de uso de administrar mesas .....	51
Gráfico 10 Caso de uso consultar ventas .....	54
Gráfico 11 Caso de uso de administrar inventario .....	55
Gráfico 12 Caso de uso administrar productos .....	59

## INTRODUCCIÓN

Cuando se delibera sobre el fundamento del uso de computadoras en las empresas o instituciones, la respuesta idónea parece bastante obvia, para registrar los eventos que van aconteciendo.

Sin embargo existe un motivo aún mucho más importante y no tan obvio, y es que la información puede ser utilizada como respaldo en la toma de decisiones.

Sin importar el tamaño o rubro de una empresa o institución, la administración exitosa de la misma estará dramáticamente influenciada por la precisión de sus datos y la adopción de decisiones acertadas. Los enlaces de comunicación y las bases de datos permiten el acceso a recursos de información que están más allá de la inmediatez física, sobrepasando las limitaciones geográficas.

Las computadoras consienten la utilización de incalculables masas de información las cuales, no eran concebibles hasta hace unos años. Pero no sólo basta la disponibilidad de la cantidad de información, se trata de contar con información de calidad. Los sistemas de información basados en computadoras no sólo son capaces de suministrar información de calidad oportunamente, sino que también consiguen respaldar en la toma de decisiones de la empresa o institución.

El restaurante Falcón ubicado en el Jirón 2 de Mayo, número 190, ubicado en la ciudad de Huánuco, es una empresa que ofrece una gastronomía de platos típicos líder en la ciudad, reconocida por ofrecer una variada carta de comida típica huanuqueña.

El sistema de información que se ha realizado para el Restaurante Falcón, con un conjunto de elementos orientados a generar una solución óptima utilizando diversas herramientas que permiten resolver las dificultades que presentan los procesos desarrollados en el establecimiento tales como: Pedidos por parte del cliente, visualización de estos en la cocina y control de inventarios. Como primera medida se establece la manera como se deben tomar los requerimientos del sistema y se define claramente lo que

la empresa necesita, de allí se han tomado decisiones en cuanto a diseño y factibilidad del sistema que se implementó. La información en esta etapa se obtiene mediante entrevistas con los socios del Restaurante Falcón y observación directa, actuando como clientes en el propio establecimiento.

En la etapa de diseño, se modelan los módulos que harán parte de la integración total del sistema, así como la base de datos y la distribución de los módulos, es importante destacar que la presencia del cliente en esta etapa del proyecto es de vital importancia ya que se define la interfaz gráfica que tendrá el software.

Finalmente se instaló el sistema de informático y se da una inducción completa al gerente y operarios de la empresa para que aprendan el manejo adecuado del sistema de información

## RESUMEN

El presente trabajo de suficiencia profesional tiene como objetivo principal el de Implementar un sistema informático para mejorar la calidad del servicio en el RECREO FALCON, asimismo tiene como objetivos específicos la Mejora de la gestión de ventas, caja y almacén en el RECREO FALCON.

Asimismo, el presente trabajo de suficiencia profesional es de Tipo Aplicada o Tecnológica. No aplicable a hipótesis, por el tipo de investigación que se usa, y culmina con la conclusión que se mejoró en la gestión de ventas, caja y almacén es más que evidente que se mejoró la calidad en el servicio en el RECREO FALCON.

Se recomienda con respecto a la atención al cliente, ya que el sistema ha mejorado la velocidad de atención con el cliente se le recomienda al personal una pequeña capacitación de atención para así apoyar a que el sistema cumpla la función para la que ha sido creada.

Palabras claves: Sistema informático, desarrollo de software, metodología ICONIX.

## **SUMARY**

The main objective of this work is to implement a computer system to improve the quality of service in the RECREO FALCON, also has as specific objectives the improvement of the management of sales, collection and storage in the RECREO FALCON.

Also the present work of professional sufficiency is of Applied or Technological Type. Not applicable to hypothesis, because of the type of research used, and culminates in the conclusion that it was improved in the management of sales, collections and warehouse it is more than evident that the quality of the service in the RECREO FALCON was improved.

It is recommended with respect to customer service, since the system has improved the speed of customer service, it is recommended that staff receive a small training to support the system to fulfill the function for which it was created.

Keywords: Information system, software development, ICONIX methodology.

## **CAPÍTULO I**

### **ASPECTOS DE LA ENTIDAD RECEPTORA**

#### **1.1 NOMBRE O RAZÓN SOCIAL**

RECREO FALCON

#### **1.2 RUBRO**

RESTAURANTE

#### **1.3 UBICACIÓN / DIRECCIÓN**

JR. DOS DE MAYO. #190 – HUÁNUCO.

#### **1.4 RESEÑA**

Recreo Falcon, ofrece platos típicos de la región Huánuco, como Pachamanca, Chancho al cilindro, cuy frito, chicharrón, trucha, caldo de gallina, además de bebidas, tragos, postres, entre otros.

#### **VISIÓN**

Busca ser líder a nivel regional, brindando un servicio de calidad a través del buen sabor de las comidas que lo caracteriza, y manteniendo siempre la capacitación constante de sus colaboradores.

#### **MISIÓN**

Somos una empresa gastronómica que ofrece una variedad de platos típicos de la región, comprometidos con el buen sabor de las comidas para satisfacer a todos nuestros clientes, ofreciendo un ambiente acogedor para su comodidad.

## **CAPITULO II**

### **ASPECTOS DEL ÁREA O SECCIÓN**

El Recreo Falcon tiene como definición de calidad de servicio al cliente el cumplir con los requisitos que exige el cliente, ya que el satisfacerlos debe ser parte fundamental de la filosofía del negocio y el enfoque central del plan estratégico de toda la empresa ya que el mejorar continuamente el producto y el servicio de calidad significa el elemento clave del éxito de la empresa.



## **CAPITULO III**

### **IDENTIFICACIÓN DE LA SITUACIÓN PROBLEMÁTICA**

#### **3.1. PROBLEMA DE INVESTIGACIÓN**

##### **3.1.1. DESCRIPCIÓN DEL PROBLEMA**

En Huánuco existen una gran cantidad de restaurantes entre los cuales están los recreos, de comida típica y especializada.

En los recreos el costo va de acuerdo al servicio y la calidad de los platos que se consumen. El servicio, la decoración, la ambientación, comida y bebidas son esmeradamente escogidos. Así mismo, para perfeccionar la atención a sus clientes se requiere que:

- Los alimentos sean cocinados al momento, por lo que es necesario tener la información de los pedidos lo más pronto posible.
- El cliente observe el listado de pedidos que ha realizado y el costo total de los mismos.
- Sea más ágil el proceso de pagos, por lo que se requiere que pueda conocer el valor de su consumo rápidamente.

Actualmente los recreos de Huánuco tienen muchas exigencias en cuanto a dar un buen servicio, como por ejemplo que el cliente se sienta cómodo al realizar un pedido, esto muchas veces no se da debido a que los meseros no se abastecen en atender rápidamente a las mesas, además de que se toman las órdenes manualmente para después ir a la cocina y dar a conocer el pedido realizado por el cliente. De esta manera, el proceso lleva mucho tiempo y más cuando el recreo está lleno.

Para resolver la problemática presentada, se propone el Desarrollo de un Sistema informático para la gestión de ventas (comidas y otros enseres como gaseosas, cervezas, etc.), la gestión de caja a los comensales (se realiza los pedido de las mesas correspondientes y posteriormente son atendidos), la gestión de almacén: (ingreso y salida de los productos para la elaboración de las comidas y otros enseres

como gaseosas, cervezas, etc.), mismo que permitirá gestionar los pedidos de una manera rápida, segura y amigable con el cliente.

### **3.1.2. FORMULACIÓN DEL PROBLEMA**

#### **3.1.2.1. PROBLEMA PRINCIPAL**

¿De que manera implementar un sistema informático de caja y almacén para el RECREO FALCON?

#### **3.1.2.2. PROBLEMAS ESPECÍFICOS**

PE1. ¿Cómo implementar un Subsistema de gestión de caja en el RECREO FALCON?

PE2. ¿Cómo implementar un Subsistema de gestión de almacén en el RECREO FALCON?

### **3.1.3. OBJETIVOS**

#### **3.1.3.1. OBJETIVO GENERAL**

Implementar un sistema informático de caja y almacén para la gestión en el RECREO FALCON.

#### **3.1.3.2. OBJETIVOS ESPECÍFICOS**

OE1. Implementar un Subsistema de gestión de caja en el RECREO FALCON.

OE2. Implementar un Subsistema de gestión de almacén en el RECREO FALCON.

### **3.1.4. JUSTIFICACIÓN**

El sistema a desarrollar no tiene una alta complejidad, el código debe ser sencillo, tanto que pueda modificarse fácilmente, las iteraciones deben ser presentadas en corto tiempo y requiere ser hecho en el menor tiempo posible. Técnicamente, no se requiere abundante documentación a lo largo del proceso de desarrollo.

### 3.1.5. HIPÓTESIS

No aplicable, por el tipo de investigación.

### 3.1.6. VARIABLES

#### 3.1.6.1. VARIABLE INDEPENDIENTE

✓ Sistema informático de caja y almacen.

#### 3.1.6.2. VARIABLE DEPENDIENTE

✓ Gestión del RECREO FALCON.

### 3.1.7. OPERACIONALIZACIÓN DE VARIABLES

Variables	Dimensiones	Indicadores
Sistema informático de caja y almacen	Funcionalidad	Programación funcional
		Interfaz adecuada
		Tiempo de atención
		Tiempo de búsqueda
	Seguridad	Control de accesos Respaldo de datos(Backups)
Gestión del RECREO FALCON	Registro	Tiempo de respuesta de registros
	Búsqueda	Tiempo de respuesta de búsquedas
	Alertas	Tiempo de Secion para usuarios

### 3.1.8. LIMITACIONES

No se observaron limitantes significativas para el desarrollo de la investigación ya que se tomará en cuenta al personal que labora en el RECREO FALCON por lo tanto se tiene todas las condiciones necesarias para cumplir satisfactoriamente con los objetivos trazados.

### **3.1.9. VIABILIDAD**

#### **3.1.9.1. RECURSOS HUMANOS:**

El presente trabajo de investigación es viable porque el personal intervendrá directamente en el diseño e implementación del sistema informático.

#### **3.1.9.2. RECURSOS ECONÓMICOS:**

Se dispondrá de recursos económicos para cubrir con los objetivos trazados al realizar la implementación del sistema informático.

#### **3.1.9.3. RECURSOS TECNOLÓGICOS (MATERIALES)**

Se dispone de recursos materiales para para cumplir con los objetivos trazados al realizar la implementación del sistema informático.

### **3.2. MARCO TEORICO**

#### **3.2.1. ANTECEDENTES DE LA INVESTIGACIÓN**

##### **3.2.1.1. INTERNACIONALES**

- a) **ERNESTO ZEPEDA GÓMEZ (2013) EN LA INVESTIGACIÓN “ANÁLISIS DE LA CALIDAD EN EL SERVICIO DE UN RESTAURANTE FAMILIAR EN EL ESTADO DE CHIAPAS”- ESCUELA DE NEGOCIOS - DEPARTAMENTO DE ADMINISTRACIÓN DE HOTELES Y RESTAURANTES UNIVERSIDAD DE LAS AMÉRICAS PUEBLA - CHOLULA, PUEBLA, MÉXICO. SE LLEGÓ A LAS SIGUIENTES CONCLUSIONES:**

Se identificó por medio de cuestionarios las deficiencias en diversas áreas del restaurante, Los Faroles según la percepción del cliente. El mayor número de deficiencias se encontró en el servicio que se recibió. La gente expreso su opinión en cuanto a que hace falta una mejor capacitación para los meseros que no saben actuar de manera adecuada a las circunstancias que se presentan.

La sazón de los alimentos también recibió críticas muy variadas, mientras hay gente a la que le gusta mucho la comida del restaurante hay otros que no quedan del todo satisfechos. El motivo de que muchos regresen es porque les gusta el establecimiento en general.

El diseño exterior recibió muy buenas críticas. A la gente le gusta que cuente con estacionamiento amplio y tenga una buena ubicación.

**b) CARRILLO, SEBASTIAN ORTIZ (2012) EN LA INVESTIGACIÓN “PROGRAMAS DE CALIDAD EN SERVICIO EN RESTAURANTES DE LUJO Y PRIMERA CATEGORÍA DE LA CIUDAD DE GUAYAQUIL” ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL FACULTAD DE INGENIERÍA MARÍTIMA Y CIENCIAS DEL MAR – GUAYAQUIL, ECUADOR. SE LLEGÓ A LAS SIGUIENTES CONCLUSIONES:**

Se válida el método utilizado cuando el mejoramiento de la calidad percibida a través de las medidas correctivas implementadas es afectado cuando una de las variables de investigación se ve alterada por una circunstancia externa incontrolable (falta de agua potable en el local) y resulta ser valorada en grados significativamente negativa.

A pesar de que un programa de calidad de servicio es solamente un proceso de capacitación, se puede concluir que la implementación de un pequeño modelo de instrucción a meseros afecta significativamente la percepción de calidad que tienen los clientes.

El trabajo realizado en la implantación del programa de calidad generó en el investigador un afecto especial por el restaurante al trabajo en pos de mejorar, encontrando las debilidades del servicio y generando medidas que nos acerquen más a la calidad deseada

### **3.2.1.2. NACIONALES**

#### **a) NICHOR MIGUEL TORRES PILCO (2011) EN LA INVESTIGACIÓN: “IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA EL RESTAURANTE CEVICHERÍA PATY’S”. SE LLEGÓ A LAS SIGUIENTES CONCLUSIONES:**

Para el desarrollo del sistema de información, se realizó el análisis de las necesidades y procesos del área de ventas del Restaurante, los cuales permitieron un conocimiento mejor de los procesos para sistematizarlo y trabajar de manera eficiente y en tiempo real.

Se logró realizar la implementación del sistema de información, utilizando la metodología de desarrollo RUP en el Restaurante Cevichería Mary's de la Ciudad de Huaraz.

Se logró diseñar el sistema de información, utilizando el lenguaje de programación Java y la Base de Datos SQL Server en el Restaurante Cevichería Mary's de la Ciudad de Huaraz, la cual permitió garantizar la mejora del nivel y calidad del servicio de atención a los clientes.

#### **b) JOEL ANDRÉS SARAZA GRANDE (2014) EN LA INVESTIGACIÓN:" IMPLEMENTACIÓN DE UN SISTEMA VÍA WEB CON APLICACIÓN MÓVIL PARA LA RESERVA Y PEDIDOS EN LÍNEA DE RESTAURANTES” , LLEGO A LA CONCLUSIÓN:**

El sistema implementado cumple con la funcionalidad de todo el proceso que se realizaba para registrar una reserva.

El sistema implementado ayuda al jefe de reserva para estar al tanto de las reservas que están cerca para su atención.

Se logró dar facilidad al cliente, para que tenga una nueva vía para realizar su reserva.

La integración de la implementación del sistema web y móvil permite una nueva gama de posibilidades para ofrecer un mejor servicio de calidad.

El sistema móvil facilita el marketing del restaurante a la hora de ofrecer todos sus platillos el cual permite ganarse con el tiempo la fidelización de los clientes.

### **3.2.2. BASES TEÓRICAS**

#### **3.2.2.1. Sistema de información**

#### **3.2.2.2. Concepto:**

Según Bertalanffy, L. V. (1976).

El término sistema de información (SI) en la actualidad tiene dos planteamientos: el primero y más común emparentado con artefactos tecnológicos fundados para recibir, almacenar, procesar y ofrecer información. El segundo concierne con el manejo organizado de la información en las organizaciones (ACM, AIS, IEEE-CS en Olave 2005).

Los sistemas de información se encargan también de convertir los datos en información y se precisa como un "conjunto formal de procesos operando sobre una colección de datos, distribuidos de acuerdo con las necesidades de la organización, que recopilan, elaboran y distribuyen la información necesaria para las acciones de dicha organización y para las acciones de dirección y control convenientes".

No existe una sola definición consensada sobre sistema de información:

Definición general de los sistemas de información. Según (Laudon K, Laudon J): "conjunto de componentes interrelacionados que recolectan (o recobran), procesan, acopian y distribuyen información para apoyar a la toma de decisiones y el control de una organización".

Definición basada en la tecnología de la información, "conjunto de personas, procesos, datos y tecnología de la información que

interactúan para acopiar, procesar, acopiar y proveer la información ineludible para el educado funcionamiento de la organización”.

Definición desde una figura estratégica. “conjunto sensato de procesos que, aplicándose con un conjunto estructurado de datos constituida de acuerdo con las necesidades de la empresa, recopila, elabora y distribuye (parte de) la información necesaria para la evolución de dicha empresa y para las diligencias de dirección de control convenientes, afirmando al menos en parte, la toma de decisiones netamente necesaria para alcanzar desempeñar las funciones y procesos de negocio de la empresa de acuerdo a su estrategia”.

Los sistemas de información “como un acumulado de componentes que interactúan entre ellas para alcanzar un objetivo común: satisfacer todas las necesidades de información de una organización”.

Fundamentándose en los conceptos exhibidos, para efectos de este trabajo de investigación explotaremos la siguiente definición de sistema de información: “un sistema de información es un conjunto de personas, procesos, datos y tecnología de la información (*hardware*, *software*, medios de almacenamiento y/o base de datos) que reciben datos de entrada, producen procesamientos ya puntualizados y conciben información de salida o cualquier otro resultado demandado, la cual se presenta a través de distintos medios de difusión y que será utilizada para las tomas de decisiones en la organización.”

Los sistemas de información están constituidos por subsistemas: software, hardware, medios de almacenamiento y/o base de datos; y a todo este acumulado de subsistemas se les designa aplicación de sistemas de información; de tal forma que los sistemas de información logran tener aplicaciones en ventas, contabilidad, inventarios, compras y cualquier otro proceso en la organización.

La intención de los sistemas de información, cuyas prestancias son: entrada, procesamiento y salida de datos. Además, consideran



que la retroalimentación sirve para mejorar o controlar el funcionamiento del sistema en la organización.

Un principio primordial en el progreso de los sistemas de información por las empresas, es que las aplicaciones son una herramienta mas no un instrumento que debe de poseerse para manejar la tecnología de la información; en secuela el sistema de información debe de desarrollarse en base a su propia capacidad para perfeccionar el desempeño de la organización, sin embargo, estas razones no simbolizan exclusivamente pérdidas y ganancias. La marcha de una empresa encierra también beneficios para todos sus empleados, clientes y otras personas con la que se tienen sociabilidades.

### **3.2.2.3. Clasificación de los sistemas de información**

Existen diversos criterios para clasificar a los sistemas de información:

- Por el grado de su formalidad, que a la vez se fraccionan en formales e informales.
- Por el nivel de automatización alcanzado, poco automatizados o muy automatizados
- Por su relación con la toma de decisiones
- Por la naturaleza de sus entradas y salidas
- Por el origen y el grado de personalización
- Por el valor que representan para la organización

Los Sistemas de Información tienen el objetivo de dar soporte a las decisiones mediante información actualizada y concreta de la organización.

Los Sistemas de Información que consiguen la automatización de procesos operativos dentro de una organización, son los Sistemas Transaccionales siendo el más generalizado universalmente. Son aquellos que se facultan de manera concreta de procesar tanto las transacciones de información inducidas por las interacciones formales

entre el entorno y la organización como las transacciones forjadas en el seno de la información. Mientras que los sistemas de apoyo a las decisiones facilitan el acceso y la preparación de datos relevantes, ofrece capacidad de cálculo.

Los sistemas de información se clasifican en subsistemas directivos y en subsistemas funcionales. Los subsistemas funcionales se describen en función de las actividades que se ejecutan en las distintas áreas funcionales de la empresa.

#### **3.2.2.4. Metodologías de desarrollo**

##### **3.2.2.5. Definición**

Las metodologías de desarrollo de software son un acumulado de técnicas, procedimientos y ayudas a la documentación para la generación de productos software.

Actualmente es ineludible considerar los peligros, aunque regularmente las empresas, no han sido sensibilizadas de los riesgos innatos al procesamiento de la información por intermedio de ordenadores, a lo que han aportado, a veces, los propios responsables de informática, que no han sabido exponer con la suficiente elocuencia las secuelas de una política de seguridad insuficiente o incluso inexistente. Por otro lado, debido a una cierta imperfección profesional en la aplicación de los razonamientos de coste/beneficio, el directivo inexperto de la informática no acostumbra a facultar inversiones que no lleven implícitamente un beneficio justificable, tangible y conmensurable.

Las técnicas prueban cómo debe ser efectuada una actividad técnica decretada identificada en la metodología. Combina el empleo de unos modelos o formas gráficas junto con el manejo de unos procedimientos puntualizados. Se debe asumir en consideración que una técnica establecida puede ser manejada en una o más actividades de la metodología del desarrollo de software. Conjuntamente se debe tener mucho cuidado cuando se pretende cambiar una técnica por otra.

Las principales metodologías ágiles son las siguientes:

- a. Scrum
- b. Kanban
- c. XP
- d. ICONIX

### **3.2.2.6. Metodología ICONIX**

Se eligió esta metodología porque se realizó un curso de programación y me enseñaron a realizar dicha metodología así que con la finalidad de practicar lo aprendido se decidió el uso de esta metodología.

Es una metodología de desarrollo de software, basada en la complejidad de análisis de la metodología RUP y la practicidad para el desarrollo de la metodología XP

Iconix se guía a través de casos de uso y sigue un ciclo de vida interactivo e incremental.

Es una metodología pesada-ligera de Desarrollo del Software que se halla entre RUP (Rational Unified Process) y XP (eXtreme Programming), unifica un conjunto de métodos de orientación a objetos con el objetivo de tener un control estricto sobre todo el ciclo de vida del producto a realizar.

Es una herramienta que nos permitiera capturar el Qué hacer y luego ayudarnos a decidir Cómo hacerlo.

Qué hacer : Análisis

Cómo hacerlo : Diseño

Fue elaborado por Doug Rosenberg y Kendall Scott a partir de una síntesis del proceso unificado de los “tres amigos” Booch, Rumbaugh y Jacobson y que ha dado soporte y conocimiento a la metodología ICONIX desde 1993. Presenta claramente las actividades de cada fase y exhibe una secuencia de pasos que deben ser seguidos. (Castillo, Figueroa, & Sevilla)

### **3.2.2.7. Características**

ICONIX cuenta con tres características fundamentales:

- **Iterativo e Incremental:** durante el desarrollo del modelo del dominio y la definición de los casos de uso se producen varias iteraciones. El ciclo de vida incremental consiste en desarrollar por partes el producto de manera que puedas integrarlas funcionalmente. Ciclo de vida Iterativo, en cada ciclo de iteración se revisa y mejora el producto.  
El desarrollo se organiza en series de mini-proyectos cortos, llamados iteraciones.
- **Trazabilidad:** Cada paso que se realiza está definido por un requisito, se define la trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos de software producidos.
- **Dinámica del UML:** Ofrece un uso dinámico del UML porque utiliza algunos diagramas UML, sin exigir la utilización de todos, como en el caso de RUP (Rational Unified Process). (Castillo, Figueroa, & Sevilla).

### **3.2.2.8. Fases de la metodología ICONIX**

Según ITSA. (2008)

#### **Revisión de los requisitos/ Análisis de Requisitos**

En esta fase se deben analizar todos los requisitos que formaran parte del sistema y con estos construir el diagrama de clases, que representa las agrupaciones funcionales que estructuraran el sistema en desarrollo.

Para esta fase se utilizan 3 herramientas:

- **Modelo de Dominio:** esto se refiere a identificar objetos y cosas del mundo real que intervienen con nuestro sistema. (Estático)

- Modelo de Casos de Uso: describe las acciones o el comportamiento que un usuario realiza dentro del sistema. Comprende de actores, casos de uso y el sistema.
- Prototipo de Interfaz de Usuario: implica la creación de un modelo o modelos operativos del trabajo de un sistema, en el que analistas y clientes deben estar de acuerdo. (Dinámico/ los usuarios se hacen participantes activos en el desarrollo)

### **Revisión del diseño preliminar /Análisis y Diseño Preliminar**

En esta fase a partir de cada caso de uso se obtendrán una ficha de caso de uso, está formada por un nombre, una descripción, una precondición que debe cumplir antes de iniciarse, una pos-condición que debe cumplir al terminar si termina correctamente.

**Realizar Diagrama de Robustez:** es un híbrido entre un Diagrama de Clases y un Diagrama de Actividades. Es una herramienta que nos permite capturar el Que hacer y a partir de eso él Como hacerlo. Facilita el reconocimiento de objetos y hace más sencilla la lectura del sistema.

El diagrama de Robustez se divide en:

- Objetos fronterizos: usado por los actores para comunicarse con el sistema.
- Objetos entidad: son objetos del modelo del dominio.
- Objetos de Control: es la unión entre la interfaz y los objetos de entidad.
- Diagrama de Clases: describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

### **Revisión crítica del diseño/Diseño**

En esta fase se registran todos los elementos que forman parte de nuestro sistema.

Diagramas de Secuencia: muestra los métodos que llevarán las clases de nuestro sistema. Muestra todos los cursos alternos que pueden tomar todos nuestros casos de uso. Se debe terminar el modelo estático, añadiendo los detalles del diseño en el diagrama de clases y verificar si el diseño satisface todos los requisitos identificados.

### **Implementación**

Después de tener el diseño se creará el software; que posteriormente se entregará. Se debe utilizar el diagrama de componentes si fuera necesario para apoyar el desarrollo, es decir mostrar una distribución física de los elementos que componen la estructura interna del sistema. Así como escribir y generar el código.

#### **3.2.2.9. Ventajas de la metodología ICONIX**

- Proceso ágil para obtener un sistema informático.
- Dedicada a la construcción de sistemas de gestión de pequeña y mediana complejidad con la participación de los usuarios finales.

#### **3.2.2.10. Desventajas de la metodología ICONIX**

- Necesita información rápida y puntual de los requisitos, del diseño y de las estimaciones .
- Es una metodología que no debe ser usada en proyectos de larga duración.

#### **3.2.2.11. Lenguaje de programación**

Un lenguaje de programación es un acuerdo entre persona que puede definirse así:

Conjunto de reglas o normas que admiten asociar a cada programa educado un cálculo que estará llevado a cabo por un ordenador (sin ambigüedades).

- Por tanto, un lenguaje de programación es un convenio o acuerdo acerca de cómo se debe interpretar el significado de los programas de dicho lenguaje.
- Muchas veces se confunden los lenguajes con los compiladores, intérpretes o con los entornos de desarrollo de software.

#### **3.2.2.12. Base de datos**

Una base de datos es un sistema informático a manera de almacén. En este almacén se recogen grandes volúmenes de información. Por ejemplo, supongamos que somos una corporación telefónica y queremos tener almacenados los datos personales y los números de teléfono de todos nuestros clientes, que seguramente sean millones de individuos.

Dicha información es de gran volumen de tamaño: estamos hablando de treinta o veinte datos multiplicados por miles si no es que millones de individuos. La arcaica gestión de datos se fundamentaba en archivos informáticos, pero para las necesidades de hoy en día hacen mucha falta sistemas más perfeccionados que son precisamente lo que se denomina sistema de base de datos. Llegamos a la conclusión de que precisaríamos una base de datos para automatizar el acceso a la información y poder acceder a ella de manera fácil y rápida además de poder realizar cambios de una manera más eficiente.

#### **3.2.2.13. Diferencia entre sistema informático y sistema de información**

Según Davis, Gordon M y Olson, Margrethe H. (1994). Cuesta entender al principio qué es un sistema informático y en qué se diferencia de un sistema de información.

Un sistema de información es un concepto más amplio que puede (o no) incluir dentro de un sistema informático.

Pensemos en un sistema de información de una biblioteca donde no hay computadoras. La persona encargada tiene que buscar en

un gran archivador las distintas fichas de los libros cuando recibe un pedido, buscar el libro, etc.

Hay todo un sistema de información (no informático) en este sencillo ejemplo. El sistema de información incluye: el bibliotecario, el archivador con las fichas de los libros, los procedimientos que debe realizar el bibliotecario, los libros, etc.

O sea, el sistema de información incluye las Personas, los Datos, las Actividades y/o técnicas de trabajo y los recursos materiales en general.

Ahora la biblioteca se informatiza (utiliza una computadora y una aplicación relacionada a la biblioteca), ya no será necesario buscar la ficha del libro físicamente en el archivador, sino que lo podrá hacer a través de un programa en la computadora, podrá también imprimir la información, saber la ubicación del libro, etc. Ahora podemos hablar que la biblioteca tiene su sistema informático, que forma parte de su sistema de información.

### 3.2.3. DEFINICIONES CONCEPTUALES

- a) **Almacén:** Conjunto de productos que tiene almacenados un comercio y que están destinados a la venta o al consumo.
- b) **Software:** De acuerdo a, (Alcalde Garcia, 1994), el software es un conjunto de elemento lógicos necesarios para que se pueda realizar las tareas encomendadas al mismo, se puede definir de la siguiente forma: en la parte lógica que dota al equipo físico de capacidad para realizar cualquier tipo de trabajo.
- c) **Hardware:** La Real Academia Española define al hardware como el conjunto de los componentes que conforman la parte material (física) de una computadora, a diferencia del software que refiere a los componentes lógicos (intangibles). Sin embargo, el concepto suele ser entendido de manera más amplia y se utiliza para denominar a todos los componentes físicos de una tecnología.



- d) Codificación:** La codificación es la operación de escribir la solución del problema (de acuerdo a la lógica del diagrama de flujo o pseudocódigo), en una serie de instrucciones detalladas, en un código reconocible por la computadora, la serie de instrucciones detalladas se le conoce como código fuente, el cual se escribe en un lenguaje de programación o lenguaje de alto nivel. (Hernández, 2010).

### **3.3. MATERIALES Y MÉTODOS**

#### **3.3.1. TIPO DE INVESTIGACIÓN**

Según Sánchez Carlessi H. y Reyes Meza C. (2006), La investigación Aplicada o Tecnológica, se utiliza la tecnología para luego aplicarla a través de un sistema informático y así optimizar los procesos dentro de una empresa o institución.

En ese sentido el presente trabajo de suficiencia profesional es de Tipo Aplicada o Tecnológica.

## CAPITULO IV

### APORTES PARA LA SOLUCIÓN DEL PROBLEMA

#### 4.1. ANÁLISIS DEL SISTEMA PROPUESTO

En la actualidad el recreo Falcón realiza sus operaciones diarias con papeles y no computarizado, donde suelen presentarse errores en el registro de pedido que genera el mozo, en caja se confunde al cobrar por mal envío de pedido del mozo y esto ocasiona atrasos en el cobro, perdida de dinero en caja, y errores de pedido, lo que perjudica la atención al cliente.

Para el control de caja y egresos suelen usar un cuaderno donde anotan los ingresos y egresos del día, pero demoran de 30 min a 1 hora en cuadras caja dependiendo del número de pedidos generados y cobrados en la empresa.

<b>Sistema de Información</b>	<b>Funciones</b>
NINGUNO	NINGUNO

Tabla 1 CONTROL DE CAJA Y EGRESOS

#### DEFINICIÓN DEL SISTEMA “ CAJA”

##### OBJETIVO.

Permitir control en general todos los ingresos y gastos de las empresas, y poder tener un stock real de la mercadería que gestionan actualmente.

##### ALCANCES DEL SISTEMA

**DENTRO DE LOS PRINCIPALES ALCANCES DEL SISTEMA SE TIENEN LOS SIGUIENTES:**

- Reportes de ingresos del día, mes
- Control de caja en ingresos y egreso (movimiento diario, mensual, anual)
- Registro de Facturas, ticket y boletas de venta
- Registro de ingreso y salida de almacén
- Cierre de caja por cajero
- Gestión de clientes y personal.

## BENEFICIOS DEL SISTEMA

- Permitir visualizar e imprimir reportes
- Facilidad en el registro de ingresos y gastos
- Mejor control de caja, facturación, meseros.
- Mejor control de inventario

## REQUERIMIENTOS FUNCIONALES

Describe la funcionalidad o los servicios que se espera proveerán el sistema

Número	Requerimiento Funcional
RF01	Registro de clientes, meseros.
RF02	Registro de Ingresos con los siguientes comprobantes(boleta, factura, recibo)
RF03	Registro de egresos del día
RF04	Reporte de egresos
RF05	El encargado de caja solo visualiza sus ventas y cierre de caja.
RF06	Reporte de ingresos grafico por periodo
RF07	Control de stock
RF08	Reporte de movimiento diario, mensual, anual en gastos e ingresos
RF09	Resumen de cierre de caja
RF10	Registro de usuario con perfiles de acceso al sistema
RF11	Permitir el registro, actualización y eliminación de platos y productos.
RF12	Configuración del sistema
RF13	Registro de mesas y categorías
RF14	Reporte de reposición de stock
RF15	Reporte en xls al adsoft
RF16	Actualizar egresos registrados
RF17	Permitir baja de documentos
RF18	Reporte de resumen diario de ventas ( )

Tabla 2 FUNCIONALIDAD

## **REQUERIMIENTOS NO FUNCIONALES DEL PRODUCTO:**

Especifican restricciones al comportamiento del producto.

- El sistema es cliente – servidor, el aplicativo es en escritorio.
- Como Gestor de Base de datos
- Fiabilidad: El sistema deberá recuperarse ante pérdidas de conexión al servidor, por ello en caso ocurra algún error de conexión con éste, el sistema indicará un mensaje de reanudar la conexión y volverá a intentar conectarse automáticamente después de un periodo corto de tiempo (5 minutos), hasta alcanzar la conexión.
- Eficacia: El sistema permitirá el acceso de múltiples usuario simultáneamente
- Portabilidad: funciona en diversas plataformas
- Facilidad de uso: El sistema debe ser fácil de usar y en caso del sistema.

## **IDENTIFICACION Y DEFINICION DE LOS SUBSISTEMAS**

### **SUB SISTEMAS.**

Los subsistemas que integran el sistema de información de control de facturación y caja contemplan el soporte a todos los beneficios indicados anteriormente.

### **LOS SUBSISTEMAS SON:**

- Subsistema de Facturación
- Subsistema de Caja

### **SUB SISTEMA FACTURACION**

#### **OBJETIVO.**

- Facilitar la emisión de comprobantes y controlar los ingresos generados por estos

### **ALCANCES DEL SUB SISTEMA**

- Registro de factura, ticket y boleta
- Resumen diario
- Resumen de bajas

### **FUNCIONES PRINCIPALES.**

El subsistema de facturación debe contemplar las siguientes funcionalidades:

- Registro de clientes
- Selección de forma de pago (efectivo, tarjeta)
- Emisión e impresión de ticket
- Baja de comprobantes
- Reporte de Ingresos y bajas

### **BENEFICIOS DEL SUB SISTEMA**

- Controlar el ingreso y baja generada por el cajero

### **SUB SISTEMA DE CONTROL EGRESOS Y CAJA**

#### **OBJETIVO**

- Permitir controlar el egreso y cierre de caja generado por el cajero

### **ALCANCES DEL SUB SISTEMA**

- Registro de cierre de caja
- Resumen de caja

### **FUNCIONES PRINCIPALES.**

El Subsistema de egresos y caja debe contemplar las siguientes funcionalidades:

- Registro de egresos
- Registro de cierre de caja
- Resumen de caja
- Reporte de egresos

## **BENEFICIOS DEL SUB SISTEMA**

- Resumen de mes vs ingresos y egresos
- Control Detallado de ingresos y gastos de caja

## **SUB SISTEMA DE USUARIO**

### **OBJETIVO.**

- Permitir controlar el acceso al sistema por permisos y perfiles

## **ALCANCES DEL SUB SISTEMA**

**Controlar los accesos al sistema**

### **FUNCIONES PRINCIPALES.**

El Subsistema de usuarios debe contemplar las siguientes funcionalidades:

- Realizar ingresos de usuarios
- Registro de personal
- Asignar permiso a un usuario determinado

## **BENEFICIOS DEL SUB SISTEMA**

- Control total de acceso al sistema

## **IDENTIFICACION DE ACTORES**

### **DESCRIPCIÓN DEL TIPO DE ACTOR**

<b>ACTOR</b>	<b>TIPO</b>
Mesero	Interno
Cajero	Principal
Gerente	Principal

**Tabla 3 DESCRIPCIÓN DEL TIPO DE ACTOR**

## DEFINICION DE ACCESO AL SISTEMA

Actor	Descripción	Acceso al sistema
Mesero	Encargado de generar los pedidos	Pedido por número de mesa
Cajero	Encargado de registrar los egresos, realizar los cobros, y entregar el comprobante al cliente	Realiza la emisión del comprobante, egresos y el cierre de caja
Gerente	Visualiza todos los reportes del sistema	Todo el sistema

Tabla 4 DEFINICION DE ACCESO AL SISTEMA

### 4.2. DISEÑO DE LA BASE DE DATOS

Según Bertalanffy, L. V. (1976). Para el sistema implementado se ha diseñado una base de datos fundamentada en el modelo relacional, la cual se describe a continuación:

- Almacenar existencias de productos en el almacén.
- Almacenar información detallada sobre los productos que se venden en el restaurante.
- Almacenar usuarios con los respectivos permisos de administración sobre el sistema.

### TABLAS

#### Categoría

En esta tabla se almacenan todas las categorías correspondientes a los tipos de productos que se venden.

- **Cod\_categoria:** Es el identificador único de la categoría.
- **Cate\_descripcion:** Es el tipo de producto al que pertenece la categoría.

## **Factura**

En esta tabla es posible almacenar la información correspondiente a la factura con la cual es posible relacionar los pedidos en el módulo visualización.

- **Cod\_factura:** Es el identificador único de factura.
- **Fact\_cajero:** Hace referencia al identificador del usuario.
- **Fact\_total:** El total de la venta registrada en cada mesa o a cada cliente.
- **Fact\_fechaemision:** fecha en la que es realizado y expedido el pedido.

## **Producto.**

En esta tabla se almacena la información correspondiente a la descripción de los productos o comidas rápidas que se venden en el restaurante.

- **Cod\_producto:** Es el número con el que se identifica el producto.
- **Prod\_descripcion:** Es el nombre asignado al producto.
- **Prod\_Precio:** El precio asignado a cada producto.
- **Cod\_categoria:** Permite identificar qué tipo de producto es: comida, bebida o adicional.

## **Almacén**

La tabla de almacén guarda los insumos que se encuentran en la bodega del restaurante, dichos insumos son los que se utilizan para preparar las comidas.

- **Cod\_almacen:** Es el identificador de cada compra o inventario que se realice.
- **Alma\_descripcion:** Nombre del producto.
- **Fecha de ingreso:** Fecha de compra del insumo,

## **Mesa**

Esta tabla hace referencia a los datos contenidos en la mesa en la que es atendido un pedido.



- **Cod\_mesa:** Es el número de la mesa en la que se realiza el pedido
- **Mesa\_descripcion:** Es el nombre de la mesa, generalmente se usa la nomenclatura mesa 1, mesa 2, mesa 3, mesa n, para nombrarla.
- **Mesa\_estado:** Permite controlar las mesas que se encuentran libres.

### **Usuario**

El objetivo de esta tabla es el de almacenar la información correspondiente a todos los usuarios que pueden utilizar el sistema, dicha información hace referencia específicamente al id usuario, contraseña, nombre, apellidos y tipo. Los campos mencionados anteriormente se especifican como sigue:

- **Cod\_usuario:** Hace referencia al número con el que puede ingresar el usuario al sistema.
- **Usua\_clave:** Es aquella con la que puede ingresar el usuario al sistema luego de haber indicado su cod\_usuario.
- **Usua\_usuario:** El nombre del usuario

## MODELO ENTIDAD RELACIÓN

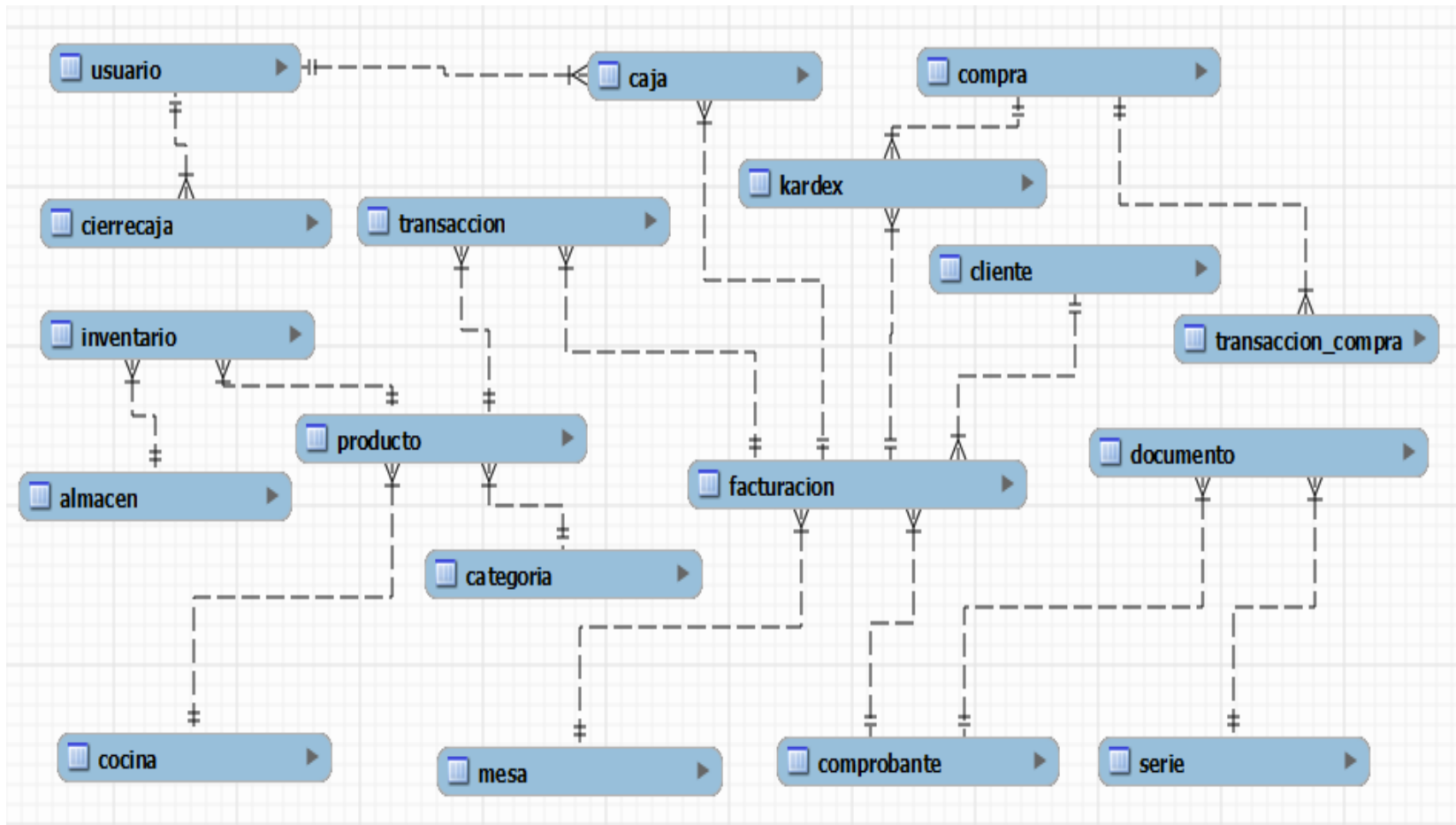


Gráfico 1 MODELO ENTIDAD RELACIÓN



### 4.3. MODELO DE CASOS DE USO

#### DIAGRAMA DE CASOS DE USO POR MODULOS DEL SISTEMA

Gráfico 3 Subsistema de facturación

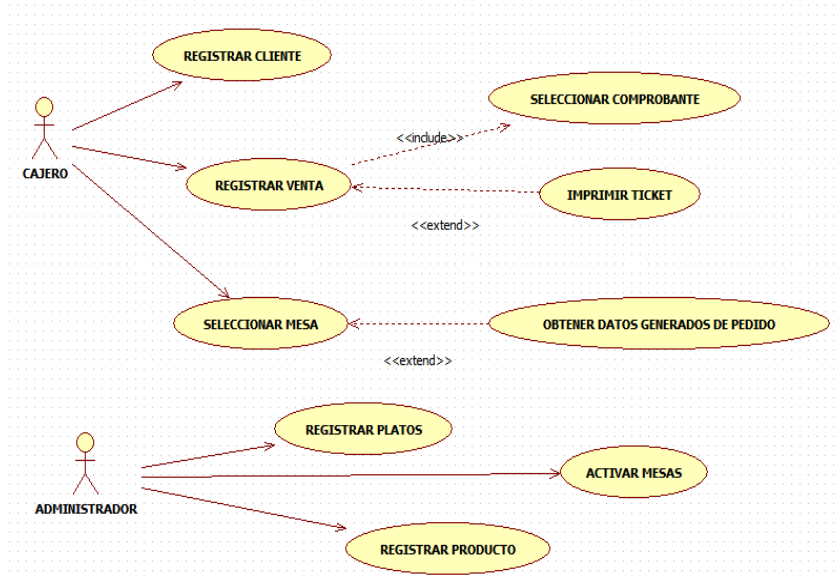


Gráfico 4 Subsistema de egresos y caja

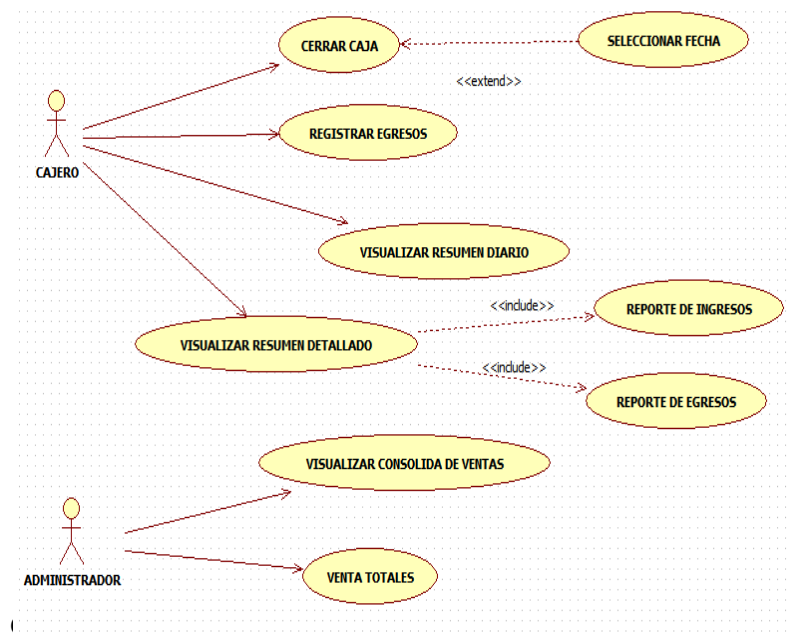


Gráfico 5 Subsistema de Usuarios

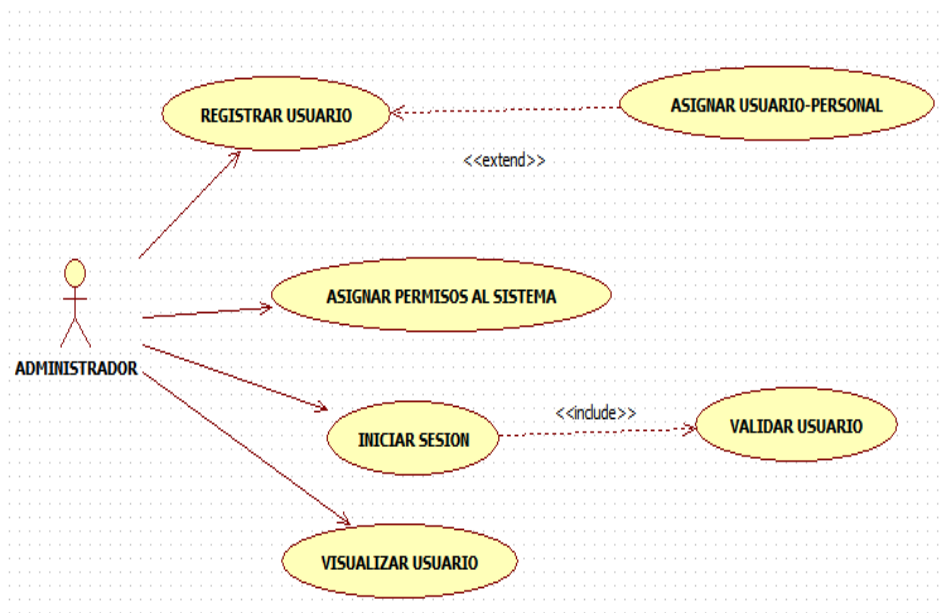
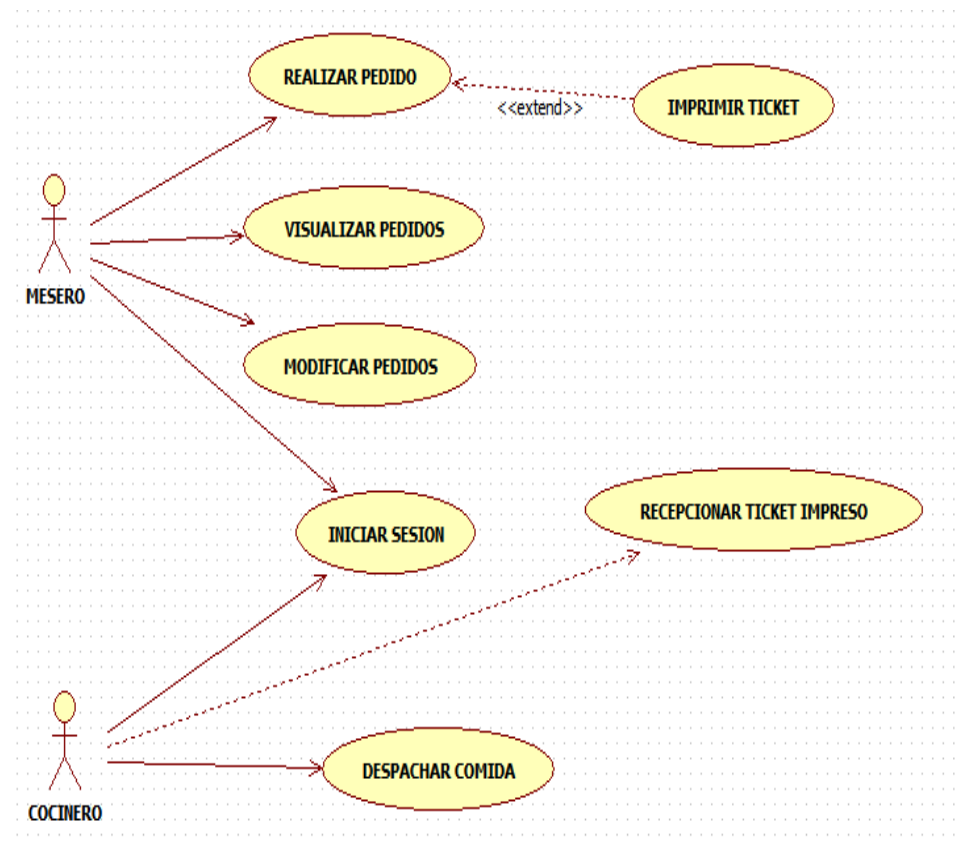
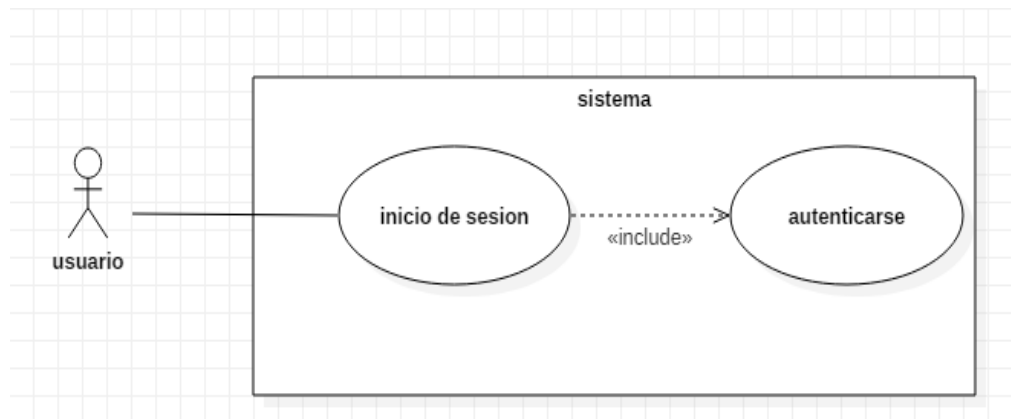


Gráfico 6 Subsistema de Pedidos



## ANÁLISIS Y DISEÑO PRELIMINAR

**Gráfico 7 Descripción de casos de uso de autenticar al usuario**



**Tabla 5 Sección principal autenticar usuario**

Caso de uso	Autenticar usuario
<b>actores</b>	Administrador, Cajero, Mesero
<b>Propósito</b>	Acceder al sistema y autenticarse como usuario
<b>Resumen</b>	Todos los actores mencionados acceden al sistema, indicando su nombre de usuario y contraseña.
<b>Tipo</b>	Esencial.

**Tabla 6 Curso normal de los eventos autenticar usuarios**

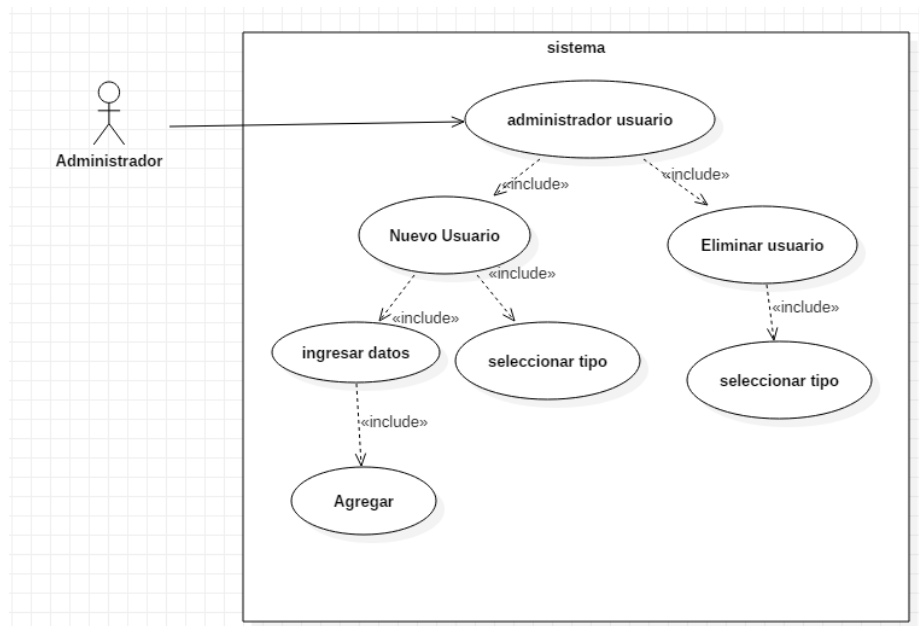
Accion de los actores	Respuesta al sistema
1. En este caso de uso comienza el cuando el usuario ingresa a la pagina inicial del sistema	2. El sistema muestra la pantalla de inicio. <ul style="list-style-type: none"> <li>• <b>Usuario</b></li> <li>• <b>Contraseña</b></li> </ul> Y la opcion ingresar
3. El usuario ingresa los datos correspondientes a su cuenta y	4. El sistema verifica la existencia del usuario y que la contraseña sea correcta

selecciona la opción ingresar.	
	5. Si los datos son correctos, muestra al usuario la pantalla que corresponde a su tipo de cuenta.
6. El usuario puede visualizar el entorno de trabajo correspondiente a su cuenta.	

**Tabla 7 Cursos alternos, autenticar usuarios**

Acción	Descripción
3.	El usuario no ingresa ningún dato y selecciona la opción <b>ingresar</b> , el sistema muestra la pantalla de inicio con un mensaje señalado que los campos son necesarios.
5.	Los datos ingresados son incorrectos, indica el error por medio de un mensaje y reintenta la acción 3.

**Gráfico 8 Diagrama de caso de uso de administrador de usuarios**



**Tabla 8 Especificación de caso de uso de administración de usuarios**

Caso de uso	Administrar usuario
<b>actores</b>	Administrador
<b>Propósito</b>	Permitir al administrador crear nuevas cuentas de usuario para sus empleados, o eliminar de ser necesario.
<b>Resumen</b>	El administrador ingresa al sistema, crea las cuentas de usuario necesario para asignar a sus empleados.
<b>Tipo</b>	Esencial

**Tabla 9 Curso normal de los eventos de administrador de usuarios**

Accion de los actores	Respuesta al sistema
1. En este caso de uso comienza cuando el Administrador ingresa a su cuenta y selecciona la opción mantenimiento, y la sub	2. Abre una ventana con las siguientes opciones: <ul style="list-style-type: none"> <li>• <b>Nuevo usuario</b></li> <li>• <b>Eliminar</b></li> </ul> Y la opcion ingresar



opción usuario o personal .	
<p>3. El administrador escoge la acción que desea realizar sobre las cuentas de usuario.</p> <p>a. Si va a crear una cuenta, vease la sección: <b>guardar usuario</b>.</p> <p>b. Si va a eliminar una cuenta, vease la sección: <b>eliminar usuario</b>.</p>	
<p>4. <b>Utiliza el botón salir</b> para retornar a la ventana principal.</p>	<p>5. El sistema muestra de nuevo la ventana principal del administrador.</p>

**Tabla 10 Cursos alternos, Administrar usuario**

Acción	Descripción
<b>3.</b>	El administrador decide no realizar ninguna acción y presiona la opción <b>salir</b> .

**Tabla 11 Seccion nuevo usuario (curso normal de los eventos)**

Acción de los actores	Respuesta del sistema
1. El administrador selecciona la opción nuevo usuario.	2. Abre una ventana con un formulario que contiene los datos necesarios para registrar el nuevo usuario <ul style="list-style-type: none"> <li>• Personal</li> <li>• DNI</li> <li>• Clave</li> </ul>
3. El usuario ingresa los datos requeridos del formulario y selecciona el tipo de usuario que desea crear. Seguido a esta selecciona “guardar”	4. Envía los datos para ser almacenado.
5.	6. Muestra al usuario la opción <b>salir</b> para ir al menú principal.
7. El usuario selecciona la opción <b>salir</b> .	8. El sistema muestra la ventana del menú principal.

**Tabla 12 Cursos alternos, seccion nuevo usuario**

Acción	Descripción
3.	El administrador no ingresa los datos completos, el sistema muestra información pidiendo ingresar todo los datos, regresa al paso 2.
3.	El administrador no desea agregar un nuevo usuario, y selecciona la opción <b>salir</b> para regresar al menú principal.

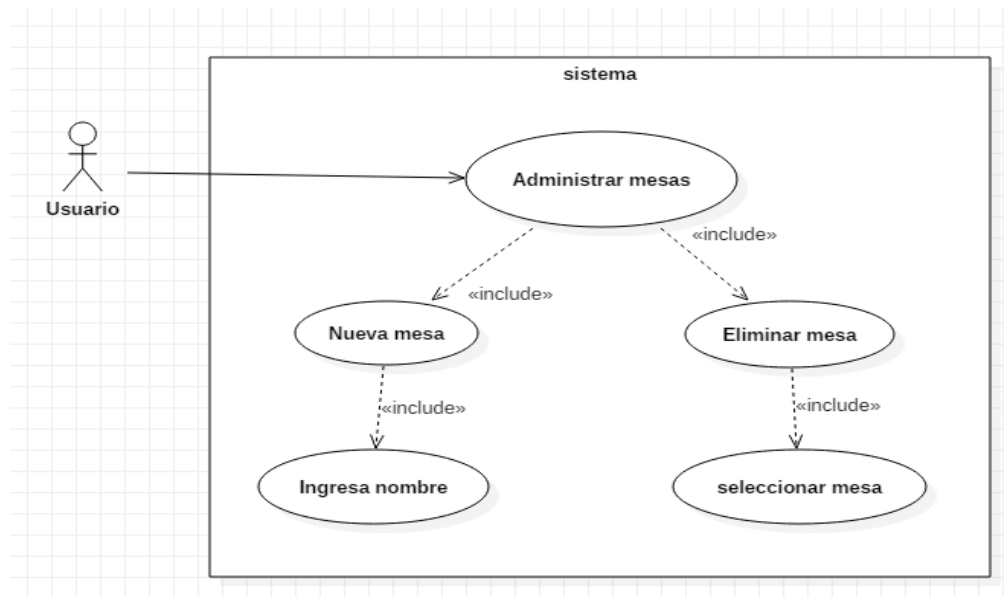
**Tabla 13 Seccion eliminar usuario**

Acción de los actores	Respuesta del sistema
1. El administrador selecciona la opción <b>eliminar</b> usuario.	2. Muestra el menú desplegable.
3. El administrador selecciona el menú desplegable.	4. Muestra un listado con todas las cuentas existentes a excepción de la cuenta del administrador.
5. Escoge una cuenta de la listas.	
6. Selecciona la opción: <ul style="list-style-type: none"> <li>• Eliminar.</li> <li>• Salir( vaya al paso 9)</li> </ul>	7. El sistema elimina la cuenta seleccionada.
	8. El sistema muestra al usuario un mensaje en donde indica que la actividad ha sido realizada con éxito
	9. Muestra al usuario la opción de Salir para ir al menú principal.
10. El usuario selecciona la opción <b>salir</b> .	11.El sistema muestra la ventana principal.

**Tabla 14 Cursos alternos, seccion eleiminar usuario**

Acción	Descripción
3.	El administrador no escoge ninguna cuanta de la lista, el sistema muestra información solicitando seleccionar el campo usuario, regresa al paso 2.
3.	El administrador no desea agregar un nuevo usuario, y selecciona la opción <b>salir</b> para regresar al menú principal.

**Gráfico 9 Diagrama de caso de uso de administrar mesas**



**Tabla 15 Especificación de caso de uso de administrar mesas**

Caso de uso	Administrar mesas
<b>actores</b>	Administrador, Mesero
<b>Propósito</b>	Permitir al usuario crear nuevas mesas en el sistema y eliminarlo cuando se requiera.
<b>Resumen</b>	El usuario ingresa al sistema, seguido selecciona la opción <b>configuración</b> , luego la <b>sub opción mesas</b> y realiza la operación deseada.
<b>Tipo</b>	Esencial

**Tabla 16 Curso normal de eventos de caso de uso de administrar mesas**

Acción de los actores	Respuesta del sistema
<p>1. Este caso de uso comienza cuando el usuario ingresa a su cuenta y selecciona la opción <b>configuración, y la sub opción mesas.</b></p>	<p>2. Abre una venta con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Registrar.</li> <li>• Guardar.</li> <li>• Eliminar.</li> </ul>
<p>3. El usuario escoge la acción que desea realizar sobre la administración de mesas.</p> <p>a. Si va a crear una mesa, véase la opción <b>registrar mesa.</b></p> <p>b. Si va a eliminar una mesa véase la opción: <b>eliminar mesa.</b></p>	
<p>4. Utiliza el botón <b>Salir</b> para retornar a la ventana principal.</p>	<p>5. Regresa a la ventana principal del sistema.</p>

**Tabla 17 Cursos alternativos**

Acción	Descripción
<p><b>3.</b></p>	<p>El usuario no realiza ninguna acción y presiona la opción salir.</p>

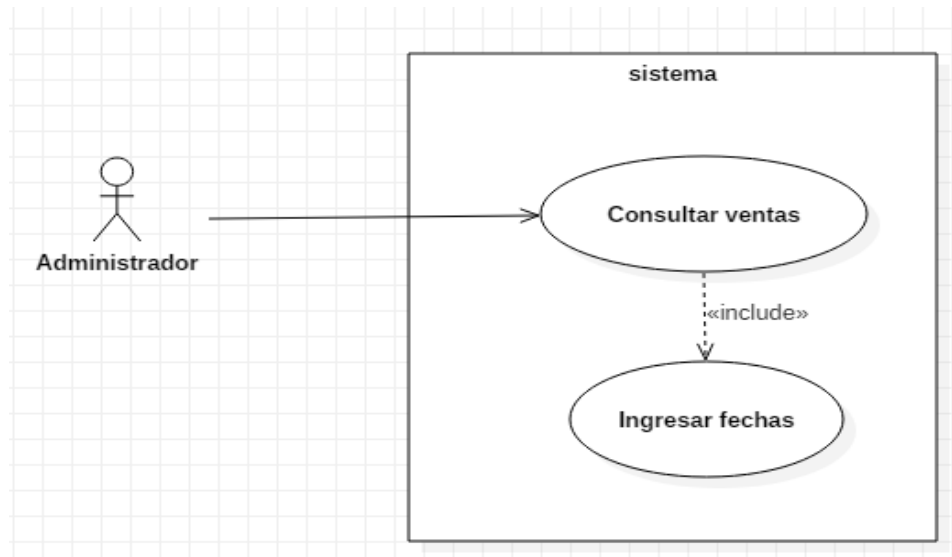
**Tabla 18 Sección administrar nuevas mesas**

Acción de los actores	Respuesta del sistema
1. El usuario selecciona la opción <b>Registrar nueva mesa.</b>	2. Muestra una ventana con las siguientes opciones: <ul style="list-style-type: none"> <li>• Nombre de la mesa y el botón</li> <li>• Guardar.</li> </ul>
3. Registra la información correspondiente al nombre de la mesa	
4. Selecciona el botón guardar.	5. El sistema muestra al usuario un mensaje en donde indica que la actividad ha sido realizado con éxito.
	6. Muestra al usuario la opción de <b>salir</b> para ir al menú principal.
7. El usuario selecciona la opción <b>salir.</b>	8. El sistema muestra la ventana del menú principal

**Tabla 19 Cursos alternos**

Acción	Descripción
1.	El usuario no realiza ninguna acción y presiona la opción salir.

**ráfico 10 Caso de uso consultar ventas**



**Tabla 20 Especificación de caso de uso consultar ventas**

Caso de uso	Consultar ventas
<b>actores</b>	Administrador
<b>Propósito</b>	Permitir al usuario consultar las ventas realizadas en un rango de tiempo requerido.
<b>Resumen</b>	El usuario ingresa al sistema, seguido selecciona la opción <b>ventas</b> , luego la <b>sub opción ventas</b> e ingresa la fecha en la que desea visualizar.
<b>Tipo</b>	Esencial

**Tabla 21 Curso normal de eventos de caso de uso (CU) de consulta de ventas**

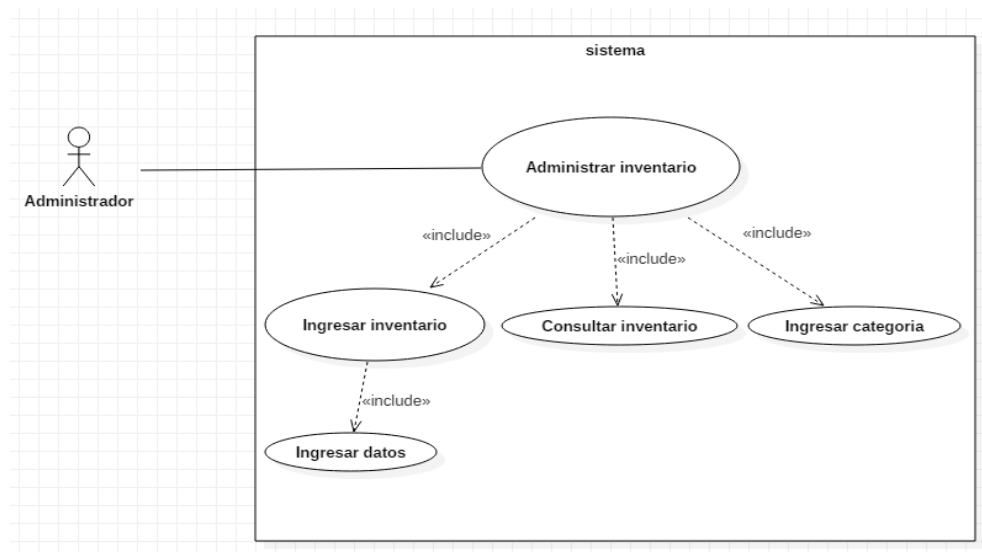
Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el administrador ingresa a su cuenta y selecciona la opción	2. Abre una venta con las siguientes opciones: <ul style="list-style-type: none"> <li>• Consulta (fecha de inicio). Enter</li> </ul>

ventas	
3. El usuario ingresa la fecha que desea consultar con el siguiente formato: día-mes-año.	
4. El usuario selecciona la opción <b>Ventas</b>	5. El sistema realiza una búsqueda de acuerdo a la fecha ingresada por el usuario.
	6. Muestra al usuario la opción de <b>salir</b> para ir al menú principal
7. El usuario selecciona el botón <b>Salir</b>	8. El sistema muestra la ventana del menú principal.

**Tabla 22 Cursos alternativos**

Acción	Descripción
3.	El usuario no realiza ninguna acción y presiona la opción <b>salir</b> .

Gráfico 11 Caso de uso de administrar inventario





**Tabla 23 Especificación de caso de uso administrar inventario**

Caso de uso	Administrar inventario
<b>actores</b>	Administrador
<b>Propósito</b>	Permitir al administrador ingresar datos al inventario, consultar estado actual y agregar categorías para el fácil manejo de los productos.
<b>Resumen</b>	El administrador ingresa al sistema, accede al módulo producto y realiza las opciones deseadas.
<b>Tipo</b>	Esencial

Tabla 24 Curso normal de eventos de caso de uso (CU) administrar inventario.

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el administrador ingresa a su cuenta y selecciona la opción productos	2. Abre una venta con las siguientes opciones: <ul style="list-style-type: none"> <li>• Ingresar inventario.</li> <li>• Consultar inventario</li> <li>• Y la opción salir.</li> </ul>
3. El administrador escoge la sección que desea realizar sobre el inventario. <p>a. Si va a ingresar nuevos insumos véase a la sección: producto.</p> <p>b. Si va a modificar consultar disponibilidad o movimientos de insumos véase a la opción de producto.</p> <p>c. Si va a ingresar nueva categoría de insumos</p>	4. Cierra la ventana de productos y retorna a la ventana principal.

véase a la sección  
ingresar categoría.

**Tabla 25 Cursos alternativos de administrar inventario**

Acción	Descripción
3.	El administrador decide no realizar ninguna acción y presiona la opción <b>salir.</b> , realiza la opción 4

Tabla 26 Curso norma de los eventos sección ingresar inventario.

Acción de los actores	Respuesta del sistema
1. El usuario selecciona la opción ingreso al almacén	2. El sistema muestra un formulario con las siguientes opciones: <ul style="list-style-type: none"> <li>• Comprobante de ingreso.</li> <li>• Proveedor</li> <li>• Producto</li> <li>• Fecha</li> <li>• Motivo</li> <li>• cantidad. Y las opciones agregar, guardar y salir</li> </ul>
3. El usuario selecciona la categoría del insumo e ingresa datos requeridos y selecciona la opción <b>agregar, guardar.</b>	4. Envía los datos para ser almacenados.
	5. El sistema muestra al usuario que la actividad ha sido realizado

	con éxito
	6. Muestra al usuario la opción de salir para ir al menú principal.
7. El usuario selecciona la opción salir.	8. El sistema muestra la ventana del menú principal.

**Tabla 27 Cursos alternativos sección ingresar inventario**

Acción	Descripción
3.	El administrador no ingresa los datos completos y presiona la opción <b>salir</b> , realiza la opción 2
3.	El administrador no desea agregar un nuevo insumo y selecciona la opción salir para regresar al menú principal.

**Tabla 28 Curso norma de los eventos sección ingresar categoría.**

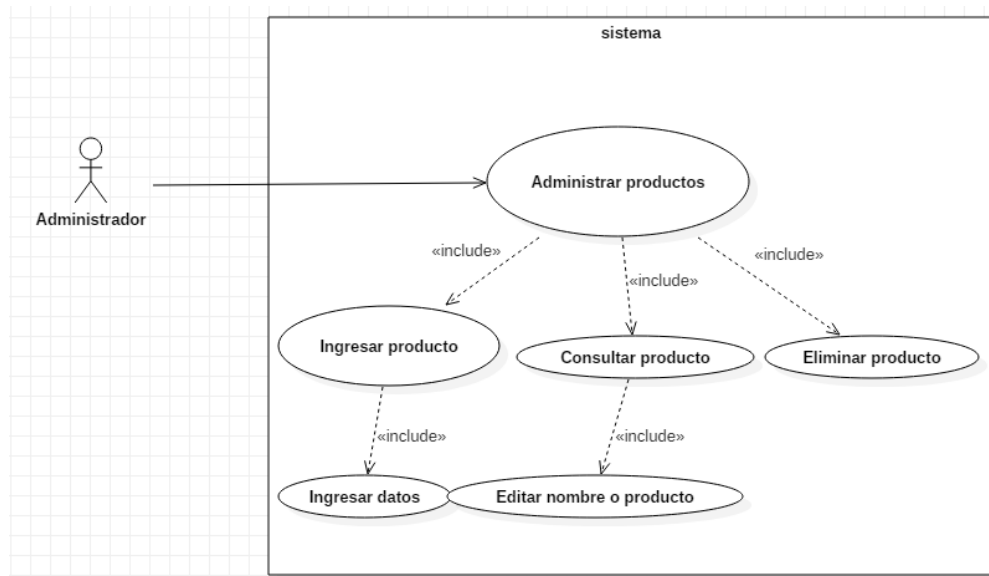
Acción de los actores	Respuesta del sistema
1. El administrador selecciona la opción ingresar categoría	2. Muestra un listado con todas las categorías existentes, un campo nuevo categoría y las opciones agregar y salir.
3. El usuario ingresa el nombre de la nueva categoría que desea crear. Seguido selecciona la opción agregar	4. El sistema almacena el dato y lo agrega a la lista de categorías existentes.
	5. El sistema muestra al usuario que la actividad ha sido realizado con éxito
	6. Muestra al usuario la opción de

	salir para ir al menú principal.
7. El usuario selecciona la opción salir.	El sistema muestra la ventana del menú principal.

**Tabla 29 Cursos alternativos sección ingresar categoría**

Acción	Descripción
3.	El administrador no ingresa ningún dato en el campo nuevo categoría y selecciona la opción agregar, el sistema muestra un mensaje que el campo es requerido. Regresa al paso 2.
3.	El administrador no desea agregar una nueva categoría y selecciona la opción salir para regresar al menú principal.

**Gráfico 12 Caso de uso administrar productos**



**Tabla 30 Sección principal administrar productos**

Caso de uso Administrar productos	
<b>actores</b>	Administrador
<b>Propósito</b>	Permitir al administrador tener el control sobre los productos, facilitando la labor de agregar, modificar y eliminar cada uno de estos.
<b>Resumen</b>	El administrador ingresa al sistema, accede al módulo producto y realiza las opciones deseadas.
<b>Tipo</b>	Esencial

Curso normal de eventos de caso de uso (CU) administrar productos.

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el administrador ingresa a su cuenta y selecciona la opción productos	2. Abre una ventana con las siguientes opciones: <ul style="list-style-type: none"> <li>• Ingresar producto.</li> <li>• Modificar producto</li> <li>• Eliminar producto y la opción salir.</li> </ul>
3. El administrador escoge la sección que desea realizar. <p>a. Si va a ingresar un producto véase la sección: producto.</p> <p><b>b. Si va a modificar véase a la opción de modificar producto.</b></p> <p>c. Si va a eliminar un producto véase a la sección eliminar producto.</p>	4. Cierra la ventana de productos y retorna a la ventana principal.

### Cursos alternativos de administrar producto

Acción	Descripción
3.	El administrador decide no realizar ninguna acción y presiona la opción <b>salir</b> , realiza la opción 4

### Curso normal de los eventos sección ingresar producto

Acción de los actores	Respuesta del sistema
1. El usuario selecciona la opción registrar producto	2. El sistema muestra un formulario con los campos: <ul style="list-style-type: none"> <li>•Nombre</li> <li>•Precio y las siguientes opciones agregar y salir.</li> </ul>
3. El usuario ingresa los datos del producto nombre, precio y selecciona la clase del producto la cual puede ser: comida, bebida o adicional seguida a esto selecciona la opción salir.	4. Envía los datos para ser almacenados.
	5. El sistema muestra al usuario un mensaje en donde indica que la actividad ha sido realizado con éxito.
	6. Muestra al usuario la opción salir para ir al menú principal.
7. El usuario	8. El sistema muestra la ventana del menú principal.

selecciona la  
opción salir.

#### Cursos alternativos sección ingresar producto

Acción	Descripción
3.	El administrador no ingresa los datos completos, el sistema muestra información pidiendo ingresar todos los datos regresa al paso 2.
3.	El administrador no desea agregar un nuevo producto y selecciona la opción salir para regresar al menú principal.

#### Curso normal de los eventos sección eliminar producto

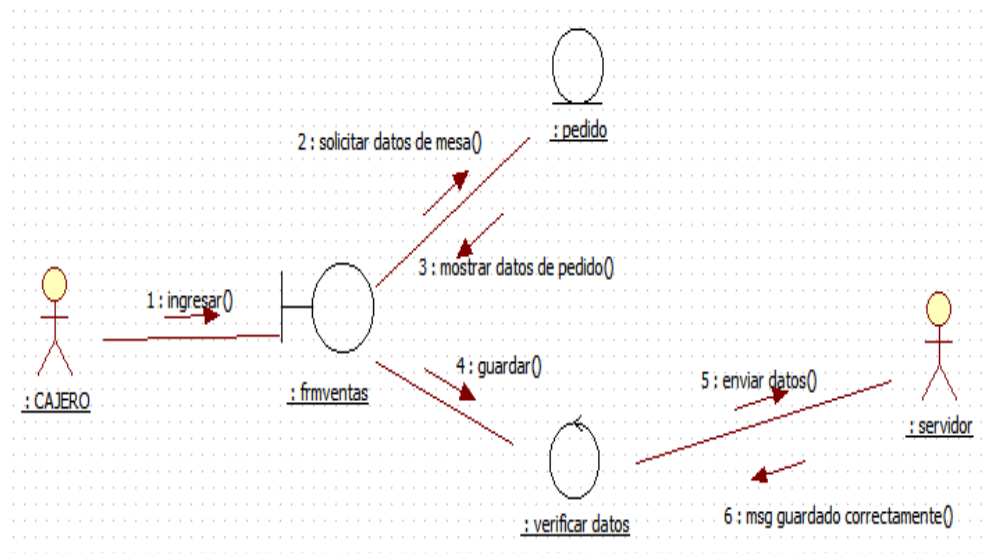
Acción de los actores	Respuesta del sistema
1. El administrador selecciona la opción eliminar producto	2. Muestra una ventana con menú desplegable y las opciones <ul style="list-style-type: none"> <li>• Eliminar producto y el botón salir.</li> </ul>
3. El usuario <b>selecciona el menú desplegable.</b>	4. El sistema despliega una lista de todos los productos que se encuentra almacenados.
5. El usuario elige de la lista el producto que desea eliminar. Seguido la opción eliminar.	6. El sistema elimina el producto y muestra un mensaje en donde indica que la actividad ha sido realizado con éxito.
	7. Muestra al usuario la opción salir

	para ir al menú principal.
8. El usuario selecciona la opción salir.	9. El sistema muestra la ventana del menú principal.

Cursos alternativos sección eliminar productos

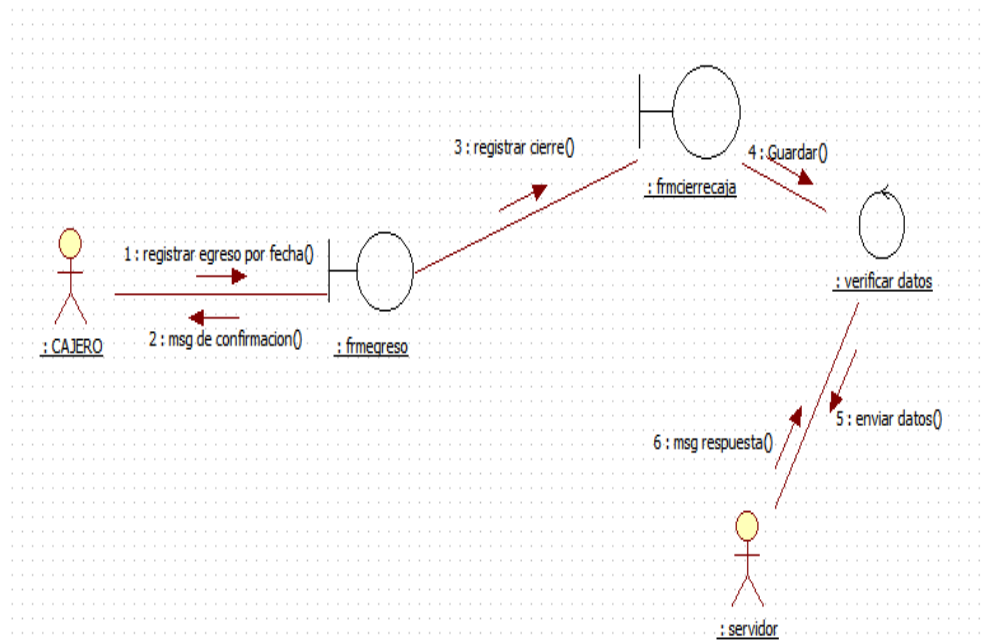
Acción	Descripción
3.	El administrador no elige ningún producto selecciona la opción eliminar producto, el sistema muestra un mensaje que el campo es requerido, regresa al paso 2.
3.	El administrador no desea eliminar ninguno de los productos listados y selecciona la opción salir para regresar al menú principal.

**4.4. DIAGRAMA DE ROBUSTEZ POR MÓDULOS  
SUBSISTEMA DE FACTURACION**

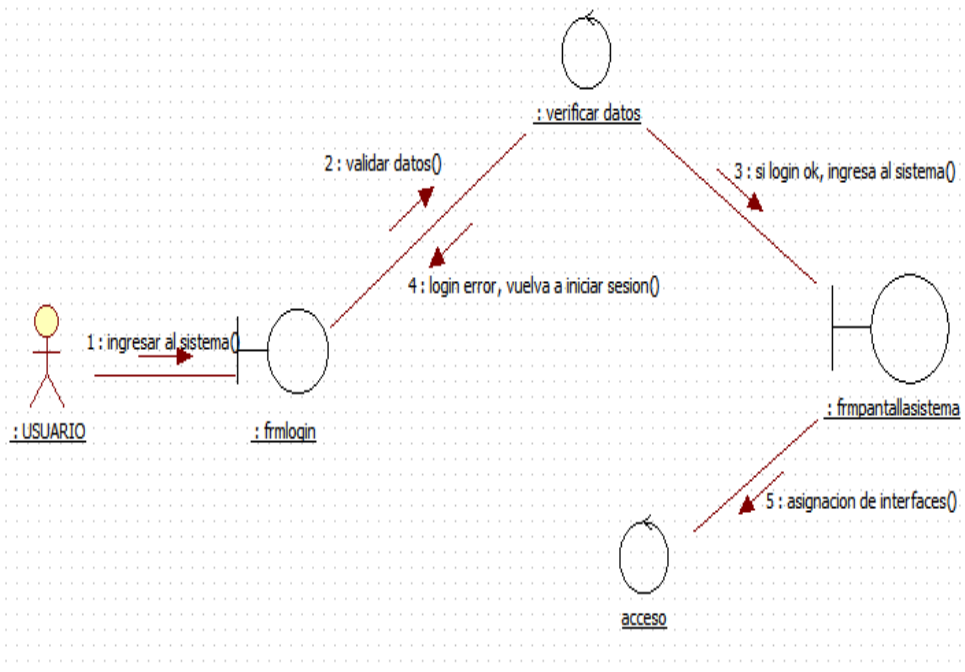




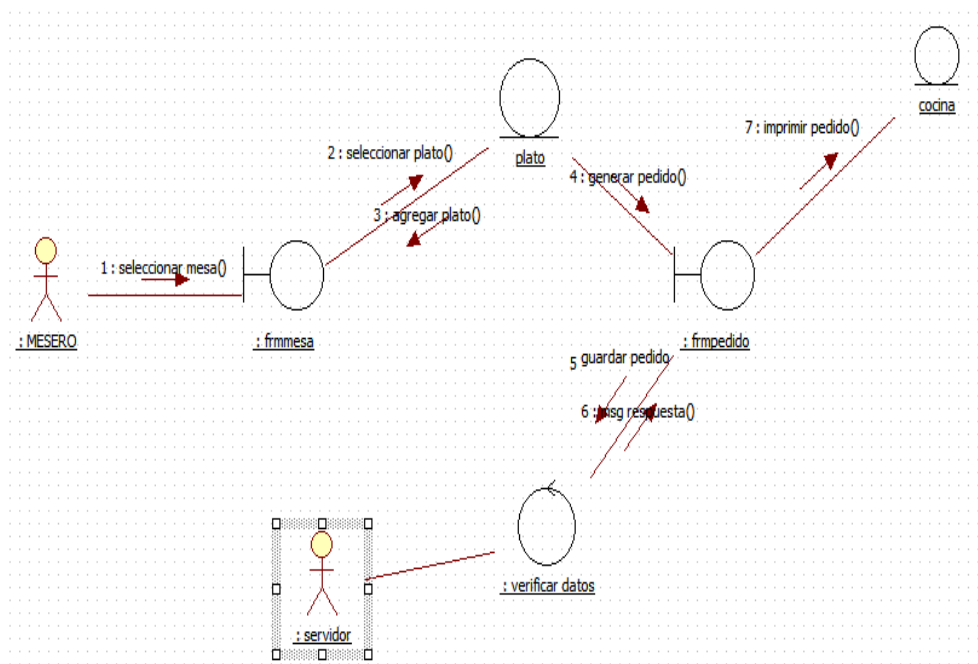
## SUBSISTEMA DE EGRESOS Y CAJA



## SUBSISTEMA DE USUARIO

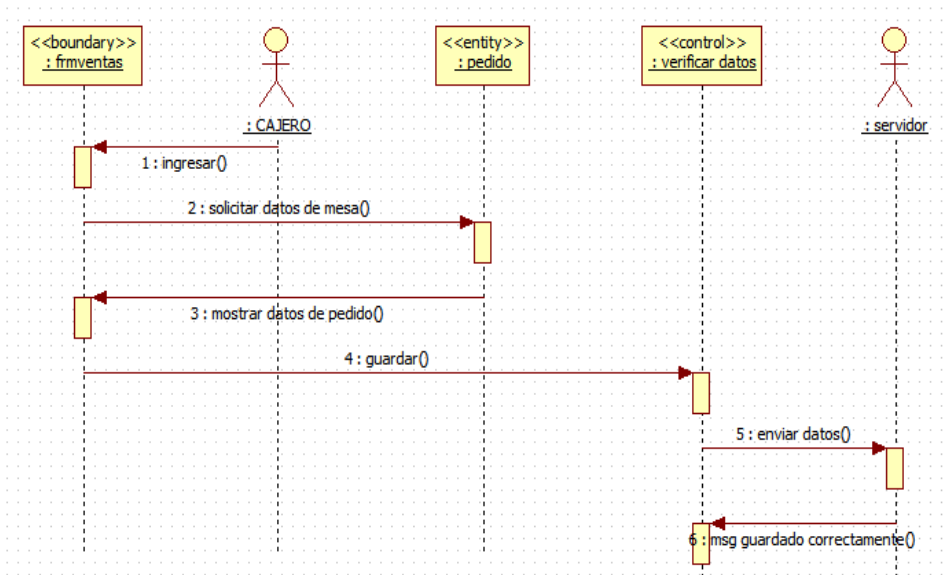


## SUBSISTEMA DE PEDIDO

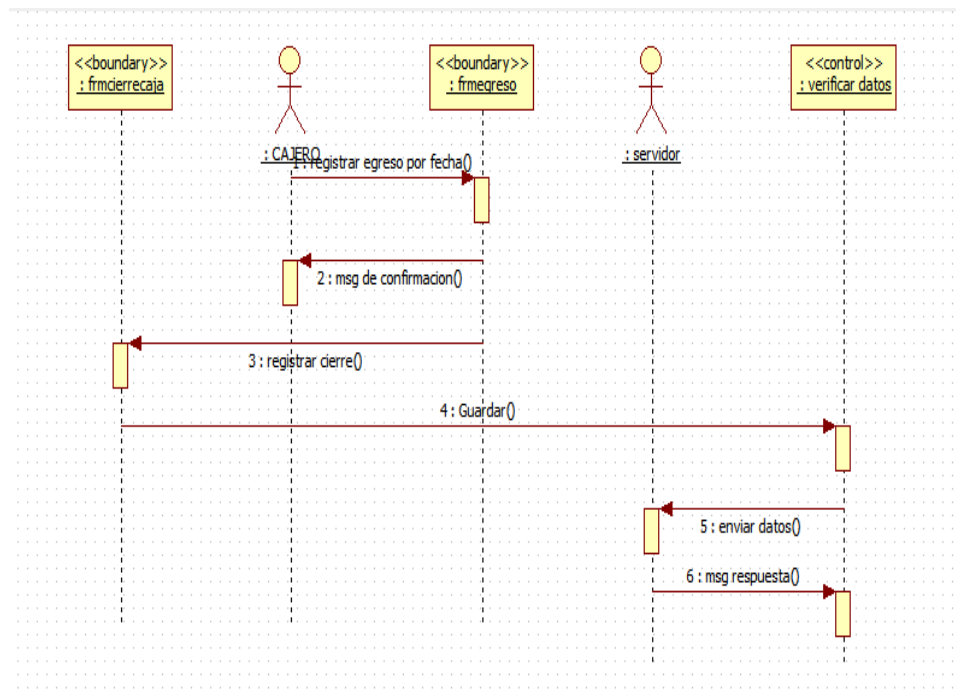


## 4.5. DIAGRAMA DE SECUENCIA POR MÓDULOS

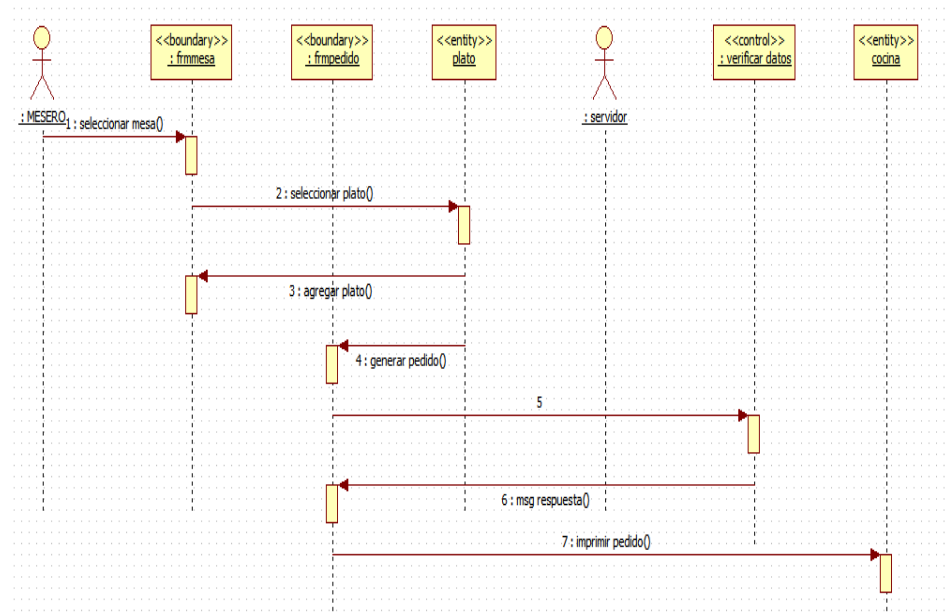
### SUBSISTEMA DE FACTURACION



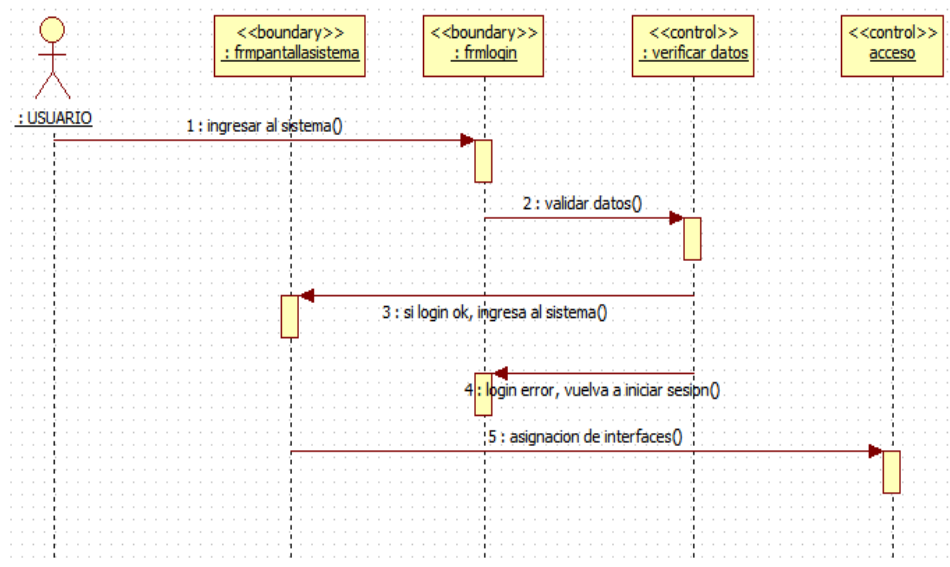
## SUBSISTEMA DE EGRESOS Y CAJA



## SUBSISTEMA DE PEDIDO



## SUBSISTEMA DE USUARIO



## 4.6. CODIFICACIÓN

### IMPLEMENTACIÓN / GENERAR CÓDIGO

El siguiente código se crea las tablas necesarias para el funcionamiento de la base de datos

#### Facturación

```
DROP TABLE IF EXISTS `facturacion`;
CREATE TABLE `facturacion` (
  `cod_facturacion` int(11) NOT NULL AUTO_INCREMENT,
  `fact_fechaemision` datetime DEFAULT NULL,
  `fact_pago` enum('Contado','Credito') DEFAULT 'Credito',
  `fact_comprel` varchar(150) DEFAULT NULL,
  `fact_monto` float DEFAULT NULL COMMENT 'venta->monto de venta',
  `fact_descuento` float DEFAULT NULL,
  `fact_subtotal` float DEFAULT NULL,
  `fact_igv` float DEFAULT NULL,
  `fact_total` float DEFAULT NULL,
  `fact_vuelto` float DEFAULT NULL,
  `fact_efectivo` float DEFAULT NULL,
  `fact_serie` varchar(25) DEFAULT NULL COMMENT 'venta: serie -numero, compra: serie',
  `fact_numero` varchar(12) DEFAULT NULL,
  `fact_estado` enum('SI','NO') DEFAULT NULL COMMENT 'si: anulado, no: no anulado',
  `cod_usuario` int(11) NOT NULL,
  `cod_cliente` int(11) DEFAULT NULL,
  `cod_comprobante` int(11) DEFAULT NULL,
  `cod_almacen` int(11) NOT NULL,
  `cod_impuestos` int(11) DEFAULT NULL,
  `fact_tarjeta` varchar(150) DEFAULT NULL,
  `fact_numoperacion` varchar(10) DEFAULT NULL,
  `fact_afectastock` enum('SI','NO') NOT NULL,
  `fact_observacion` varchar(500) DEFAULT NULL,
  `fact_alias` varchar(150) DEFAULT NULL,
  `fact_fechaanulacion` datetime DEFAULT NULL,
  `fact_cajero` varchar(200) DEFAULT NULL,
  `fact_motivo`
enum('ANULACION','DESCUENTO','BONIFICACION','DEVOLUCION','COTIZACION','VENTA')
DEFAULT NULL,
  `cod_motivooperacion` int(11) DEFAULT NULL,
  `fact_comisiontarjeta` double DEFAULT NULL,
  `fact_tipotarjeta` enum('DEBITO','NINGUNO','CREDITO') DEFAULT NULL,
  `cod_origen` int(4) DEFAULT NULL,
  `cod_mesa` int(11) DEFAULT NULL,
  PRIMARY KEY (`cod_facturacion`),
  KEY `fk_facturacion_comprobante1` (`cod_comprobante`),
  KEY `fk_facturacion_producto_almacen1` (`cod_almacen`),
  KEY `fk_facturacion_usuario1` (`cod_usuario`),
  KEY `fk_facturacion_cliente1` (`cod_cliente`),
  KEY `fk_facturacion_impuestos1` (`cod_impuestos`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;
```

#### Mesa

```
DROP TABLE IF EXISTS `mesa`;
CREATE TABLE `mesa` (
  `cod_mesa` int(11) NOT NULL AUTO_INCREMENT,
  `mesa_descripcion` varchar(100) DEFAULT NULL,
  `mesa_activo` enum('SI','NO') DEFAULT NULL,
  PRIMARY KEY (`cod_mesa`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;
```

#### 1.3. Producto

```

DROP TABLE IF EXISTS `producto`;
CREATE TABLE `producto` (
  `cod_producto` int(11) NOT NULL AUTO_INCREMENT,
  `prod_serinumber` varchar(20) DEFAULT NULL COMMENT 'PARA CODIGO DE BARRAS',
  `prod_descripcion` varchar(120) DEFAULT NULL,
  `prod_nombrecorto` varchar(150) DEFAULT "",
  `prod_observacion` text,
  `prod_stockmin` int(11) DEFAULT NULL COMMENT 'referencial:para advertir que se está
quedando con menos stockque requiere actualmente',
  `prod_costo` double NOT NULL,
  `prod_precio` double NOT NULL,
  `prod_preciocontado` double DEFAULT NULL,
  `prod_descuento` float DEFAULT NULL,
  `prod_porganancia` float DEFAULT NULL COMMENT 'porcentaje de ganancia',
  `prod_tipocambio` float DEFAULT NULL,
  `prod_imagen` varchar(400) DEFAULT NULL,
  `prod_fecharegistro` datetime NOT NULL,
  `cod_categoria` int(11) NOT NULL,
  `prod_oferta` enum('SI','NO') NOT NULL,
  `prod_xoferta` double DEFAULT NULL,
  `prod_preciooferta` double DEFAULT NULL,
  `prod_cantpromocion` int(11) DEFAULT NULL,
  `prod_activo` enum('SI','NO') DEFAULT NULL,
  `cod_cocina` int(11) DEFAULT NULL,
  `prod_tipo` enum('PLATO','PRODUCTO') DEFAULT NULL,
  PRIMARY KEY (`cod_producto`),
  UNIQUE KEY `prod_descripcion` (`prod_descripcion`)
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

```

#### 1.4. Almacen

```

DROP TABLE IF EXISTS `almacen`;
CREATE TABLE `almacen` (
  `cod_almacen` int(11) NOT NULL AUTO_INCREMENT,
  `alma_descripcion` varchar(50) DEFAULT NULL,
  `alma_abreviatura` varchar(10) DEFAULT NULL,
  `alma_direccion` varchar(50) DEFAULT NULL,
  `alma_eliminado` enum('NO','SI') NOT NULL,
  PRIMARY KEY (`cod_almacen`),
  UNIQUE KEY `almacen` (`alma_descripcion`)
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

```

#### 1.5. Cocina

```

DROP TABLE IF EXISTS `cocina`;
CREATE TABLE `cocina` (
  `cod_cocina` int(11) NOT NULL AUTO_INCREMENT,
  `coci_descripcion` varchar(250) DEFAULT NULL,
  `coci_abreviatura` varchar(50) DEFAULT NULL,
  `coci_activo` enum('SI','NO') DEFAULT NULL,
  PRIMARY KEY (`cod_cocina`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

```

#### 1.6. Comprobante

```

DROP TABLE IF EXISTS `comprobante`;
CREATE TABLE `comprobante` (
  `cod_comprobante` int(11) NOT NULL AUTO_INCREMENT,
  `comp_descripcion` varchar(45) DEFAULT NULL COMMENT 'Actualizacion\nCuando se ingresa
la mercadería por primera vez, para el aumento de stock ',
  `comp_observacion` varchar(45) DEFAULT NULL,
  `comp_activo` enum('NO','SI') NOT NULL,
  PRIMARY KEY (`cod_comprobante`)
) ENGINE=MyISAM AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;

```

## CODIGO DEL SOFTWARE DE RESTAURANTE FALCÓN

-----  
-- Table structure for acceso

```

-----
DROP TABLE IF EXISTS `acceso`;
CREATE TABLE `acceso` (
  `cod_acceso` int(11) NOT NULL AUTO_INCREMENT,
  `acce_acceso` varchar(250) DEFAULT NULL,
  `acce_visible` enum('TRUE','FALSE') DEFAULT 'TRUE',
  `cod_perfil` int(11) DEFAULT NULL,
  `cod_formulario` int(11) DEFAULT NULL,
  `acce_eliminado` enum('SI','NO') DEFAULT 'NO',
  PRIMARY KEY (`cod_acceso`)
) ENGINE=InnoDB AUTO_INCREMENT=76 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for almacén
-----

```

```

DROP TABLE IF EXISTS `almacen`;
CREATE TABLE `almacen` (
  `cod_almacen` int(11) NOT NULL AUTO_INCREMENT,
  `alma_descripcion` varchar(50) DEFAULT NULL,
  `alma_abreviatura` varchar(10) DEFAULT NULL,
  `alma_direccion` varchar(50) DEFAULT NULL,
  `alma_eliminado` enum('NO','SI') NOT NULL,
  PRIMARY KEY (`cod_almacen`),
  UNIQUE KEY `almacen` (`alma_descripcion`)
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for caja
-----

```

```

DROP TABLE IF EXISTS `caja`;
CREATE TABLE `caja` (
  `cod_caja` int(11) NOT NULL AUTO_INCREMENT,
  `caja_fecha` datetime DEFAULT NULL,
  `caja_operacion` enum('INGRESO','EGRESO') DEFAULT NULL,
  `cod_tipomovimiento` int(11) DEFAULT NULL,
  `caja_monto` double(11,4) DEFAULT NULL,
  `cod_comprobante` int(11) DEFAULT NULL,
  `caja_serienumero` varchar(20) DEFAULT NULL,
  `caja_fecharegistro` datetime DEFAULT NULL,
  `caja_detalle` varchar(250) DEFAULT NULL,
  `caja_observacion` varchar(250) DEFAULT NULL,
  `caja_persona` varchar(250) DEFAULT NULL,
  `cod_facturacion` int(255) DEFAULT NULL,
  `cod_egreso` int(11) DEFAULT NULL,
  `cod_cliente` int(255) DEFAULT NULL,
  `cod_proveedor` int(11) DEFAULT NULL,
  `cod_usuario` int(11) DEFAULT NULL,
  `caja_afecta` enum('SI','NO') DEFAULT NULL,
  `caja_proviene` enum('VENTA','EGRESO','CIERRE','ANULACION') DEFAULT NULL,
  PRIMARY KEY (`cod_caja`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for categoría
-----

```

```

DROP TABLE IF EXISTS `categoria`;
CREATE TABLE `categoria` (
  `cod_categoria` int(11) NOT NULL AUTO_INCREMENT,
  `cate_descripcion` varchar(250) DEFAULT NULL,
  `cate_abreviatura` varchar(250) DEFAULT NULL,
  `cate_activo` enum('NO','SI') NOT NULL,
  PRIMARY KEY (`cod_categoria`),
  UNIQUE KEY `categoria` (`cate_descripcion`)
) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for cierrecaja
-----

```

```

DROP TABLE IF EXISTS `cierrecaja`;
CREATE TABLE `cierrecaja` (
  `cod_cierrecaja` int(11) NOT NULL AUTO_INCREMENT,
  `cica_fecha` datetime DEFAULT NULL,
  `cica_monto` double(11,4) DEFAULT NULL,
  `cica_cajachica` double(11,4) DEFAULT NULL,
  `cica_observacion` varchar(250) DEFAULT NULL,
  `cod_usuario` int(11) DEFAULT NULL,
  `cica_encargado` varchar(150) DEFAULT NULL,
  `cica_montoanterior` double(11,4) DEFAULT NULL,
  `cica_tarjeta` double(11,4) DEFAULT NULL,
  `cica_total` double(11,4) DEFAULT NULL,
  PRIMARY KEY (`cod_cierrecaja`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

```

-- Table structure for cliente

```
-----  
DROP TABLE IF EXISTS `cliente`;  
CREATE TABLE `cliente` (  
  `cod_cliente` int(11) NOT NULL AUTO_INCREMENT,  
  `clie_nombre` varchar(150) DEFAULT NULL COMMENT 'igual a razon social',  
  `clie_direccion` varchar(150) DEFAULT NULL,  
  `clie_ruc` varchar(12) DEFAULT NULL COMMENT 'dni\r\nruc',  
  `cod_clientetipo` int(11) NOT NULL COMMENT 'cliente: cliente sin empresanegocio: empresa de negocio',  
  `clie_distrito` varchar(200) DEFAULT NULL,  
  `clie_provincia` varchar(200) DEFAULT NULL,  
  `clie_departamento` varchar(200) DEFAULT NULL,  
  `clie_nacimiento` date DEFAULT NULL,  
  `clie_lineacredito` double DEFAULT NULL,  
  `clie_disponible` double DEFAULT NULL,  
  `clie_otorgado` double DEFAULT NULL,  
  `clie_movistar` varchar(100) DEFAULT NULL,  
  `clie_whatsapp` varchar(30) DEFAULT NULL,  
  `clie_fijo` varchar(15) DEFAULT NULL,  
  `clie_email` varchar(100) DEFAULT NULL,  
  `clie_activo` enum('SI','NO') DEFAULT NULL,  
  PRIMARY KEY (`cod_cliente`),  
  UNIQUE KEY `cliente` (`clie_ruc`) USING BTREE  
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;
```

-- Table structure for clientetipo

```
-----  
DROP TABLE IF EXISTS `clientetipo`;  
CREATE TABLE `clientetipo` (  
  `cod_clientetipo` int(11) NOT NULL AUTO_INCREMENT,  
  `clti_nombre` varchar(45) NOT NULL,  
  `clti_descripcion` varchar(100) DEFAULT NULL,  
  `clti_activo` enum('NO','SI') NOT NULL DEFAULT 'NO',  
  PRIMARY KEY (`cod_clientetipo`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

-- Table structure for cocina

```
-----  
DROP TABLE IF EXISTS `cocina`;  
CREATE TABLE `cocina` (  
  `cod_cocina` int(11) NOT NULL AUTO_INCREMENT,  
  `coci_descripcion` varchar(250) DEFAULT NULL,  
  `coci_abreviatura` varchar(50) DEFAULT NULL,  
  `coci_activo` enum('SI','NO') DEFAULT NULL,  
  PRIMARY KEY (`cod_cocina`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

-- Table structure for comisionarjeta

```
-----  
DROP TABLE IF EXISTS `comisionarjeta`;  
CREATE TABLE `comisionarjeta` (  
  `cod_tarjetacomision` int(11) NOT NULL AUTO_INCREMENT,  
  `taco_tipo` enum('DEBITO','CREDITO') DEFAULT NULL,  
  `taco_porcentaje` double DEFAULT NULL,  
  `cod_tarjeta` int(11) DEFAULT NULL,  
  `taco_activo` enum('SI','NO') DEFAULT 'NO',  
  PRIMARY KEY (`cod_tarjetacomision`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
```

-- Table structure for compra

```
-----  
DROP TABLE IF EXISTS `compra`;  
CREATE TABLE `compra` (  
  `cod_compra` int(11) NOT NULL AUTO_INCREMENT,  
  `comp_fechaemision` datetime DEFAULT NULL,  
  `comp_pago` enum('Credito','Letras','Contado') NOT NULL,  
  `comp_comprel` varchar(150) DEFAULT NULL,  
  `comp_oc` varchar(15) DEFAULT NULL,  
  `comp_monto` float DEFAULT NULL COMMENT 'venta->monto de venta',  
  `comp_descuento` float DEFAULT NULL,  
  `comp_subtotal` float DEFAULT NULL,  
  `comp_igv` float DEFAULT NULL,  
  `comp_total` float DEFAULT NULL,  
  `comp_operacion` varchar(100) DEFAULT NULL,  
  `comp_serie` varchar(25) DEFAULT NULL COMMENT 'venta: serie -numero, compra: serie',  
  `comp_numero` varchar(12) DEFAULT NULL,  
  `comp_anulado` enum('SI','NO') DEFAULT NULL COMMENT 'si: anulado, no: no anulado',  
  `cod_transporte` int(11) NOT NULL,  
  `cod_usuario` int(11) NOT NULL,  
  `cod_comprobante` int(11) DEFAULT NULL,  
  `cod_almacen` int(11) NOT NULL,  
  `comp_recepcionado` enum('SI','NO') DEFAULT NULL COMMENT 'si:recepcionado, no: no recepcionado',
```



```

`cod_proveedor` int(11) NOT NULL,
`cod_impuestos` int(11) NOT NULL,
`comp_afectastock` enum('SI','NO') NOT NULL,
`cod_moneda` int(11) DEFAULT '1',
`comp_tipocambio` double DEFAULT '1',
`comp_observacion` varchar(500) DEFAULT NULL,
`comp_imagen` varchar(100) DEFAULT NULL,
`comp_motivo` enum('COMPRA','ANULACION','DESCUENTO','BONIFICACION','DEVOLUCION','OTROS','INGRESO','AJUSTE
DE STOCK','DEVOLUCION') DEFAULT NULL,
`cod_motivooperacion` int(11) DEFAULT NULL,
`comp_percepcion` float DEFAULT NULL COMMENT 'impuesto adelanto no influye en el costo',
`comp_totalcompra` float DEFAULT NULL,
PRIMARY KEY (`cod_compra`),
KEY `fk_facturacion_comprobante1` (`cod_comprobante`),
KEY `fk_compra_transportista1` (`cod_transporte`),
KEY `fk_compra_impuestos1` (`cod_impuestos`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for comprobante
-----

```

```

DROP TABLE IF EXISTS `comprobante`;
CREATE TABLE `comprobante` (
  `cod_comprobante` int(11) NOT NULL AUTO_INCREMENT,
  `comp_descripcion` varchar(45) DEFAULT NULL COMMENT 'Actualizacion\nCuando se ingresa la mercaderia por primera vez,
para el aumento de stock ',
  `comp_observacion` varchar(45) DEFAULT NULL,
  `comp_activo` enum('NO','SI') NOT NULL,
  PRIMARY KEY (`cod_comprobante`)
) ENGINE=MyISAM AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for conexionlocal
-----

```

```

DROP TABLE IF EXISTS `conexionlocal`;
CREATE TABLE `conexionlocal` (
  `cod_localip` int(11) NOT NULL AUTO_INCREMENT,
  `colo_ipremota` varchar(29) DEFAULT NULL,
  `colo_usuariobd` varchar(20) DEFAULT NULL,
  `colo_clavebd` varchar(20) DEFAULT NULL,
  `colo_basedata` varchar(100) DEFAULT NULL,
  `colo_posicion` enum('ALMACEN','EMPRESA') DEFAULT NULL,
  `colo_almacen` varchar(50) DEFAULT NULL,
  `colo_direccion` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`cod_localip`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for configuracion
-----

```

```

DROP TABLE IF EXISTS `configuracion`;
CREATE TABLE `configuracion` (
  `cod_configuracion` int(11) NOT NULL AUTO_INCREMENT,
  `conf_tcventa` double DEFAULT NULL,
  `conf_tccompra` double DEFAULT NULL,
  `conf_igv` double DEFAULT NULL,
  `conf_rutaimagen` varchar(40) DEFAULT NULL,
  `conf_logo` varchar(20) DEFAULT NULL,
  `conf_encabezado` varchar(100) DEFAULT NULL,
  `conf_piepagina` varchar(100) DEFAULT NULL,
  `conf_apertura` double DEFAULT NULL,
  `conf_impuesto` double DEFAULT NULL,
  `conf_impresoraseleccionada` varchar(100) DEFAULT NULL,
  `conf_nombreempresa` varchar(300) DEFAULT NULL,
  `conf_ruc` varchar(15) DEFAULT NULL,
  `conf_direccionempresa` varchar(300) DEFAULT NULL,
  `conf_nombrecomercial` varchar(300) DEFAULT NULL,
  `conf_ubigeo` varchar(255) DEFAULT NULL,
  `conf_percepcion` varchar(10) DEFAULT NULL,
  `conf_direccionempresaprincipal` varchar(250) DEFAULT NULL,
  `conf_facebook` varchar(150) DEFAULT NULL,
  PRIMARY KEY (`cod_configuracion`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for configuracionimpresora
-----

```

```

DROP TABLE IF EXISTS `configuracionimpresora`;
CREATE TABLE `configuracionimpresora` (
  `cod_confimpresora` int(11) NOT NULL AUTO_INCREMENT,
  `coim_serie` varchar(25) DEFAULT NULL COMMENT 'serie impresora',
  `coim_modelo` varchar(10) DEFAULT NULL,
  `coim_marca` varchar(10) DEFAULT NULL,
  `coim_pos` varchar(5) DEFAULT NULL,
  `cod_impresora` int(11) DEFAULT NULL,

```

```

`coim_tipo` varchar(100) DEFAULT NULL,
`cod_serie` int(11) DEFAULT NULL,
`coim_autorizacionsunat` varchar(25) DEFAULT NULL,
PRIMARY KEY (`cod_confimpresora`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for cronogramasunat
-----

```

```

DROP TABLE IF EXISTS `cronogramasunat`;
CREATE TABLE `cronogramasunat` (
  `cod_cronogramasunat` int(11) NOT NULL AUTO_INCREMENT,
  `crgr_periodo` varchar(20) DEFAULT NULL,
  `crgr_periodoanio` year(4) DEFAULT NULL,
  `crgr_0` int(11) DEFAULT NULL,
  `crgr_1` int(11) DEFAULT NULL,
  `crgr_2` int(11) DEFAULT NULL,
  `crgr_3` int(11) DEFAULT NULL,
  `crgr_4` int(11) DEFAULT NULL,
  `crgr_5` int(11) DEFAULT NULL,
  `crgr_6` int(11) DEFAULT NULL,
  `crgr_7` int(11) DEFAULT NULL,
  `crgr_8` int(11) DEFAULT NULL,
  `crgr_9` int(11) DEFAULT NULL,
  `crgr_mes` varchar(15) DEFAULT NULL,
  `crgr_anio` year(4) DEFAULT NULL,
  `cod_mes` int(11) DEFAULT NULL,
  PRIMARY KEY (`cod_cronogramasunat`)
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for documento
-----

```

```

DROP TABLE IF EXISTS `documento`;
CREATE TABLE `documento` (
  `cod_documento` int(11) NOT NULL AUTO_INCREMENT,
  `docu_numero` varchar(8) DEFAULT NULL,
  `docu_afecto` enum('SI','NO') DEFAULT NULL,
  `cod_serie` int(11) NOT NULL,
  `cod_almacen` int(11) NOT NULL,
  `cod_comprobante` int(11) NOT NULL,
  `docu_estado` enum('SI','NO') DEFAULT NULL COMMENT 'ACTIVO?',
  PRIMARY KEY (`cod_documento`),
  KEY `fk_documento_serie1` (`cod_serie`),
  KEY `fk_documento_almacen1` (`cod_almacen`),
  KEY `fk_documento_comprobante1` (`cod_comprobante`)
) ENGINE=MyISAM AUTO_INCREMENT=31 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for facturacion
-----

```

```

DROP TABLE IF EXISTS `facturacion`;
CREATE TABLE `facturacion` (
  `cod_facturacion` int(11) NOT NULL AUTO_INCREMENT,
  `fact_fechaemision` datetime DEFAULT NULL,
  `fact_pago` enum('Contado','Credito') DEFAULT 'Credito',
  `fact_comprel` varchar(150) DEFAULT NULL,
  `fact_monto` float DEFAULT NULL COMMENT 'venta->monto de venta',
  `fact_descuento` float DEFAULT NULL,
  `fact_subtotal` float DEFAULT NULL,
  `fact_igv` float DEFAULT NULL,
  `fact_total` float DEFAULT NULL,
  `fact_vuelto` float DEFAULT NULL,
  `fact_efectivo` float DEFAULT NULL,
  `fact_serie` varchar(25) DEFAULT NULL COMMENT 'venta: serie -numero, compra: serie',
  `fact_numero` varchar(12) DEFAULT NULL,
  `fact_estado` enum('SI','NO') DEFAULT NULL COMMENT 'si: anulado, no: no anulado',
  `cod_usuario` int(11) NOT NULL,
  `cod_cliente` int(11) DEFAULT NULL,
  `cod_comprobante` int(11) DEFAULT NULL,
  `cod_almacen` int(11) NOT NULL,
  `cod_impuestos` int(11) DEFAULT NULL,
  `fact_tarjeta` varchar(150) DEFAULT NULL,
  `fact_numoperacion` varchar(10) DEFAULT NULL,
  `fact_afectastock` enum('SI','NO') NOT NULL,
  `fact_observacion` varchar(500) DEFAULT NULL,
  `fact_alias` varchar(150) DEFAULT NULL,
  `fact_fechaanulacion` datetime DEFAULT NULL,
  `fact_cajero` varchar(200) DEFAULT NULL,
  `fact_motivo` enum('ANULACION','DESCUENTO','BONIFICACION','DEVOLUCION','COTIZACION','VENTA') DEFAULT NULL,
  `cod_motivooperacion` int(11) DEFAULT NULL,
  `fact_comisiontarjeta` double DEFAULT NULL,
  `fact_tipotarjeta` enum('DEBITO','NINGUNO','CREDITO') DEFAULT NULL,
  `cod_origen` int(4) DEFAULT NULL,
  `cod_mesa` int(11) DEFAULT NULL,
  PRIMARY KEY (`cod_facturacion`),

```

```

KEY `fk_facturacion_comprobante1` (`cod_comprobante`),
KEY `fk_facturacion_producto_almacen1` (`cod_almacen`),
KEY `fk_facturacion_usuario1` (`cod_usuario`),
KEY `fk_facturacion_cliente1` (`cod_cliente`),
KEY `fk_facturacion_impuestos1` (`cod_impuestos`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for impresora
-----

```

```

DROP TABLE IF EXISTS `impresora`;
CREATE TABLE `impresora` (
  `cod_impresora` int(11) NOT NULL AUTO_INCREMENT,
  `impr_impresora` enum('MATRICIAL','TICKETERA') DEFAULT NULL,
  `impr_activo` enum('SI','NO') DEFAULT NULL,
  PRIMARY KEY (`cod_impresora`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for impresorafisica
-----

```

```

DROP TABLE IF EXISTS `impresorafisica`;
CREATE TABLE `impresorafisica` (
  `cod_impresorafisica` int(11) DEFAULT NULL,
  `imfi_impresora` varchar(255) DEFAULT NULL,
  `cod_impresora` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for impuestos
-----

```

```

DROP TABLE IF EXISTS `impuestos`;
CREATE TABLE `impuestos` (
  `cod_impuestos` int(11) NOT NULL AUTO_INCREMENT,
  `impu_descripcion` varchar(250) DEFAULT NULL,
  `impu_observacion` varchar(100) DEFAULT NULL,
  `impu_afecto` enum('SI','NO') DEFAULT NULL,
  `impu_valor` double DEFAULT NULL COMMENT 'valor del igv en porcentaje',
  `impu_formula` enum('MULTIPLICACION','DIVISION') DEFAULT NULL COMMENT 'formula del igv',
  `impu_modocalculo` enum('MULTIPLICACION','DIVISION') NOT NULL DEFAULT 'MULTIPLICACION',
  `impu_eliminado` enum('NO','SI') NOT NULL,
  PRIMARY KEY (`cod_impuestos`)
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for kardex
-----

```

```

DROP TABLE IF EXISTS `kardex`;
CREATE TABLE `kardex` (
  `cod_kardex` int(11) NOT NULL AUTO_INCREMENT,
  `cod_facturacion` int(11) DEFAULT NULL,
  `cod_compra` int(11) DEFAULT NULL,
  `cod_almacen` int(11) NOT NULL,
  `kard_fecha` datetime DEFAULT NULL,
  `cod_producto` int(11) DEFAULT NULL,
  `kard_movimiento` double DEFAULT NULL,
  `cod_comprobante` int(11) DEFAULT NULL,
  `kard_operacion` enum('INGRESO','EGRESO') DEFAULT NULL,
  `cod_motivooperacion` int(11) NOT NULL,
  `cod_traspaso` int(11) NOT NULL,
  PRIMARY KEY (`cod_kardex`),
  KEY `fk_kardex_facturacion1` (`cod_facturacion`),
  KEY `fk_kardex_compra1` (`cod_compra`),
  KEY `fk_kardex_producto_almacen1` (`cod_almacen`)
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for mesa
-----

```

```

DROP TABLE IF EXISTS `mesa`;
CREATE TABLE `mesa` (
  `cod_mesa` int(11) NOT NULL AUTO_INCREMENT,
  `mesa_descripcion` varchar(100) DEFAULT NULL,
  `mesa_activo` enum('SI','NO') DEFAULT NULL,
  PRIMARY KEY (`cod_mesa`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for moneda
-----

```

```

DROP TABLE IF EXISTS `moneda`;
CREATE TABLE `moneda` (

```

```

`cod_moneda` int(11) NOT NULL AUTO_INCREMENT,
`mone_descripcion` varchar(150) DEFAULT NULL,
`mone_simbolo` varchar(10) DEFAULT NULL,
`mone_activo` enum('SI','NO') DEFAULT NULL,
PRIMARY KEY (`cod_moneda`)
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for motivo_operacion
-----

```

```

DROP TABLE IF EXISTS `motivo_operacion`;
CREATE TABLE `motivo_operacion` (
  `cod_motivooperacion` int(11) NOT NULL AUTO_INCREMENT,
  `mope_nombre` varchar(30) NOT NULL,
  `mope_observacion` varchar(40) DEFAULT NULL,
  `mope_formulario` enum('VENTA','COMPRA','ANULACION','TRASPASO','NOTACREDITO') NOT NULL COMMENT 'se usa para definir a que formulario va.',
  `mope_eliminado` enum('NO','SI') NOT NULL,
  PRIMARY KEY (`cod_motivooperacion`)
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for ofertaproducto
-----

```

```

DROP TABLE IF EXISTS `ofertaproducto`;
CREATE TABLE `ofertaproducto` (
  `cod_oferta` int(11) NOT NULL AUTO_INCREMENT,
  `ofer_precio` double DEFAULT NULL,
  `ofer_descuento` double DEFAULT NULL,
  `ofer_observacion` varchar(300) DEFAULT NULL,
  `ofer_fechainicio` date DEFAULT NULL,
  `ofer_activado` enum('SI','NO') DEFAULT NULL,
  `cod_producto` int(11) DEFAULT NULL,
  `ofer_fechavencimiento` date DEFAULT NULL,
  PRIMARY KEY (`cod_oferta`),
  UNIQUE KEY `oferta` (`ofer_activado`,`cod_producto`,`ofer_fechainicio`,`ofer_fechavencimiento`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for perfilusuario
-----

```

```

DROP TABLE IF EXISTS `perfilusuario`;
CREATE TABLE `perfilusuario` (
  `cod_perfil` int(11) NOT NULL AUTO_INCREMENT,
  `perf_descripcion` varchar(100) DEFAULT NULL,
  `perf_activo` enum('SI','NO') DEFAULT NULL,
  PRIMARY KEY (`cod_perfil`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for producto
-----

```

```

DROP TABLE IF EXISTS `producto`;
CREATE TABLE `producto` (
  `cod_producto` int(11) NOT NULL AUTO_INCREMENT,
  `prod_serinumber` varchar(20) DEFAULT NULL COMMENT 'PARA CODIGO DE BARRAS',
  `prod_descripcion` varchar(120) DEFAULT NULL,
  `prod_nombrecorto` varchar(150) DEFAULT "",
  `prod_observacion` text,
  `prod_stockmin` int(11) DEFAULT NULL COMMENT 'referencial:para advertir que se está quedando con menos stockque requiere actualmente',
  `prod_costo` double NOT NULL,
  `prod_precio` double NOT NULL,
  `prod_preciocontado` double DEFAULT NULL,
  `prod_descuento` float DEFAULT NULL,
  `prod_porganancia` float DEFAULT NULL COMMENT 'porcentaje de ganancia',
  `prod_tipocambio` float DEFAULT NULL,
  `prod_imagen` varchar(400) DEFAULT NULL,
  `prod_fecharegistro` datetime NOT NULL,
  `cod_categoria` int(11) NOT NULL,
  `prod_oferta` enum('SI','NO') NOT NULL,
  `prod_xoferta` double DEFAULT NULL,
  `prod_preciooferta` double DEFAULT NULL,
  `prod_cantpromocion` int(11) DEFAULT NULL,
  `prod_activo` enum('SI','NO') DEFAULT NULL,
  `cod_cocina` int(11) DEFAULT NULL,
  `prod_tipo` enum('PLATO','PRODUCTO') DEFAULT NULL,
  PRIMARY KEY (`cod_producto`),
  UNIQUE KEY `prod_descripcion` (`prod_descripcion`)
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for producto_almacen
-----
DROP TABLE IF EXISTS `producto_almacen`;
CREATE TABLE `producto_almacen` (
  `cod_proalm` int(11) NOT NULL AUTO_INCREMENT,
  `proalm_stockinicial` int(11) DEFAULT NULL,
  `proalm_stock` int(11) DEFAULT NULL,
  `cod_producto` int(11) NOT NULL,
  `cod_almacen` int(11) NOT NULL,
  PRIMARY KEY (`cod_proalm`),
  KEY `fk_producto_almacen_producto1` (`cod_producto`),
  KEY `fk_producto_almacen_almacen1` (`cod_almacen`)
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for proveedor
-----
DROP TABLE IF EXISTS `proveedor`;
CREATE TABLE `proveedor` (
  `cod_proveedor` int(11) NOT NULL AUTO_INCREMENT,
  `prov_ruc` varchar(12) DEFAULT NULL,
  `prov_razonsocial` varchar(45) DEFAULT NULL,
  `prov_direccion` varchar(200) DEFAULT NULL,
  `prov_referencia` varchar(200) DEFAULT NULL,
  `prov_gironegocio` varchar(100) DEFAULT NULL,
  `prov_departamento` varchar(200) DEFAULT NULL,
  `prov_provincia` varchar(45) DEFAULT NULL,
  `prov_distrito` varchar(45) DEFAULT NULL,
  `prov_celular` varchar(14) DEFAULT NULL,
  `prov_fijo` varchar(15) DEFAULT NULL,
  `prov_email` varchar(45) DEFAULT NULL,
  `prov_nrocuenta` varchar(500) DEFAULT NULL,
  `prov_activo` enum('SI','NO') DEFAULT NULL,
  `cod_tipoproveedor` int(11) DEFAULT NULL,
  `prov_whatsapp` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`cod_proveedor`),
  UNIQUE KEY `proveedor` (`prov_ruc`) USING BTREE
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for serie
-----
DROP TABLE IF EXISTS `serie`;
CREATE TABLE `serie` (
  `cod_serie` int(11) NOT NULL AUTO_INCREMENT,
  `seri_descripcion` varchar(250) DEFAULT NULL,
  `seri_serie` varchar(40) DEFAULT NULL,
  `seri_activo` enum('SI','NO') DEFAULT NULL,
  PRIMARY KEY (`cod_serie`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for tarjeta
-----
DROP TABLE IF EXISTS `tarjeta`;
CREATE TABLE `tarjeta` (
  `cod_tarjeta` int(11) NOT NULL AUTO_INCREMENT,
  `tarj_tarjeta` varchar(150) DEFAULT NULL,
  `tarj_observacion` varchar(150) DEFAULT NULL,
  `tarj_activo` enum('SI','NO') DEFAULT NULL,
  PRIMARY KEY (`cod_tarjeta`),
  UNIQUE KEY `tarj_tarjeta` (`tarj_tarjeta`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for tipomovimiento
-----
DROP TABLE IF EXISTS `tipomovimiento`;
CREATE TABLE `tipomovimiento` (
  `cod_tipomovimiento` int(255) NOT NULL AUTO_INCREMENT,
  `timo_descripcion` varchar(200) NOT NULL,
  `timo_activo` enum('SI','NO') DEFAULT NULL,
  PRIMARY KEY (`cod_tipomovimiento`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for tipoproveedor
-----
DROP TABLE IF EXISTS `tipoproveedor`;

```

```

CREATE TABLE `tipoproveedor` (
  `cod_tipoproveedor` int(11) NOT NULL AUTO_INCREMENT,
  `tipo_descripcion` varchar(45) NOT NULL,
  `tipo_activo` enum('NO','SI') DEFAULT NULL,
  PRIMARY KEY (`cod_tipoproveedor`),
  UNIQUE KEY `tipo_descripcion` (`tipo_descripcion`),
  UNIQUE KEY `tipo_descripcion_2` (`tipo_descripcion`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for transaccion
-----

```

```

DROP TABLE IF EXISTS `transaccion`;
CREATE TABLE `transaccion` (
  `cod_transaccion` int(11) NOT NULL AUTO_INCREMENT,
  `tran_cantidad` int(11) DEFAULT NULL,
  `tran_precio` double DEFAULT NULL,
  `tran_descuento` double DEFAULT NULL,
  `tran_importe` double DEFAULT NULL,
  `cod_producto` int(11) NOT NULL,
  `cod_facturacion` int(11) NOT NULL,
  `tran_igv` double DEFAULT NULL COMMENT 'solo cuando es afecto al igven documento',
  `tran_total` double NOT NULL,
  `tran_lote` varchar(30) DEFAULT NULL,
  `tran_fechavencimiento` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`cod_transaccion`),
  KEY `fk_transaccion_producto1` (`cod_producto`),
  KEY `fk_transaccion_facturacion1` (`cod_facturacion`)
) ENGINE=MyISAM AUTO_INCREMENT=13 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for transaccion_compra
-----

```

```

DROP TABLE IF EXISTS `transaccion_compra`;
CREATE TABLE `transaccion_compra` (
  `cod_transaccion` int(11) NOT NULL AUTO_INCREMENT,
  `tran_cantidad` float DEFAULT NULL,
  `tran_precio` float DEFAULT NULL,
  `tran_descuento` float DEFAULT NULL,
  `tran_importe` double DEFAULT NULL,
  `cod_producto` int(11) NOT NULL,
  `cod_compra` int(11) NOT NULL,
  `tran_igv` double DEFAULT NULL,
  `tran_total` double NOT NULL COMMENT 'total=igv+importe',
  `tran_lote` varchar(20) DEFAULT NULL,
  `tran_fechavencimiento` date DEFAULT NULL,
  PRIMARY KEY (`cod_transaccion`),
  KEY `fk_transaccion_producto1` (`cod_producto`),
  KEY `fk_transaccion_compra_compra1` (`cod_compra`)
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for transportista
-----

```

```

DROP TABLE IF EXISTS `transportista`;
CREATE TABLE `transportista` (
  `cod_transporte` int(11) NOT NULL AUTO_INCREMENT,
  `tran_nombre` varchar(45) DEFAULT NULL,
  `tran_direccion` varchar(45) DEFAULT NULL,
  `tran_ruc` varchar(12) DEFAULT NULL COMMENT 'dni\n',
  `tran_distrito` varchar(45) DEFAULT NULL,
  `tran_provincia` varchar(45) DEFAULT NULL,
  `tran_movistar` varchar(15) DEFAULT NULL,
  `tran_claro` varchar(15) DEFAULT NULL,
  `tran_nextel` varchar(20) DEFAULT NULL,
  `tran_fijo` varchar(20) DEFAULT NULL,
  `tran_email` varchar(45) DEFAULT NULL,
  `tran_fax` varchar(45) DEFAULT NULL,
  `tran_placa` varchar(45) DEFAULT NULL,
  `tran_marca` varchar(45) DEFAULT NULL,
  `tran_vehiculo` varchar(45) DEFAULT NULL,
  `tran_certificado` varchar(45) DEFAULT NULL,
  `tran_licencia` varchar(45) DEFAULT NULL,
  `tran_activo` enum('SI','NO') DEFAULT NULL,
  `tran_chofer` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`cod_transporte`),
  UNIQUE KEY `tran_nombre` (`tran_nombre`),
  UNIQUE KEY `tran_ruc` (`tran_ruc`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

-----
-- Table structure for traspaso
-----

```

```

DROP TABLE IF EXISTS `traspaso`;

```

```

CREATE TABLE `traspaso` (
  `cod_traspaso` int(11) NOT NULL AUTO_INCREMENT,
  `cod_comprobante` int(11) NOT NULL,
  `tras_serie` varchar(6) NOT NULL,
  `tras_numero` varchar(10) NOT NULL,
  `cod_motivooperacion` int(11) NOT NULL,
  `tras_subtotal` double(11,5) NOT NULL DEFAULT '0.00000',
  `tras_igv` double(11,5) NOT NULL DEFAULT '0.00000',
  `tras_total` double(11,5) NOT NULL DEFAULT '0.00000' COMMENT 'valor total del traspaso',
  `tras_percepcion` double(11,5) NOT NULL DEFAULT '0.00000',
  `cod_almacen` int(11) NOT NULL COMMENT 'almacen origen',
  `cod_usuario` int(11) DEFAULT NULL,
  `tras_fecha_traspaso` datetime DEFAULT NULL,
  `tras_fecha_recepcion` datetime DEFAULT NULL,
  `tras_recepcionado` enum('NO','SI') NOT NULL,
  `tras_observacion` text,
  `tras_partida` varchar(100) DEFAULT NULL,
  `tras_llegada` varchar(100) DEFAULT NULL,
  `cod_almacendestino` int(11) NOT NULL DEFAULT '0',
  `cod_transportista` int(11) NOT NULL,
  `cod_vehiculo` int(11) NOT NULL DEFAULT '0',
  `cod_conductor` int(11) NOT NULL DEFAULT '0',
  `tras_motivo` varchar(100) DEFAULT NULL,
  `cod_proveedor` int(11) DEFAULT '0',
  `cod_cliente` int(11) DEFAULT '0',
  PRIMARY KEY (`cod_traspaso`)
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8;

```

-----  
-- Table structure for traspaso\_detalle

```

DROP TABLE IF EXISTS `traspaso_detalle`;
CREATE TABLE `traspaso_detalle` (
  `cod_detalletraspaso` int(11) NOT NULL AUTO_INCREMENT,
  `deta_cantidad` double(11,3) DEFAULT '0.000',
  `deta_precio` double(11,5) DEFAULT '0.00000',
  `deta_descuento` double(11,5) DEFAULT NULL,
  `deta_importe` double(11,5) DEFAULT NULL,
  `cod_producto` int(11) NOT NULL,
  `cod_traspaso` int(11) NOT NULL,
  `deta_igv` double(11,5) DEFAULT NULL,
  `deta_total` double(11,5) DEFAULT '0.00000',
  `deta_lote` varchar(20) DEFAULT NULL,
  `deta_fecha_vencimiento` date DEFAULT NULL,
  PRIMARY KEY (`cod_detalletraspaso`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;

```

-----  
-- Table structure for turnocaja

```

DROP TABLE IF EXISTS `turnocaja`;
CREATE TABLE `turnocaja` (
  `cod_turno` int(11) NOT NULL AUTO_INCREMENT,
  `turn_descripcion` enum('HE-MAÑANA','HS-MAÑANA','HE-TARDE','HS-TARDE','HE-NOCHE','HS-NOCHE') DEFAULT NULL,
  `turn_turno` enum('MAÑANA','TARDE','NOCHE') DEFAULT NULL,
  `turn_estado` enum('INGRESO','SALIDA') DEFAULT NULL,
  `turn_hora_entrada` time DEFAULT NULL,
  `turn_hora_salida` time DEFAULT NULL,
  `turn_eliminado` enum('SI','NO') DEFAULT NULL,
  PRIMARY KEY (`cod_turno`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

-----  
-- Table structure for ubigeo

```

DROP TABLE IF EXISTS `ubigeo`;
CREATE TABLE `ubigeo` (
  `cod_ubigeo` int(11) NOT NULL AUTO_INCREMENT,
  `ubig_departamento` varchar(250) DEFAULT NULL,
  `ubig_provincia` varchar(250) DEFAULT NULL,
  `ubig_distrito` varchar(250) DEFAULT NULL,
  PRIMARY KEY (`cod_ubigeo`)
) ENGINE=InnoDB AUTO_INCREMENT=1840 DEFAULT CHARSET=latin1;

```

-----  
-- Table structure for usuario

```

DROP TABLE IF EXISTS `usuario`;
CREATE TABLE `usuario` (
  `cod_usuario` int(11) NOT NULL AUTO_INCREMENT,
  `usua_usuario` varchar(100) CHARACTER SET latin1 DEFAULT NULL,

```

```

`usua_clave` varchar(20) CHARACTER SET latin1 DEFAULT NULL,
`usua_fecharegistro` date DEFAULT NULL,
`usua_activo` enum('SI','NO') CHARACTER SET latin1 DEFAULT NULL,
`cod_origen` int(11) NOT NULL,
`cod_perfil` int(11) NOT NULL,
`cod_almacen` int(11) DEFAULT NULL,
PRIMARY KEY (`cod_usuario`),
KEY `fk_Usuario_Personal1_idx` (`cod_origen`),
KEY `fk_Usuario_Perfil1_idx` (`cod_perfil`),
CONSTRAINT `usuario_ibfk_1` FOREIGN KEY (`cod_perfil`) REFERENCES `perfilusuario` (`cod_perfil`) ON DELETE NO
ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;

```

```

-----
-- View structure for vadsoft
-----

```

```

DROP VIEW IF EXISTS `vadsoft`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` VIEW `vadsoft` AS SELECT
facturacion.cod_facturacion AS CODIGO,
CONVERT(facturacion.fact_serie, UNSIGNED INT) AS SERIE,
CONVERT(facturacion.fact_numero, UNSIGNED INT) AS NUMERO,
DATE(facturacion.fact_fechaemision) AS FECFAC,
DATE(facturacion.fact_fechaemision) AS FECVEN,
if(CHAR_LENGTH(cliente.clie_ruc)=11,cliente.clie_ruc,") AS NRO_RUC,
if(strcmp(facturacion.fact_estado,_latin1'NO'),'ANULADO',if(strcmp(facturacion.cod_cliente,(_utf8'NULL' or
_utf8'0')),cliente.clie_nombre,facturacion.fact_alias)) AS NOMBRE,
comprobante.comp_observacion AS TIPDOC,
'SOLES' AS TIPMON,
" AS DETRAC,
if(strcmp(facturacion.fact_estado,_latin1'NO'),0,round(facturacion.fact_subtotal,2)) AS IMPVTA,
" AS IMPINA,
" AS RECARGO,
if(strcmp(facturacion.fact_estado,_latin1'NO'),0,round(facturacion.fact_igv,2)) AS IMPIGV,
if(strcmp(facturacion.fact_estado,_latin1'NO'),0,round(facturacion.fact_total,2)) AS IMPTOT,
if(strcmp(facturacion.fact_estado,_latin1'NO'),'A,") AS ST,
" AS SER_DQM,
" AS NUM_DQM,
" AS FEC_DQM,
" AS TIP_DQM,
" AS SERIE_FIN,
" AS NUMERO_FIN,
if(CHAR_LENGTH(cliente.clie_ruc)<>11,cliente.clie_ruc,")AS NRO_DNI,
" AS PASAPORTE
FROM
((((facturacion
LEFT JOIN usuario ON ((facturacion.cod_usuario = usuario.cod_usuario)))
JOIN comprobante ON ((comprobante.cod_comprobante = facturacion.cod_comprobante)))
JOIN almacen ON ((almacen.cod_almacen = facturacion.cod_almacen)))
LEFT JOIN cliente ON ((cliente.cod_cliente = facturacion.cod_cliente)))
WHERE (comprobante.cod_comprobante='1' OR comprobante.cod_comprobante='2' OR comprobante.cod_comprobante='3') ;

```

```

-----
-- View structure for vcliente
-----

```

```

DROP VIEW IF EXISTS `vcliente`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vcliente` AS SELECT
cliente.cod_cliente AS CODIGO,
cliente.clie_ruc AS RUC,
cliente.clie_nombre AS CLIENTE,
clientetipo.clti_nombre AS TIPOCLIENTE,
cliente.clie_direccion AS DIRECCION,
cliente.clie_departamento AS DEPARTAMENTO,
cliente.clie_provincia AS PROVINCIA,
cliente.clie_distrito AS DISTRITO,
cliente.clie_email AS EMAIL,
cliente.clie_movistar AS CELULAR,
cliente.clie_whatsapp AS WHATSAPP,
cliente.clie_fijo AS FIJO,
cliente.clie_nacimiento AS FNACIMIENTO,
cliente.clie_lineacredito,
cliente.clie_disponible,
cliente.clie_otorgado,
cliente.clie_activo AS ACTIVO,
cliente.cod_clientetipo
from `cliente`
left join `clientetipo` on((`clientetipo`.`cod_clientetipo` = `cliente`.`cod_clientetipo`));

```

```

-----
-- View structure for vcomisiontarjeta
-----

```

```

DROP VIEW IF EXISTS `vcomisiontarjeta`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vcomisiontarjeta` AS
SELECT
comisiontarjeta.cod_tarjetacomision as CODIGO,
tarjeta.tarj_tarjeta AS TARJETA,
comisiontarjeta.taco_tipo AS TIPO,

```



```

comisiontarjeta.taco_porcentaje AS PORCENTAJE,
tarjeta.cod_tarjeta AS CODTARJETA,
comisiontarjeta.taco_activo as ACTIVO
FROM
comisiontarjeta
INNER JOIN tarjeta ON comisiontarjeta.cod_tarjeta = tarjeta.cod_tarjeta ;

-----
-- View structure for vcompra
-----
DROP VIEW IF EXISTS `vcompra`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vcompra` AS SELECT
compra.cod_compra AS CODIGO,
comprobante.comp_descripcion AS COMPROBANTE,
compra.comp_serie AS SERIE,
compra.comp_numero AS NUMERO,
DATE(compra.comp_fechaemision) AS FEMISION,
compra.comp_pago AS FPAGO,
proveedor.prov_razonsocial AS PROVEEDOR,
proveedor.prov_ruc AS RUC,
proveedor.prov_direccion AS DIRECCION,
compra.comp_monto AS MONTO,
compra.comp_descuento AS DESCUENTO,
compra.comp_subtotal AS SUBTOTAL,
compra.comp_igv AS IGV,
compra.comp_total AS TOTAL,
compra.comp_percepcion as PERCEPCION,
if(moneda.cod_moneda=1,compra.comp_totalcompra,0) AS TOTALCOMPRA,
if(moneda.cod_moneda<>1,compra.comp_totalcompra,0) AS $TOTALCOMPRA,
compra.comp_tipocambio AS TC,
moneda.mone_descripcion AS MONEDA,
compra.comp_anulado AS ANULADO,
compra.comp_operacion AS OPERACION,
compra.comp_observacion AS OBERVACION,
compra.comp_motivo AS MOTIVO,
comprobante.comp_observacion AS TIPOCOMPROBANTE,
almacen.alma_descripcion AS ALMACEN,
usuario.usua_usuario AS usua_nombre,
usuario.usua_clave AS usua_celular,
usuario.usua_activo AS usua_estado,
compra.cod_moneda AS cod_moneda,
compra.cod_usuario AS cod_usuario,
compra.cod_comprobante AS cod_comprobante,
compra.cod_almacen AS cod_almacen,
compra.cod_proveedor AS cod_proveedor,
compra.cod_motivooperacion
from `compra`
left join `usuario` on `compra`.`cod_usuario` = `usuario`.`cod_usuario`
join `comprobante` on `comprobante`.`cod_comprobante` = `compra`.`cod_comprobante`
join `almacen` on `compra`.`cod_almacen` = `almacen`.`cod_almacen`
join `proveedor` on `proveedor`.`cod_proveedor` = `compra`.`cod_proveedor`
left join `moneda` on `compra`.`cod_moneda` = `moneda`.`cod_moneda` ;

-----
-- View structure for vdetallecompra
-----
DROP VIEW IF EXISTS `vdetallecompra`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` VIEW `vdetallecompra` AS SELECT
transaccion_compra.cod_transaccion AS CODIGO,
producto.prod_descripcion AS PRODUCTO,
transaccion_compra.tran_cantidad AS CANT,
transaccion_compra.tran_precio AS PRECIO,
transaccion_compra.tran_importe AS IMPORTE,
transaccion_compra.tran_descuento AS DESCUENTO,
transaccion_compra.tran_total AS TOTAL,
producto.cod_producto AS cod_producto,
transaccion_compra.cod_compra AS cod_compra,
DATE(compra.comp_fechaemision) AS comp_fechaemision,
comprobante.cod_comprobante AS cod_comprobante,
compra.comp_serie AS SERIE,
compra.comp_numero AS NUMERO,
comprobante.comp_descripcion as COMP,
proveedor.prov_razonsocial AS PROVEEDOR,
proveedor.prov_ruc AS RUC,
compra.comp_subtotal,
compra.comp_igv,
compra.comp_total,
compra.comp_percepcion,
compra.comp_totalcompra,
proveedor.cod_proveedor
from `producto`
join `transaccion_compra` on `transaccion_compra`.`cod_producto` = `producto`.`cod_producto`
join `compra` on `transaccion_compra`.`cod_compra` = `compra`.`cod_compra`
join `comprobante` on `compra`.`cod_comprobante` = `comprobante`.`cod_comprobante`
join `proveedor` on `compra`.`cod_proveedor` = `proveedor`.`cod_proveedor` ;

```

```

-----
-- View structure for vdetallefactura
-----
DROP VIEW IF EXISTS `vdetallefactura`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` VIEW `vdetallefactura` AS SELECT
transaccion.cod_transaccion AS CODIGO,
producto.prod_descripcion AS PRODUCTO,
transaccion.tran_cantidad AS CANT,
transaccion.tran_precio AS PRECIO,
transaccion.tran_importe AS IMPORTE,
transaccion.tran_descuento AS DESCUENTO,
transaccion.tran_total AS TOTAL,
transaccion.cod_producto,
transaccion.cod_facturacion,
date(facturacion.fact_fechaemision) as fact_fechaemision,
facturacion.cod_comprobante,
facturacion.fact_serie as SERIE,
facturacion.fact_numero as NUMERO,
comprobante.comp_descripcion as COMP,
facturacion.fact_estado
FROM
transaccion
INNER JOIN facturacion on facturacion.cod_facturacion=transaccion.cod_facturacion
LEFT JOIN producto ON producto.cod_producto = transaccion.cod_producto
inner join comprobante on comprobante.cod_comprobante=facturacion.cod_comprobante ;

-----
-- View structure for vdetalletraspaso
-----
DROP VIEW IF EXISTS `vdetalletraspaso`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vdetalletraspaso` AS
SELECT
traspaso_detalle.cod_detalletraspaso AS CODIGO,
producto.prod_descripcion AS PRODUCTO,
traspaso_detalle.deta_cantidad AS CANT,
traspaso_detalle.cod_producto,
traspaso_detalle.cod_traspaso,
date(traspaso.tras_fecha(traspaso)) as fact_fechaemision,
traspaso.cod_comprobante,
traspaso.tras_serie as SERIE,
traspaso.tras_numero as NUMERO,
comprobante.comp_descripcion as COMP,
traspaso.cod_motivooperacion AS cod_motivooperacion,
motivo_operacion.mope_nombre
FROM
traspaso_detalle
INNER JOIN traspaso on traspaso.cod_traspaso= traspaso_detalle.cod_traspaso
LEFT JOIN producto ON producto.cod_producto = traspaso_detalle.cod_producto
inner join comprobante on comprobante.cod_comprobante=traspaso.cod_comprobante
join motivo_operacion on traspaso.cod_motivooperacion=motivo_operacion.cod_motivooperacion ;

-----
-- View structure for vdocumento
-----
DROP VIEW IF EXISTS `vdocumento`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vdocumento` AS SELECT
documento.cod_documento AS cod_documento,
documento.docu_numero AS docu_numero,
documento.docu_afecto AS docu_afecto,
comprobante.cod_comprobante AS cod_comprobante,
comprobante.comp_descripcion AS comp_descripcion,
comprobante.comp_observacion AS comp_observacion,
serie.cod_serie AS cod_serie,
serie.seri_serie AS seri_serie,
almacen.cod_almacen AS cod_almacen,
almacen.alma_descripcion AS alma_descripcion,
almacen.alma_direccion AS alma_direccion,
documento.docu_estado AS docu_estado
from ((( `documento` join `serie` on(( `serie`.`cod_serie` = `documento`.`cod_serie` ))) join `almacen` on(( `almacen`.`cod_almacen` =
`documento`.`cod_almacen` ))) join `comprobante` on(( `comprobante`.`cod_comprobante` = `documento`.`cod_comprobante` )))
WHERE
documento.docu_estado = 'SI' ;

-----
-- View structure for vegresocaja
-----
DROP VIEW IF EXISTS `vegresocaja`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vegresocaja` AS SELECT
caja.cod_caja,
date(caja.caja_fecha) AS FECHA,
comprobante.comp_descripcion AS COMP,
caja.caja_serienumero AS NUMERO,
if(cliente.cod_cliente is null , proveedor.prov_ruc, cliente.clie_ruc) AS RUC,
if(cliente.cod_cliente is null , caja.caja_persona, cliente.clie_nombre) AS PERSONA,
caja.caja_monto AS MONTO,

```

```

caja.caja_detalle AS CONCEPTO,
caja.caja_observacion AS OBSERVACION,
if(tipomovimiento.timo_descripcion is null,"",tipomovimiento.timo_descripcion) as timo_descripcion,
caja.cod_facturacion,
caja.cod_egreso,
caja.cod_comprobante
FROM
caja
left JOIN tipomovimiento ON caja.cod_tipomovimiento = tipomovimiento.cod_tipomovimiento
LEFT JOIN cliente ON caja.cod_cliente = cliente.cod_cliente
LEFT JOIN proveedor ON proveedor.cod_proveedor = caja.cod_proveedor
LEFT JOIN comprobante ON comprobante.cod_comprobante = caja.cod_comprobante
where caja.caja_operacion='EGRESO' and caja.cod_tipomovimiento<>3 ;

```

```

-- -----
-- View structure for vfacturacion
-- -----

```

```

DROP VIEW IF EXISTS `vfacturacion`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vfacturacion` AS SELECT
facturacion.cod_facturacion AS CODIGO,
comprobante.comp_descripcion AS COMPROBANTE,
facturacion.fact_serie AS SERIE,
facturacion.fact_numero AS NUMERO,
date(facturacion.fact_fechaemision) AS FEMISION,
facturacion.fact_pago AS FPAGO,
if(facturacion.cod_cliente<>0,cliente.clie_nombre,facturacion.fact_alias) AS CLIENTE,
IF(facturacion.cod_cliente<>0,cliente.clie_ruc, "") as RUC,
IF(facturacion.cod_cliente<>0,cliente.clie_direccion, "") as DIRECCION,
facturacion.fact_monto AS IMPORTE,
facturacion.fact_descuento AS DESCT,
facturacion.fact_subtotal AS SUBTOTAL,
facturacion.fact_igv AS IGV,
facturacion.fact_total AS TOTAL,
facturacion.fact_estado AS ANULADO,
facturacion.fact_tarjeta AS TARJETA,
facturacion.fact_tipotarjeta AS TIPOTARJETA,
facturacion.fact_numoperacion AS NOOPERACION,
facturacion.fact_comisiontarjeta AS XCOMISION,
facturacion.fact_fechaanulacion AS FANULACION,
facturacion.cod_comprobante,
facturacion.cod_usuario,
facturacion.cod_cliente,
origen.orig_descripcion,
time(facturacion.fact_fechaemision) AS HORA
FROM
facturacion
LEFT JOIN cliente ON cliente.cod_cliente = facturacion.cod_cliente
INNER JOIN comprobante ON comprobante.cod_comprobante = facturacion.cod_comprobante
left JOIN usuario on usuario.cod_usuario=facturacion.cod_usuario
left join origen on origen.cod_origen=usuario.cod_origen ;

```

```

-- -----
-- View structure for vmovimientocaja
-- -----

```

```

DROP VIEW IF EXISTS `vmovimientocaja`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vmovimientocaja` AS
SELECT
caja.cod_caja AS CODIGO,
DATE(caja.caja_fecha) AS FECHA,
TIME(caja.caja_fecha) AS HORA,
if(comprobante.cod_comprobante<>'0',comprobante.comp_descripcion,'NINGUNO') AS DCTO,
caja.caja_serienumero AS NUMERO,
tipomovimiento.timo_descripcion AS TIPO,
caja.caja_detalle AS GLOSA,
IF(caja.cod_cliente<>0 OR caja.cod_cliente is null, cliente.clie_nombre,caja.caja_persona) AS ORIGEN,
FORMAT(IF(caja.caja_operacion='INGRESO',caja.caja_monto,'0'),2) AS INGRESO,
FORMAT(IF(caja.caja_operacion='EGRESO',caja.caja_monto,'0'),2) AS EGRESO,
FORMAT((IF(caja.caja_operacion='INGRESO',caja.caja_monto,'0')-IF(caja.caja_operacion='EGRESO',caja.caja_monto,'0') +
saldo-caja(caja.caja_fecha,caja.cod_caja,TIME(caja.caja_fecha))),2) AS SALDO,
caja.caja_persona AS ENCARGADO,
caja.caja_observacion AS OBSERVACION,
caja.caja_afecta AS AFECTACAJA,
caja.caja_operacion AS OPERACION,
caja.cod_cliente,
caja.cod_proveedor,
caja.cod_usuario,
caja.cod_comprobante,
caja.cod_facturacion,
caja.cod_tipomovimiento,
caja.caja_fecharegistro
FROM
caja
LEFT JOIN comprobante ON caja.cod_comprobante = comprobante.cod_comprobante
LEFT JOIN facturacion ON caja.cod_facturacion = facturacion.cod_facturacion
LEFT JOIN cliente ON caja.cod_cliente = cliente.cod_cliente
LEFT JOIN proveedor ON caja.cod_proveedor = proveedor.cod_proveedor

```

```

left JOIN usuario ON caja.cod_usuario = usuario.cod_usuario
left JOIN tipomovimiento ON tipomovimiento.cod_tipomovimiento = caja.cod_tipomovimiento
order by caja.caja_fecha asc ;

```

```

-- -----
-- View structure for vofertaproducto
-- -----

```

```

DROP VIEW IF EXISTS `vofertaproducto`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vofertaproducto` AS
SELECT
ofertaproducto.cod_oferta,
producto.prod_descripcion as PRODUCTO,
producto.prod_serinumber AS CODIGOBARRA,
producto.prod_preciocontado AS PRECIOVENTA,
ofertaproducto.ofer_descuento AS OFERTA,
ofertaproducto.ofer_precio AS PRECIOOFERTA,
ofertaproducto.ofer_observacion AS OBSERVACION,
ofertaproducto.ofer_fechainicio AS FINICIO,
ofertaproducto.ofer_fechavencimiento AS FFINAL,
categoria.cate_descripcion AS LINEA,
ofertaproducto.ofer_activado as ESTADO,
producto.cod_producto
FROM
ofertaproducto
INNER JOIN producto ON ofertaproducto.cod_producto = producto.cod_producto
LEFT join categoria on categoria.cod_categoria=producto.cod_categoria ;

```

```

-- -----
-- View structure for vproducto
-- -----

```

```

DROP VIEW IF EXISTS `vproducto`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vproducto` AS SELECT
producto.cod_producto AS CODIGO,
producto.prod_serinumber AS BARRA,
producto.prod_descripcion AS PRODUCTO,
producto.prod_nombrecorto AS NOMBRECORTO,
cocina.coci_descripcion AS COCINA,
categoria.cate_descripcion AS TIPO,
producto.prod_costo AS COSTO,
if(producto.prod_oferta='SI',producto.prod_preciooferta, producto.prod_precio) AS PRECIO,
producto_almacen.proalm_stock AS STOCK,
producto_almacen.proalm_stockinicial AS STOCKINICIAL,
producto.prod_stockmin AS STOCKMIN,
producto.prod_tipocambio AS TC,
producto.cod_categoria,
cocina.cod_cocina,
producto.prod_oferta,
producto.prod_xoferta,
producto.prod_preciooferta,
producto.prod_cantpromocion,
producto.prod_activo,
almacen.alma_descripcion,
almacen.cod_almacen,
producto.prod_observacion,
producto.prod_imagen AS IMAGEN,
producto.prod_fecharegistro AS FREGISTRO,
producto.prod_preciocontado AS PRECIO2,
producto.prod_descuento AS DESCUENTO,
producto.prod_porganancia AS XGANANCIA,
producto.prod_serinumber AS FUENTE BARRA,
producto.prod_tipo
FROM
producto
LEFT JOIN categoria ON producto.cod_categoria = categoria.cod_categoria
LEFT JOIN producto_almacen ON producto.cod_producto = producto_almacen.cod_producto
INNER JOIN almacen ON producto_almacen.cod_almacen = almacen.cod_almacen
LEFT JOIN cocina ON producto.cod_cocina = cocina.cod_cocina
ORDER by producto.cod_producto asc ;

```

```

-- -----
-- View structure for vproveedor
-- -----

```

```

DROP VIEW IF EXISTS `vproveedor`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vproveedor` AS SELECT
proveedor.cod_proveedor,
proveedor.prov_ruc AS RUC,
proveedor.prov_razonsocial AS PROVEEDOR,
tipoproveedor.tipo_descripcion AS TIPO,
proveedor.prov_direccion AS DIRECCION,
proveedor.prov_referencia AS REFERENCIA,
proveedor.prov_gironegocio AS GIRONEGOCIO,
proveedor.prov_celular AS CELULAR,
proveedor.prov_fijo AS TFIJO,
proveedor.prov_email AS EMAIL,
proveedor.prov_nrocuenta AS NROCuenta,

```

```

proveedor.prov_departamento AS DEPARTAMENTO,
proveedor.prov_provincia AS PROVINCIA,
proveedor.prov_distrito AS DISTRITO,
proveedor.prov_activo AS ACTIVO,
proveedor.cod_tipoproveedor
FROM
proveedor
LEFT JOIN tipoproveedor ON proveedor.cod_tipoproveedor = tipoproveedor.cod_tipoproveedor ;

-- -----
-- View structure for vrepkardexcontable
-- -----
DROP VIEW IF EXISTS `vrepkardexcontable`;
CREATE ALGORITHM=UNDEFINED DEFINER='root'@'localhost' SQL SECURITY DEFINER VIEW `vrepkardexcontable` AS
SELECT
kardex.cod_kardex AS cod_kardex,
date(kardex.kard_fecha) AS FECHA,
comprobante.comp_descripcion AS COMP,
if(kardex.`cod_traspaso`<>'0',traspaso.tras_numero,if(kardex.cod_facturacion<>'0',facturacion.fact_numero,compra.comp_numero
)) AS NUMERO,
producto.prod_descripcion AS PRODUCTO,
motivo_operacion.mope_nombre AS OPERACION,
if(strcmp(kardex.kard_operacion,'EGRESO'),0,kardex.kard_movimiento) AS EGRESO,
if(strcmp(kardex.kard_operacion,'INGRESO'),0,kardex.kard_movimiento) AS INGRESO,
if(STRCMP(kardex.kard_operacion,'INGRESO'),0,kardex.kard_movimiento)-if(STRCMP(kardex.kard_operacion,'EGRESO'),0,
kardex.kard_movimiento) + kardexstockinicial('2013-01-01 00:00:00', kardex.kard_fecha,kardex.cod_almacen ,
kardex.cod_producto,kardex.cod_kardex,time(kardex.kard_fecha),kardex.kard_movimiento ) AS STOCKACTUAL,
kardex.cod_producto AS cod_producto,
kardex.cod_almacen as cod_almacen,
time(kardex.kard_fecha) AS hora
from `kardex`
left join `compra` on `compra`.`cod_compra` = `kardex`.`cod_compra`
left join `facturacion` on `facturacion`.`cod_facturacion` = `kardex`.`cod_facturacion`
left join `traspaso` on `traspaso`.`cod_traspaso` = `kardex`.`cod_traspaso`
join `producto` on `kardex`.`cod_producto` = `producto`.`cod_producto`
join `comprobante` on `kardex`.`cod_comprobante` = `comprobante`.`cod_comprobante`
left join `cliente` on `facturacion`.`cod_cliente` = `cliente`.`cod_cliente`
left join `proveedor` on `proveedor`.`cod_proveedor` = `compra`.`cod_proveedor`
join `motivo_operacion` on `motivo_operacion`.`cod_motivooperacion` = `kardex`.`cod_motivooperacion`
order by producto.prod_descripcion, kardex.kard_fecha, TIME(kardex.kard_fecha) asc ;

-- -----
-- View structure for vtraslado
-- -----
DROP VIEW IF EXISTS `vtraslado`;
CREATE ALGORITHM=UNDEFINED DEFINER='root'@'localhost' SQL SECURITY DEFINER VIEW `vtraslado` AS
SELECT
traspaso.cod_traspaso AS cod_traspaso,
traspaso.cod_comprobante AS cod_comprobante,
traspaso.tras_fecha_traspaso AS tras_fecha_traspaso,
traspaso.tras_fecha_recepcion AS tras_fecha_recepcion,
comprobante.comp_descripcion AS comp_descripcion,
traspaso.tras_serie AS tras_serie,
traspaso.tras_numero AS tras_numero,
traspaso.tras_motivo,
traspaso.tras_partida AS tras_partida,
traspaso.tras_llegada AS tras_llegada,
traspaso.tras_subtotal AS tras_subtotal,
traspaso.tras_igv AS tras_igv,
traspaso.tras_total AS tras_total,
traspaso.tras_percepcion AS tras_percepcion,
traspaso.tras_observacion AS tras_observacion,
almacen.alma_descripcion AS alma_descripcion,
almacen.alma_direccion AS alma_direccion,
traspaso.cod_almacen AS cod_almacen,
traspaso.cod_usuario AS cod_usuario,
traspaso.cod_almacendestino AS cod_almacendestino,
traspaso.cod_transportista AS cod_transportista,
traspaso.cod_vehiculo AS cod_vehiculo,
traspaso.cod_conductor AS cod_conductor,
traspaso.cod_motivooperacion AS cod_motivooperacion,
usuario.usua_usuario,
traspaso.tras_recepcionado,
traspaso.cod_proveedor,
motivo_operacion.mope_nombre
from `traspaso` join `comprobante`
on `comprobante`.`cod_comprobante` = `traspaso`.`cod_comprobante`
left join `usuario` on `usuario`.`cod_usuario` = `traspaso`.`cod_usuario`
LEFT join `almacen` on `almacen`.`cod_almacen` = `traspaso`.`cod_almacendestino`
join motivo_operacion on traspaso.cod_motivooperacion=motivo_operacion.cod_motivooperacion ;

-- -----
-- View structure for vusuario
-- -----
DROP VIEW IF EXISTS `vusuario`;
CREATE ALGORITHM=UNDEFINED DEFINER='root'@'localhost' SQL SECURITY DEFINER VIEW `vusuario` AS
SELECT
usuario.cod_usuario,
origen.orig_descripcion as PERSONAL,

```

```

perfilusuario.perf_descripcion AS CARGO,
usuario.usua_usuario AS USUARIO,
usuario.usua_clave AS CLAVE,
usuario.usua_activo AS ACTIVO,
usuario.cod_origen,
usuario.cod_perfil,
usuario.cod_almacen,
almacen.alma_descripcion AS ALMACEN
FROM
usuario
INNER JOIN perfilusuario ON usuario.cod_perfil = perfilusuario.cod_perfil
INNER JOIN almacen ON usuario.cod_almacen = almacen.cod_almacen
INNER JOIN origen ON origen.cod_origen = usuario.cod_origen ;

-----
-- Function structure for kardexstockinicial
-----
DROP FUNCTION IF EXISTS `kardexstockinicial`;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` FUNCTION `kardexstockinicial`(finicio datetime,ffinal datetime,codalmacen int,codproducto
int,codkardex int, horak time,cantidad float) RETURNS float(11,3)
    DETERMINISTIC
BEGIN
    DECLARE stockinicial FLOAT;
    DECLARE ingreso FLOAT;
    DECLARE salida FLOAT;
    declare contarsalida int;
    declare contaringreso int;
    DECLARE salidaactual float;
    DECLARE inicial float;
    SET salida=0;
    SET ingreso=0;
    SET salidaactual=0;
    set contarsalida=0;
    set contaringreso=0;
    set inicial=0;
    #DECLARE fechainico DATE;
    #DECLARE fechafinal DATE;
    SET inicial =(SELECT proalm_stockinicial FROM `producto_almacen` WHERE `cod_almacen` = codalmacen and
cod_producto = codproducto);
    SET contarsalida =(SELECT COUNT(cod_kardex) FROM `kardex` WHERE `cod_almacen` = codalmacen AND
DATE(kard_fecha) = DATE(ffinal) and time(kard_fecha)=horak and cod_producto = codproducto AND cod_motivooperacion !=15
AND cod_motivooperacion !=16 AND cod_motivooperacion !=17 AND kard_operacion = 'EGRESO' and cod_kardex<=codkardex);

    SET contaringreso =(SELECT COUNT(cod_kardex) FROM `kardex` WHERE `cod_almacen` = codalmacen AND
DATE(kard_fecha) = DATE(ffinal) and time(kard_fecha)=horak and cod_producto = codproducto AND cod_motivooperacion !=15
AND cod_motivooperacion !=16 AND cod_motivooperacion !=17 AND kard_operacion = 'INGRESO' and cod_kardex<=codkardex);

    if contarsalida>1 then
        SET salida =(SELECT IF( SUM( kard_movimiento ) IS NULL , 0, SUM( kard_movimiento ) ) FROM `kardex`
WHERE `cod_almacen` = codalmacen AND kard_fecha <= ffinal and cod_producto = codproducto AND cod_motivooperacion !=15
AND cod_motivooperacion !=7 AND cod_motivooperacion !=16 AND cod_motivooperacion !=17 AND kard_operacion = 'EGRESO'
and cod_kardex<codkardex);
        SET salida=salida;
    else
        SET salida =(SELECT IF( SUM( kard_movimiento ) IS NULL , 0, SUM( kard_movimiento ) ) FROM `kardex`
WHERE `cod_almacen` = codalmacen AND kard_fecha < ffinal and cod_producto = codproducto AND cod_motivooperacion !=15
AND cod_motivooperacion !=7 AND cod_motivooperacion !=16 AND cod_motivooperacion !=17 AND kard_operacion = 'EGRESO');
    end if;

    if contaringreso>1 then
        SET ingreso =(SELECT IF( SUM( kard_movimiento ) IS NULL , 0, SUM( kard_movimiento ) ) FROM `kardex`
WHERE `cod_almacen` = codalmacen AND kard_fecha <= ffinal AND cod_producto = codproducto AND cod_motivooperacion !=15
AND cod_motivooperacion !=7 AND cod_motivooperacion !=16 AND cod_motivooperacion !=17 AND kard_operacion = 'INGRESO'
and cod_kardex<codkardex);
        SET ingreso=ingreso;
    else
        SET ingreso =(SELECT IF( SUM( kard_movimiento ) IS NULL , 0, SUM( kard_movimiento ) ) FROM `kardex`
WHERE `cod_almacen` = codalmacen AND kard_fecha < ffinal AND cod_producto = codproducto AND cod_motivooperacion !=15
AND cod_motivooperacion !=7 AND cod_motivooperacion !=16 AND cod_motivooperacion !=17 AND kard_operacion =
'INGRESO');
    end if;

    SET stockinicial=ingreso-salida+inicial;

    RETURN stockinicial;
END
;;
DELIMITER ;

-----
-- Function structure for saldocaja
-----
DROP FUNCTION IF EXISTS `saldocaja`;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` FUNCTION `saldocaja`(ffinal datetime, codcaja int, horak time) RETURNS float(11,3)

```

```

DETERMINISTIC
BEGIN
    DECLARE saldo FLOAT;
    DECLARE ingreso FLOAT;
    DECLARE salida FLOAT;
    declare contarsalida int;
    declare contaringreso int;
    DECLARE salidaactual float;
    SET salida=0;
    SET ingreso=0;
    SET salidaactual=0;
    set contarsalida=0;
    set contaringreso=0;
    #DECLARE fechainico DATE;
    #DECLARE fechafinal DATE;
    SET contarsalida =(SELECT COUNT(cod_caja) FROM `caja` WHERE DATE(caja_fecha) = DATE(ffinal) and
time(caja_fecha)=horak and caja_operacion = 'EGRESO' and cod_caja<=codcaja);

    SET contaringreso =(SELECT COUNT(cod_caja) FROM `caja` WHERE dATE(caja_fecha) = DATE(ffinal) and
time(caja_fecha)=horak and caja_operacion = 'INGRESO' and cod_caja<=codcaja);

    if contarsalida>1 then
        SET salida =(SELECT IF( SUM( caja_monto ) IS NULL , 0, SUM( caja_monto ) ) FROM `caja` WHERE
caja_fecha <= ffinal and caja_operacion = 'EGRESO' and cod_caja<codcaja);
        SET salida=salida;
    else
        SET salida =(SELECT IF( SUM( caja_monto ) IS NULL , 0, SUM( caja_monto ) ) FROM `caja` WHERE
caja_fecha < ffinal AND caja_operacion = 'EGRESO');
    end if;

    if contaringreso>1 then
        SET ingreso =(SELECT IF( SUM( caja_monto ) IS NULL , 0, SUM( caja_monto ) ) FROM `caja` WHERE
caja_fecha <= ffinal and caja_operacion = 'INGRESO' and cod_caja<codcaja);
        SET ingreso=ingreso;
    else
        SET ingreso =(SELECT IF( SUM( caja_monto ) IS NULL , 0, SUM( caja_monto ) ) FROM `caja` WHERE
caja_fecha < ffinal AND caja_operacion = 'INGRESO');
    end if;

    SET saldo=ingreso-salida;

    RETURN saldo;
END
;;
DELIMITER ;

```

#### 4.7. PRUEBAS

Para documentar las pruebas del software se ha diseñado un instrumento o tabla en la cual se especifican las pruebas realizadas a cada módulo y los resultados obtenidos.

Se realizarán las pruebas de cada módulo a través de la ejecución de una tarea específica. Es importante destacar que no se está siguiendo estrictamente una metodología de pruebas de ingeniería de software, sino que se está haciendo uso de los pasos esenciales de dichas metodologías para la aplicación específica desarrollada, en la que se espera obtener los diferentes resultados del sistema

OPCIÓN O MÓDULO A PROBAR	DESCRIPCIÓN DE CASO DE PRUEBA	PRE CONDICIONES O DATOS	PASOS	RESULTADO ESPERADO	RESULTADO REAL	OBSERVACIONES /COMENTARIOS	ESTADO (CONCLUIDO, PENDIENTE, EN PROCESO)
<b>Inicio de</b>	Verificar que el inicio de sesión sea exitoso al ingresar con los datos de un usuario valido	El sistema debe permitir visualizar la pantalla de inicio con sus campos correspondientes.	El usuario ingresa el nombre de usuario, contraseña y hace click en iniciar.	ingresar a la interfaz correspondiente al usuario autenticado	Correcto	Ninguna	Concluido
<b>sesión</b>	Verificar que al iniciar sesión con datos no validos el sistema indique que estos son incorrectos	El sistema debe permitir visualizar la pantalla de inicio con sus campos correspondientes	Se ingresa el nombre de la usuario, contraseña y click en iniciar.	El sistema debe validar los datos e indicar que los ingresados son incorrectos	Incorrecto	Es sistema muestra un error de código	Concluido



	Verificar que al iniciar sesión con datos no validos el sistema indique que estos son incorrectos	El sistema debe permitir visualizar la pantalla de inicio con sus campos correspondientes	Se ingresa el nombre de usuario y contraseña, teniendo en cuenta que uno de estos dos no sea válido; se hace click en ingresar	El sistema debe validar los datos e indicar que los datos ingresados son incorrectos	correcto	Se ha realizado Las modificaciones pertinentes del código y corregido el error que presentaba	Concluido
<b>Administrar</b>	Verificar que el sistema cree cuentas de usuarios de forma correcta	Haber ingresado al sistema como administrador y haber seleccionado las opciones Administrar Usuarios y Nuevo Usuario.	Se ingresan los datos correspondientes "Nombre, Apellido, etc..." y por ultimo selecciona la opción agregar	El sistema debe crear el usuario y mostrar un mensaje en el cual indica que la tarea ha sido exitosa	Correcto	Ninguna	Correcto
<b>usuario</b>	Verificar que los campos solo permitan ingresar el tipo de datos que le corresponde (numérico o alfanumérico)	Haber ingresado al sistema como administrador y estar ubicado en la ventana del formulario de nuevo usuario	Se ingresan en los diferentes campos del formulario datos que no corresponden a estos, por ejemplo: Si se tiene un campo teléfono el cual espera solo números se le ingresan letras, etc...	El sistema debe mostrar un mensaje de error en cual indique que el tipo de datos ingresado en el campo que corresponde	Incorrecto	El sistema permite ingresar tipos de datos en campos que no corresponde y continúa con el proceso.	Pendiente

<b>Administrar</b>	Verificar que al ingresar espacios en blanco en los campos requeridos para la creación de un nuevo usuario emita el error de campos requeridos	Haber ingresado al sistema como administrador y estar ubicado en la ventana del formulario de nuevo usuario	Se ingresan en los diferentes campos del formulario espacios en blanco. Por ejemplo en el campo nombre: ingresar únicamente espacios	El sistema debe mostrar un mensaje de error que indique que los campos son requeridos.	Correcto	El sistema muestra un mensaje de error indicando que es necesario ingresar la información de los campos del formulario	Concluido
	Verificar que al realizar la acción eliminar usuario, efectivamente lo elimine del sistema	Haber ingresado al sistema como administrador y estar ubicado en la ventana del eliminar usuario	Se ingresa a la opción eliminar usuario, se selecciona el usuario que se desea eliminar y finalmente se selecciona la opción "Eliminar usuario"	El sistema debe mostrar un mensaje sobre el éxito de la tarea realizada	Correcto	El sistema muestra un mensaje indicando que la tarea ha sido realizada con éxito	Concluido
	Verificar que en la sección eliminar usuario, al seleccionar la opción "eliminar usuario" sin haber seleccionado uno previamente aparezca un	Haber ingresado al sistema como administrador y estar ubicado en la ventana del eliminar usuario	Se ingresa a la opción eliminar usuario y finalmente se selecciona la opción "Eliminar usuario"	El sistema debe mostrar un mensaje de error que indique que el campo usuario es necesario	Correcto	El sistema muestra un mensaje indicando que el campo usuario es necesario	Concluido
<b>usuario</b>							

	mensaje de error						
<b>Administrar inventario</b>	Verificar que al ingresar una nueva categoría dentro del inventario, se agregue correctamente al mismo	Haber ingresado al sistema como administrador y estar ubicado en la ventana de "ingresar categoría"	Dentro de la opción "producto" seleccionar la opción "ingreso al almacén, ingresar categoría", en el campo nueva categoría	El sistema debe mostrar un mensaje sobre el éxito de la tarea realizada	Correcto	El sistema muestra un mensaje indicando que la tarea ha sido realizada con éxito	Concluido
	Verificar que al realizar la acción ingresar nueva categoría sin diligenciar el campo correspondiente el sistema muestre un mensaje de error informando sobre la acción	Haber ingresado al sistema como administrador y estar ubicado en la ventana de "ingresar categoría"	Dentro de la opción "Producto ingreso al almacén" diligenciar el formulario con los datos requeridos y finalmente seleccionar la opción agregar	El sistema debe mostrar un mensaje de error que indique que el campo es necesario	Correcto	El sistema muestra un mensaje indicando que el campo usuario es necesario	Concluido
	Verificar que al seleccionar la opción agregar sin ingresar los datos del formulario el sistema muestre un mensaje de error	Haber ingresado al sistema como administrador y estar ubicado en la ventana de "nuevo inventario"	Dentro de la opción "Producto" seleccionar la opción "ingreso al almacén, diligenciar el	El sistema debe mostrar un mensaje de error que indique que el campo usuario es necesario	Correcto	El sistema muestra un mensaje indicando que la tarea ha sido realizada con éxito	Concluido

			formulario con los datos requeridos y finalmente seleccionar la opción agregar				
	Verificar que al ingresar los datos del formulario para realizar un nuevo inventario, el sistema muestre un mensaje de éxito sobre dicha tarea	Haber ingresado al sistema como administrador y estar ubicado en la ventana de "ingresar al almacén"	Dentro de la opción "producto" seleccionar la opción "ingresar al almacén", diligenciar el formulario y finalmente seleccionar la opción agregar.	El sistema debe mostrar un mensaje sobre el éxito de la tarea realizada	Correcto	El sistema muestra un mensaje indicando que los campos son necesarios	Concluido
	Verificar que al seleccionar la opción "nuevo producto" y al ingresar los datos correspondientes el sistema muestre un mensaje informando sobre el éxito de la tarea	Haber ingresado al sistema como administrador y estar ubicado en la ventana de ingresar productos en la opción nuevo producto	Dentro de la opción "producto", luego diligenciar los campos del formulario y finalmente seleccionar la opción agregar	El sistema debe mostrar un mensaje indicando que la tarea ha sido realizada con éxito	Correcto	El sistema muestra un mensaje indicando el éxito de la tarea realizada	Concluido
	Verificar que al agregar un nuevo producto no permita ingresar el producto en caso de que este exista	Haber ingresado al sistema como administrador, encontrarse en la sesión producto y haber	Dentro de la opción "producto" seleccionar la opción "nuevo producto", luego ingresar un	El sistema deberá mostrar un mensaje indicando que el campo nombre debe contener un valor	Incorrecto	El sistema almacena dos productos con el mismo nombre, precio y clase	Concluido

		seleccionado la opción ingresar producto	producto que ya exista y finalmente seleccionar la opción agregar	único, es decir que este ya existe			
<b>Administrar</b>	Verificar que al seleccionar la opción "modificar producto", al seleccionar la opción editar producto el sistema permita modificar correspondientes a cada producto	Haber ingresado al sistema como administrador y estar ubicado en la ventana de productos en la opción modificar producto, agregar	Dentro de la opción "producto" seleccionar la opción "modificar producto", seleccionar del menú desplegable el producto a ser modificado, seleccionar la opción editar	El sistema debe mostrar el producto con la lista de que han sido agregados	Correcto	El sistema muestra los productos agregados	Concluido
<b>productos</b>	Verificar que al seleccionar la opción "modificar producto", al seleccionar la opción editar y agregar el sistema permita visualizarse por el usuario	Haber ingresado al sistema como administrador y estar ubicado en la ventana de productos en la opción modificar producto, agregar	Dentro de la opción "producto" seleccionar la opción "modificar producto", seleccionar del menú desplegable el producto a ser modificado	El sistema debe mostrar el producto con la lista cantidad de que han sido agregados	Correcto	El sistema muestra el producto con la cantidad de cada ingrediente	Concluido

<p><b>Administrar productos</b></p>	<p>Verificar que al seleccionar un producto para eliminar, este sea correctamente eliminado del sistema</p>	<p>Haber ingresado al sistema como administrador y estar ubicado en la ventana de productos en la opción eliminar producto.</p>	<p>Dentro de la opción "producto" seleccionar el producto que se desea eliminar y finalmente seleccionar la opción "eliminar producto"</p>	<p>El sistema debe mostrar un mensaje indicando que la tarea ha sido realizada con éxito</p>	<p>Correcto</p>	<p>El sistema muestra un mensaje indicando que la tarea ha sido realizada con éxito</p>	<p>Concluido</p>
<p><b>Administrar</b></p>	<p>Verificar que al crear una nueva mesa el sistema realice la tarea correctamente</p>	<p>Haber ingresado al sistema como administrador y estar ubicado en la opción mesa</p>	<p>Dentro de la opción nueva mesa ingresar el nombre de la mesa, el cual debe contener caracteres alfabéticos y un identificador numérico y finalmente seleccionar la opción agregar</p>	<p>El sistema debe mostrar un mensaje indicando que la tarea ha sido realizada con éxito</p>	<p>Correcto</p>	<p>El sistema muestra un mensaje indicando que la tarea ha sido realizada con éxito</p>	<p>Concluido</p>

<b>mesas</b>	Verificar que al eliminar una mesa el sistema realice la tarea correctamente	Haber ingresado al sistema como administrador y estar ubicado en la opción eliminar mesa	En la opción eliminar mesa seleccionar la mesa que se desea eliminar, seguido la opción eliminar	El sistema debe mostrar un mensaje indicando que la tarea ha sido realizada con éxito	Correcto	El sistema muestra un mensaje indicando que la tarea ha sido realizada con éxito	Concluido
<b>Consultar ventas</b>	Verificar que el sistema muestre al usuario las ventas consultadas en un rango de tiempo determinado	Haber ingresado al sistema como administrador, y estas ubicado en la opción Consultar Ventas	Se ingresa en los campos consultar desde y Consultar hasta las fechas entre las cuales se desea realizar la consulta	El sistema debe proporcionar la información correspondiente al rango de tiempo	Correcto	El sistema efectivamente despliega en la ventana la información que corresponde a las fechas consultadas	Concluido

#### 4.8. IMPLANTACIÓN DEL SISTEMA

Este programa básicamente está diseñado para administrar y controlar las ventas en caja y almacén, de los productos que ingresan y salen

##### **En caja:**

Cobranza a los comensales, ya que en esta opción se ingresa los pedidos de las mesas correspondientes y posteriormente son atendidos (esto solo sucede los fines de semana y feriado---- primero se pagan luego son atendidos) y los días normales (primero son los pedidos y luego se pagan)

**Almacén:** ingreso y salida de los productos para la elaboración de las comidas, y otros productos como gaseosas, cervezas, etc.

Seguidamente se presenta una visión general de los apartados de la aplicación:

- **Inicio**, donde nos pedirá que insertemos nuestro usuario y contraseña para validarnos y darnos acceso al sistema.
- **Menú principal**, mostrará un menú superior para acceder a todas las secciones del sistema.
- **Producto**, permitirá una vista de los submenús de: configuración donde el usuario podrá ingresar o modificar los datos; platos, productos, almacén y reporte.
- **Ventas** este menú permitirá la gestión de los clientes, ventas reimprimir y anular venta, Reporte, Pedido mesero, Anular Pedido.
- **Caja**, este menú no permitirá hacer el cierre de caja, registro de egreso y movimiento de caja.
- **Mantenimiento**, permitirá hacer el mantenimiento de la base de datos, configuración, personal, usuario, inicializar caja y ADESOFT.



A continuación, detallamos las interfaces del sistema.



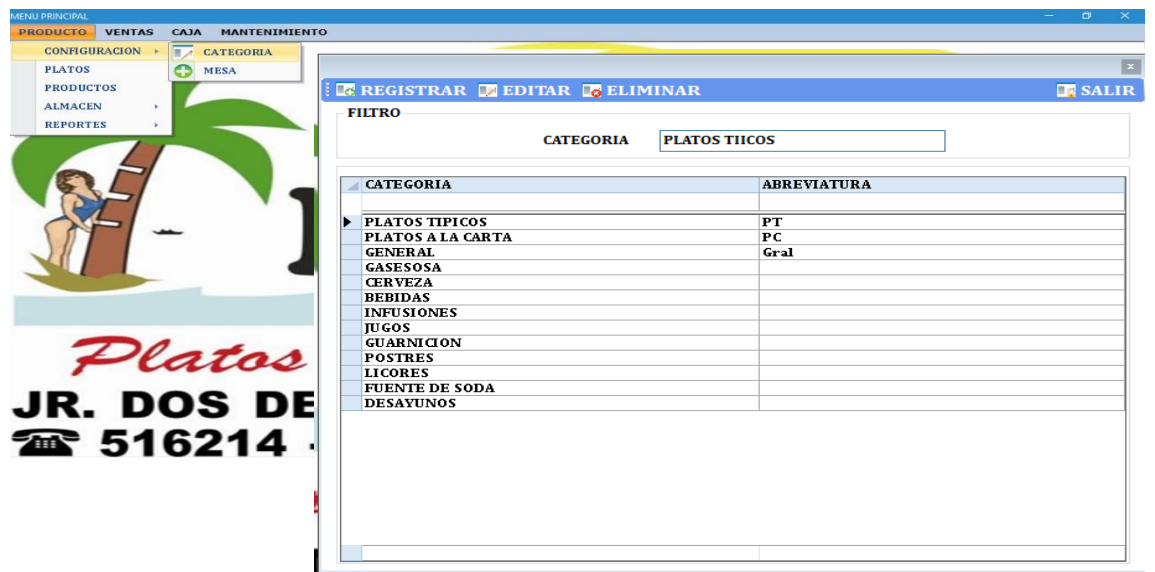
The screenshot shows a window titled 'INICIO DE SESION'. On the left is a logo with a yellow sun and the text 'RECREO FALCON'. On the right, there are two input fields labeled 'USUARIO' and 'CONTRASEÑA'. Below these fields are two buttons: 'INGRESAR' and 'CANCELAR'.

Este módulo permite la validación de todos los usuarios registrados en la base de datos del sistema, se presenta el controlador de dicho módulo, donde se validan los datos identificación y contraseña de usuario



Este módulo es el mayor contenedor de métodos de toda la aplicación, contiene los módulos de Productos, Ventas, Caja y Mantenimiento.

## MENÚ PRODUCTO, SUB MENÚ CONFIGURACIÓN, OPCIÓN CATEGORÍA



Esta opción nos permite crear las categorías de los productos donde se puede registrar, editar o eliminar las categorías.

## MENÚ PRODUCTO, SUB MENÚ CONFIGURACIÓN, OPCIÓN MESA

Esta opción nos permite crear las Mesas del salón, esta opción se puede registrar, editar o eliminar las mesas.

MENU PRINCIPAL

**PRODUCTO** VENTAS CAJA MANTENIMIENTO

CONFIGURACION >

- PLATOS
- PRODUCTOS
- ALMACEN >
- REPORTES >

CATEGORIA

- MESA

REGISTRAR EDITAR ELIMINAR SALIR

FILTRO

MESA

MESA
MESA 01
MESA 02
MESA 03
MESA 04
MESA 05
MESA 06
MESA 07
MESA 08
MESA 09
MESA 10
MESA 11
MESA 12
MESA 13
MESA 14
MESA 15
MESA 16



# MENÚ PRODUCTO, SUB MENÚ PLATOS

MENU PRINCIPAL

**PRODUCTO** VENTAS CAJA MANTENIMIENTO

CONFIGURACION

**PLATOS**

REGISTRO DE PLATOS

GUARDAR ELIMINAR SALIR

PRODUCTO IMAGEN

CATEGORIA: PLATOS A LA CARTA PRODUCTO: PACHAMANCA

COCINA: COCINA1 PRECIO VENTA: 0

00212

PRODUCTO	COCINA	TIPO	PRECIO
PACHAMANCA	COCINA1	PLATOS TIPICO	24
PICANTE DE CUY 1/2	COCINA1	PLATOS TIPICO	24
CUY FRITO 1/2	COCINA1	PLATOS TIPICO	24
CUY FRITO ENTERO	COCINA1	PLATOS TIPICO	48
LOCHO DE CUY 1/2	COCINA1	PLATOS TIPICO	24
LOCHO DE GALLINA	COCINA1	PLATOS TIPICO	24
ESCABECHE DE POLLO 1/4	COCINA1	PLATOS A LA C	21
POLLO DORADO 1/4	COCINA1	PLATOS A LA C	21
PICANTE DE POLLO 1/4	COCINA1	PLATOS TIPICO	21
FRITANGUITA	COCINA1	PLATOS TIPICO	13
CALDO SOLO	COCINA1	PLATOS TIPICO	5
PORCION ARROZ	COCINA1	GUARNICION	4
PORCION DE PAPA	COCINA1	GUARNICION	4
ZARSA DE PICANTE DE CUY CON PAPA	COCINA1	PLATOS TIPICO	7
CHICHA MORADA JARRA	FUENTESODA	BEBIDAS	8
CHICHA MORADA 1/2 JARRA	FUENTESODA	BEBIDAS	4
CHICHA MORADA VASO	FUENTESODA	BEBIDAS	3
JORA 1/2 JARRA	FUENTESODA	BEBIDAS	4
JORA JARRA	FUENTESODA	BEBIDAS	8
JORA VASO	FUENTESODA	BEBIDAS	3
COCONA JARRA	FUENTESODA	BEBIDAS	8
COCONA 1/2 JARRA	FUENTESODA	BEBIDAS	4
COCONA VASO	FUENTESODA	BEBIDAS	3
ARROZ CON PATO	COCINA2	PLATOS TIPICO	24
PATO ASADO	COCINA2	PLATOS TIPICO	24
LECHON AL HORNO	COCINA2	PLATOS TIPICO	24
BISTECK A LO POBRE	COCINA2	PLATOS A LA C	24
LOMO SALTADO	COCINA2	PLATOS A LA C	24
CHORIZO CON TACHO	COCINA2	PLATOS A LA C	24
TRUCHA FRITA	COCINA2	PLATOS TIPICO	22
JUANES DE GALLINA	COCINA2	PLATOS TIPICO	22
CEVICHE DE PATO	COCINA2	PLATOS TIPICO	23
PECHUGA A LA PLANCHA	COCINA2	PLATOS A LA C	21
MILANESA DE POLLO	COCINA2	PLATOS A LA C	21
ARROZ A LA CUBANA	COCINA2	PLATOS A LA C	9
POLLO CANGA	COCINA2	PLATOS A LA C	21
PORCION DE PLATANO	COCINA2	GUARNICION	4
PORCION DE TACHO	COCINA2	GUARNICION	4
PORCION DE PATACONES	COCINA2	GUARNICION	4
PORCION DE MOTE	COCINA2	GUARNICION	4
PORCION YUCAS	COCINA2	GUARNICION	4
CAU CAU	COCINA2	DESAYUNOS	8
PAPAYA - VASO	FUENTESODA	JUGOS	4
CHAUFA DE POLLO	COCINA2	DESAYUNOS	8
LOMITO AL JUGO	COCINA2	DESAYUNOS	10
CALDO DE GALLINA	COCINA2	DESAYUNOS	12
PICANTE DE CUY 1/4	COCINA2	DESAYUNOS	13
CECINA CON MOTE Y PLATANO 1/2	COCINA2	DESAYUNOS	12
JUANE DE POLLO	COCINA2	DESAYUNOS	12
JUANE DE POLLO	COCINA2	DESAYUNOS	12
CHICHARRON CON MOTE Y PLATANO	COCINA2	PLATOS TIPICO	22
POLLO FRITO	COCINA2	DESAYUNOS	10
CHORIZO CON MOTE Y PLATANO 1/2	COCINA2	DESAYUNOS	12
ARANADO DE REC	COCINA2	DESAYUNOS	12
MIXTO (CESINA TACHO MOTE PLATANO)	COCINA2	PLATOS TIPICO	35
MIXTO (CHORIZO TACHO MOTE PLATANO)	COCINA2	PLATOS TIPICO	35
LIMONADA FROZEN	COCINA2	BEBIDAS	8
LIMONADA 1/2 JARRA	COCINA2	BEBIDAS	4
DELICIA DE DURAZNO	FUENTESODA	POSTRES	6
TAZA MANZANILLA	FUENTESODA	INFUSIONES	2,5
SERVICIO DE MESA	FUENTESODA	GENERAL	5
TAZA TE	FUENTESODA	INFUSIONES	2,5
SALSA PICANTE DE CUY	COCINA1	PLATOS TIPICO	10
ENSALADA FRESCA	COCINA2	GUARNICION	5
PORCION CAMOTE ASADO	COCINA2	PLATOS A LA C	2
PATITA CON MANI	COCINA2	DESAYUNOS	8
PORCION PLATANO ASADO	COCINA2	GUARNICION	2
CAFE CON LECHE	COCINA2	DESAYUNOS	3
MILANESA DE POLLO 1/2	COCINA2	DESAYUNOS	12
MOUSSE DE GUANABANA	FUENTESODA	POSTRES	5
MOUSSE DE GUANABANA	FUENTESODA	POSTRES	5
SANDWICH CHICHARRON	COCINA2	DESAYUNOS	8

Este sub menu nos permite crear Los platos de la carta, esta opción se puede registrar, editar o eliminar los menús de la carta.

## MENÚ PRODUCTO, SUB MENÚ PRODUCTOS

REGISTRO DEL PRODUCTO

GUARDAR ELIMINAR

PRODUCTO PRECIOS E IMAGENES

CATEGORIA  CODIGO BARRAS  STOCK  0

PRODUCTO  STOCK MINIMO  0

COSTO  0 % DE UTILIDAD  0 % PRECIO VENTA  0

00212

RA	PRODUCTO	TIPO	PRECIO	STOCK	STOCKMIN
	TABERNERO GRAN BLANCO SEMI SECO	LICORES	28	1	0
	TABERNERO CABERNET SAUVIGNON TINTO SECO	LICORES	35	3	0
	SHAMPAGNE VIVACE ESPUMANTE	LICORES	0	1	0
	TABERNERO PISCO PREMIUM MOSTO VERDE	LICORES	80	1	0
	GRAN FINCA PISCO ROTONDO ACHOLADO	LICORES	40	1	0
	PISCO VARGAS	LICORES	35	6	0
	PISCO VARGAS	LICORES	35	1	0
	PISCO VARGAS	LICORES	35	1	0
	SANTIAGO QUEIROLO SAUVIGNON BLANCO	LICORES	28	1	0
	JARABE DE GOMA	LICORES	11	2	0
	GRANADIVA	LICORES	0	1	0
	QUEIROLO BORGOÑA TINTO DULCE	LICORES	25	0	0
	QUEIROLO MAGDALENA TINTO SEMI SECO	LICORES	25	18	0
	QUEIROLO VINO ROSE SHIRAZ	LICORES	25	6	0
	QUEIROLO GRAN VINO BORGOÑA	LICORES	25	3	0
	SPORADE ROJA	BEBIDAS	3,5	2	0

Este sub menú nos permite crear Los productos de venta, esta opción se puede registrar o eliminar los productos de venta.

## MENÚ PRODUCTO, SUB MENÚ ALMACEN

MENU PRINCIPAL

PRODUCTO VENTAS CAJA MANTENIMIENTO

CONFIGURACION PLATOS PRODUCTOS ALMACEN REPORTES

INGRESO ALMACEN

COMPROBANTE  GUIA  000002  FECHA 04/04/2016

ALMACEN  PRINCIPAL  MOTIVO  INGRESO

PROVEEDOR  1045329050  PROVEEDOR 1

DIRECCION  URBA LAS FLORES LIMA

PRODUCTO  CODIGO BARRA

PRODUCTO  INCA 2 LITROS CANTIDAD  34 AGREGAR

CODIGO PRODUCTO CANTIDAD

¿ESTA SEGURO DE GRABAR LA OPERACION?

SI No

OBSERVACION  INCA COLA DE 2 LITROS

LIMPIAR GUARDAR

Este sub menú nos permite crear Ingresar los productos al almacén.

## MENÚ PRODUCTO, SUB MENÚ ALMACEN

MENU PRINCIPAL  
PRODUCTO VENTAS CAJA MANTENIMIENTO

CONFIGURACION ▶  
PLATOS  
PRODUCTOS  
ALMACEN ▶  
REPORTES ▶

INGRESO ALMACEN  
SALIDA ALMACEN

SALIR

COMPROBANTE ALMACEN: NOTA DE SALIDA 0001 000014  
FECHA: 02/02/2016  
MOTIVO: MALOGRADO

CLIENTE: 10283452343 mspartida ortega Victor  
DIRECCION: Jr. los Incas numero 145 paucarbamba, amarilis

PRODUCTO: carne CANTIDAD: 50 AGREGAR

ERP  
¿ESTA SEGURO DE GRABAR LA OPERACION?  
SI NO

OBSERVACION: dañado  
LIMPIAR GUARDAR

Este sub menu nos permite crear Las salidas de los productos de consumo ya sea que está dañado, malogrado, devolución, etc.

## MENÚ PRODUCTO, SUB MENÚ REPORTES (ingreso y salida del almacen)

MENU PRINCIPAL  
PRODUCTO VENTAS CAJA MANTENIMIENTO

CONFIGURACION ▶  
PLATOS  
PRODUCTOS  
ALMACEN ▶  
REPORTES ▶

REPOSICION DE STOCK  
INGRESO ALMACEN  
SALIDA ALMACEN  
DETALLE INGRESO ALMACEN  
DETALLE SALIDA ALMACEN  
KARDEX

SALIR

FECHA\_INICIO: 01/02/2016 FECHA\_FINAL: 31/04/2016 BUSCAR

SAP CENTRAL REPORTER

SERIE	NUMERO	FECHA	P. DE PARTIDA	P. DE LLEGADA	MOTIVO
-------	--------	-------	---------------	---------------	--------

Este sub menu nos permite crear el reporte de los productos ya sea de ingreso y salida.

## MENÚ PRODUCTO, SUB MENÚ REPORTES (detalle ingreso al almacén y detalle salida almacén)

FECHA\_INICIO 01/01/2016 FECHA\_FINAL 18/04/2016 BUSCAR SALIR

FECHA	COMP	SERIE	NUMERO	PRODUCTO	CANT
18/01/2016	NOTA DE INGR	1	000026	PACHAMANGA	5.00
19/01/2016	NOTA DE INGR	1	000037	PACHAMANGA	5.00
19/01/2016	NOTA DE INGR	1	000037	FICARTE DE CUY 1/2	5.00
19/01/2016	NOTA DE INGR	1	000038	CUY PRITO 1/2	5.00

Esta opción nos permite crear el reporte de los productos detallados ingresando una fecha de inicio y final para poder visualizar el rango de tiempo

## MENÚ PRODUCTO, SUB MENÚ REPORTES (Kardex)

Esta opción nos permite crear el reporte de los productos detallados ingresando una fecha de inicio y final para poder visualizar el rango de tiempo

FECHA	COMP	NUMERO	PRODUCTO	OPERACION	EGRESO	INGRESO	STOCKACTUAL
13/02/2016	TICKET	0000093	QVY FRITO 1/2	Venta	1	0	1
13/02/2016	TICKET	0000094	QVY FRITO 1/2	Venta	2	0	3
09/03/2016	TICKET	0000049	LOCO DE GALLINA	Venta	9	0	9
04/04/2016	TICKET	0000092	PAGRAMANSA	ANULACION COMPROM	0	1	1

## MENÚ VENTAS, SUB MENÚ CLIENTES

Esta opción nos permite registrar, editar y eliminar a los clientes de la empresa

RUC	CLIENTE	TIPOCLIENTE	DIRECCION	DEPARTAMENT	PROVINCIA	DISTRITO	EMAIL	CELULAR	WHATSAPP	FONO
213123	rsdsdad	USTARIO PNA	rsdsdad				@	0	0	0
54545454	jjjjjjjjjjjjjjjjjjjj	USTARIO PNA	jjjjjjjjjjjjjjjjjjjj				@	0	0	0
10450032225	PAGANO RAUSTINO GUILLERMO	USTARIO PNA	RR. MANCO CAPACANBO.				@	0	0	0



## MENÚ VENTAS, SUB MENÚ VENTA.

Esta opción nos permite registrar la venta del producto con relación al número de mesa

The screenshot displays the 'SUB MENÚ VENTA' interface. On the left, a sidebar menu is open with 'VENTA' highlighted. The main area shows a product list with columns for 'PRODUCTO', 'CANT', 'PRECIO', 'IMPORTE', 'DESC', 'TOTAL', and 'COCINA'. The right-hand panel contains a summary of the sale with fields for 'IMPORTE', 'DESC', 'SUBTOTAL', 'IGV', 'TOTAL', 'EFFECTIVO', and 'VUELTO', along with buttons for 'LIMPIAR', 'GUARDAR', 'IMPRIMIR', and 'CLIENTE'.

PRODUCTO	CANT	PRECIO	IMPORTE	DESC	TOTAL	COCINA
PACHAMANCA	2	24,00	48,00	0,00	48,00	
CUT FRITO 1/2	1	24,00	24,00	0,00	24,00	
LOCO DE CUT 1/2	2	24,00	48,00	0,00	48,00	
POLLO DORADO 1/4	1	21,00	21,00	0,00	21,00	
JORA 1/2 BARRA	1	4,00	4,00	0,00	4,00	
MILANESA DE POLLO	1	21,00	21,00	0,00	21,00	
PORCIÓN TUCAS	1	4,00	4,00	0,00	4,00	
CUT FRITO EXTERNO	1	48,00	48,00	0,00	48,00	
LOCO DE GALLINA	1	24,00	24,00	0,00	24,00	

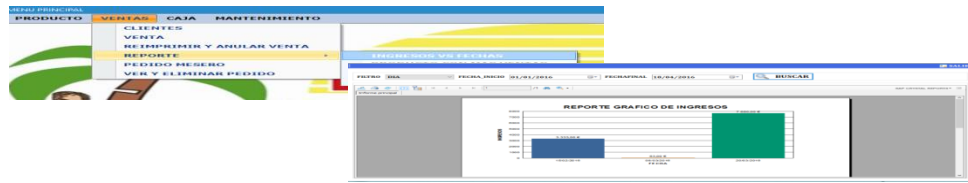
## MENÚ VENTAS, SUB MENÚ REIMPRIMIR Y ANULAR VENTA.

The screenshot displays the 'REIMPRIMIR Y ANULAR VENTA' interface. It features a search bar with 'COMPROBANTE', 'SERIE', 'NUMERO', 'FECHA INICIAL', and 'FECHA FINAL' fields. Below the search bar is a table of sales records with columns for 'COMPROBANTE', 'SERIE', 'NUMERO', 'FECHA INICIAL', 'FECHA FINAL', 'RUC O DNI', 'DIRECCION', 'IMPORTE', 'DESC', 'SUBTOTAL', 'IGV', and 'TOTAL'.

COMPROBANTE	SERIE	NUMERO	FECHA INICIAL	FECHA FINAL	RUC O DNI	DIRECCION	IMPORTE	DESC	SUBTOTAL	IGV	TOTAL
TICKET	0001	0000017	15/02/2016				3333	0	3333	0	3333
TICKET	0001	0000018	15/02/2016				0	0	0	0	0
TICKET	0001	0000019	08/03/2016				63	0	63	0	63
TICKET	0001	0000020	28/03/2016				3632	0	3632	0	3632
TICKET	0001	0000021	28/03/2016				3578	0	3578	0	3578
TICKET	0001	0000022	28/03/2016				400	0	400	0	400
TICKET	0001	0000023	04/04/2016				3333	0	3333	0	3333
<b>TOTAL</b>							<b>3333</b>	<b>0</b>	<b>3333</b>	<b>0</b>	<b>3333</b>

Esta opción nos permite volver a imprimir y eliminar la venta registrada anteriormente.

## MENÚ VENTAS, SUB MENÚ REPORTE (ingreso VS fechas).



Esta opción nos permite visualizar los ingresos de una manera gráfica donde se indica la fecha de inicio y final.

## MENÚ VENTAS, SUB MENÚ REPORTE (producto con más ventas)

Esta opción nos permite visualizar los productos con más ventas de una manera gráfica donde se indica la fecha de inicio y final.



## MENÚ VENTAS, SUB MENÚ REPORTE (lista de ventas)

The screenshot shows the SAP menu structure with 'VENTAS' selected. The 'REPORTE' option is highlighted. Below the menu, the SAP Crystal Reports interface is shown with the following data table:

COMP	SERIE	NUMERO	EMISION	RUC	CLIENTE	SUBTOTAL	IGV	TOTAL
TICKE0001	0000047	15/02/2016				3.333,00	0,00	3.333,00
TICKE0001	0000048	15/02/2016				0,00	0,00	0,00
TICKE0001	0000049	06/03/2016				63,00	0,00	63,00
TICKE0001	0000089	20/03/2016				3.612,00	0,00	3.612,00
TICKE0001	0000090	20/03/2016				3.578,00	0,00	3.578,00
TICKE0001	0000091	20/03/2016				490,00	0,00	490,00
TICKE0001	0000092	04/04/2016				3.333,00	0,00	3.333,00
<b>TOTAL</b>						<b>14.409,00</b>		

Esta opción nos permite visualizar la lista de ventas de los productos, donde se indica la fecha de inicio y final

## MENÚ VENTAS, SUB MENÚ REPORTE (lista detalle de ventas)

The screenshot shows the SAP menu structure with 'LISTA DETALLE VENTA' selected. Below the menu, the SAP Crystal Reports interface is shown with the following data table:

COMP	SERIE	NUMERO	FECHA	PRODUCTO	CANT	PRECIO	DESC	TOTAL
TICKET 0001	0000047	15/02/2016		PIZZERIA	1	3.333,00	0,00	3.333,00
TICKET 0001	0000048	15/02/2016		0- CUP PRETO ENTERO	1	0,00	0,00	0,00
TICKET 0001	0000049	06/03/2016		0- CUP PRETO GALIANA	1	63,00	0,00	63,00
TICKET 0001	0000089	20/03/2016		0- PIZAS DE CIV 1/3	1	3.612,00	0,00	3.612,00
TICKET 0001	0000090	20/03/2016		0- CUP PRETO CIV 1/2	1	3.578,00	0,00	3.578,00
TICKET 0001	0000091	20/03/2016		0- PIZAS DE CIV 1/2	1	490,00	0,00	490,00
TICKET 0001	0000092	04/04/2016		0- CUP PRETO ENTERO	1	3.333,00	0,00	3.333,00
TICKET 0001	0000091	20/03/2016		0- PIZAS DE CIV 1/2	2	245,00	0,00	490,00
TICKET 0001	0000091	20/03/2016		0- CUP PRETO ENTERO	1	0,00	0,00	0,00
<b>TOTAL</b>								<b>11.076,00</b>

Esta opción nos permite visualizar la lista de ventas de los productos, donde se indica la fecha de inicio y final

## MENÚ VENTAS, SUB MENÚ PEDIDO MESERO

Este módulo de pedidos permite registrar los productos para la venta de acuerdo al número de mesa y también permitirá incrementar más pedidos en dichas mesas

The screenshot shows the 'PEDIDO MESERO' interface. At the top, there are tabs for 'PRODUCTO', 'VENTAS', 'CAJA', and 'MANTENIMIENTO'. Below these, there are options: 'CLIENTES', 'VENTA', 'REIMPRIMIR Y ANULAR VENTA', 'REPORTE', and 'PEDIDO MESERO'. The main area is titled 'GESTION DE MESAS' and shows a grid of 14 tables labeled 'MESA 1' through 'MESA 14'. Below the grid, there is a form for 'PEDIDO MESERO' with fields for 'MESA', 'MESERO(A)', 'CLIENTE', 'OBS. PLATO', and 'COCINA1'. A total amount of '168,00' is displayed. To the right, there is a list of 'PLATOS' with their respective prices.

PRODUCTO	CAN	PRECI	IMPORT	C
PACHAMANCA	1	24,00	24,00	C
PICANTE DE CUY 1/2	1	24,00	24,00	C
CUY FRITO 1/2	1	24,00	24,00	C
CUY FRITO ENTERO	1	48,00	48,00	C
LOCRO DE CUY 1/2	1	24,00	24,00	C
LOCRO DE GALLINA	1	24,00	24,00	C

PLATOS	PRECIO
PACHAMANCA	24
PICANTE DE CUY 1/2	24
CUY FRITO 1/2	24
CUY FRITO ENTERO	48
LOCRO DE CUY 1/2	24
LOCRO DE GALLINA	24
ESCABECHE DE POLLO 1/4	21
POLLO DORADO 1/4	21
PICANTE DE POLLO 1/4	21

## MENÚ VENTAS, SUB MENÚ VER Y ELIMINAR PEDIDO

The screenshot shows the 'VER Y ELIMINAR PEDIDO' interface. At the top, there are tabs for 'MENU PRINCIPAL', 'PRODUCTO', 'VENTAS', 'CAJA', and 'MANTENIMIENTO'. Below these, there are options: 'CLIENTES', 'VENTA', 'REIMPRIMIR Y ANULAR VENTA', 'REPORTE', 'PEDIDO MESERO', and 'VER Y ELIMINAR PEDIDO'. The main area is titled 'VISUALIZAR Y ELIMINAR PEDIDO' and shows a search form with fields for 'COMPONENTE', 'PEDIDO', 'NUMERO', 'OBS', 'FECHA INICIAL', 'FECHA FINAL', 'FECHA INICIAL', 'FECHA FINAL', and a 'BUSCAR' button.

Este módulo de pedidos permite visualizar los pedidos y luego proceder a eliminarlo de acuerdo al número de mesa y las fechas de inicio y final

## MENÚ CAJA, SUB MENÚ CIERRE DE CAJA

**CIERRE DE CAJA**

FECHA: 16/05/2016  
 HORA: 18:17:27

EFFECTIVO: 90  
 CAJA CHICA: 0  
 INGRESOS: 1468  
 EGRESOS: 0  
 SALDO: 1558  
 T. TARJETA: 0  
 TOTAL: 1558

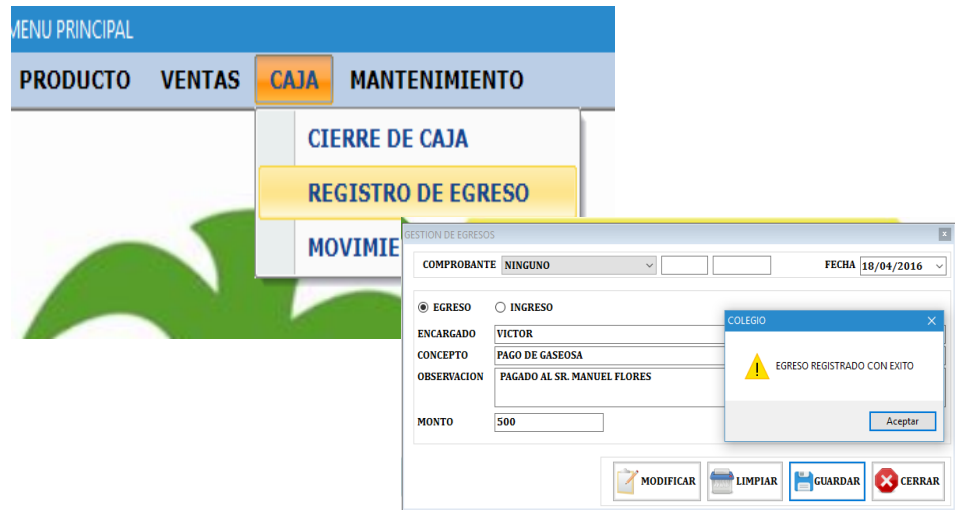
FECHAS	CONCEPTOS	SERIE	USUARIO	RUC	CLIENTE	SUBTOTAL	TOTAL
16/05/2016	TICKET	0001	0000004			242	242
16/05/2016	TICKET	0001	0000006			96	96
16/05/2016	TICKET	0001	0000006			56	56
16/05/2016	TICKET	0001	0000007			14	14
16/05/2016	TICKET	0001	0000008			164	164
16/05/2016	TICKET	0001	0000009			256	256
16/05/2016	TICKET	0001	0000100			256	256

RETIRO: \_\_\_\_\_  
 PARA CASH CHICA: \_\_\_\_\_  
 ENCARGADO: \_\_\_\_\_  
 OBSERVACION: \_\_\_\_\_

Esta opción de caja nos permite hacer el cierre de caja para poder saber cuánto es el ingreso del día.

## MENÚ CAJA, SUB MENÚ REGISTRO DE EGRESO

Esta opción de registro de egreso nos permite hacer el ingreso de los gastos ocurridos durante el día



## MENÚ CAJA, SUB MENÚ MOVIMIENTO DE CAJA



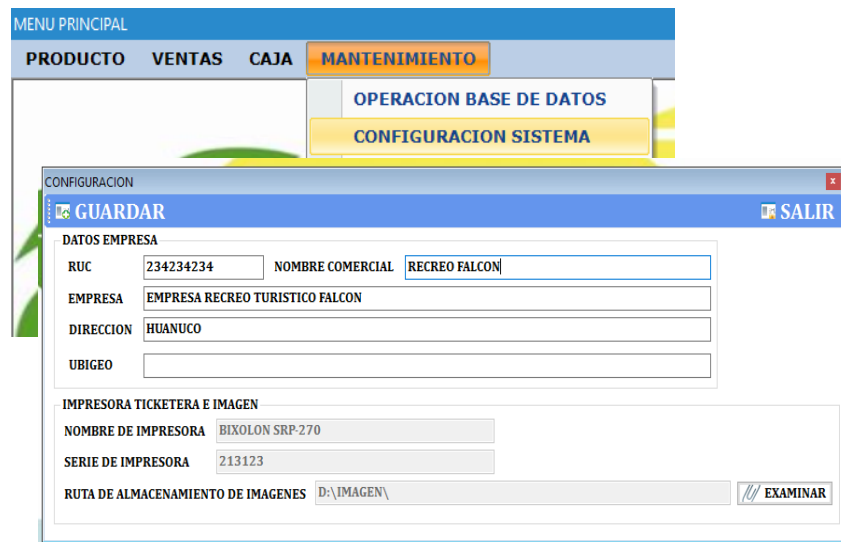
## MENÚ MANTENIMIENTO, SUB MENÚ OPERACIÓN BASE DE DATOS

Esta opción nos permite hacer las respectivas reparaciones, optimizar y crear copia de respaldo de la base de datos.



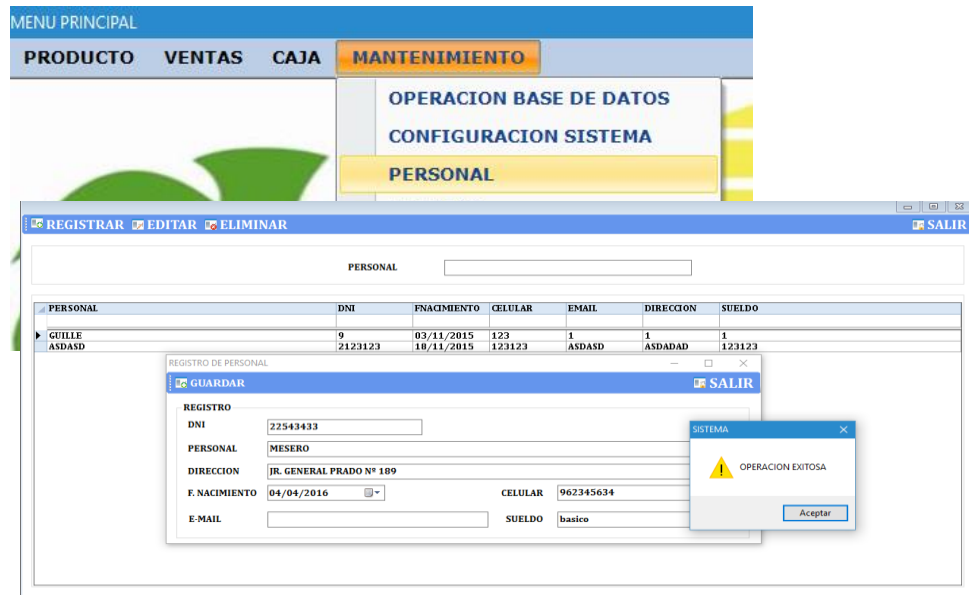
## MENÚ MANTENIMIENTO, SUB MENÚ CONFIGURACION SISTEMA

Esta opción nos permite hacer los respectivos cambios de nombre del sistema



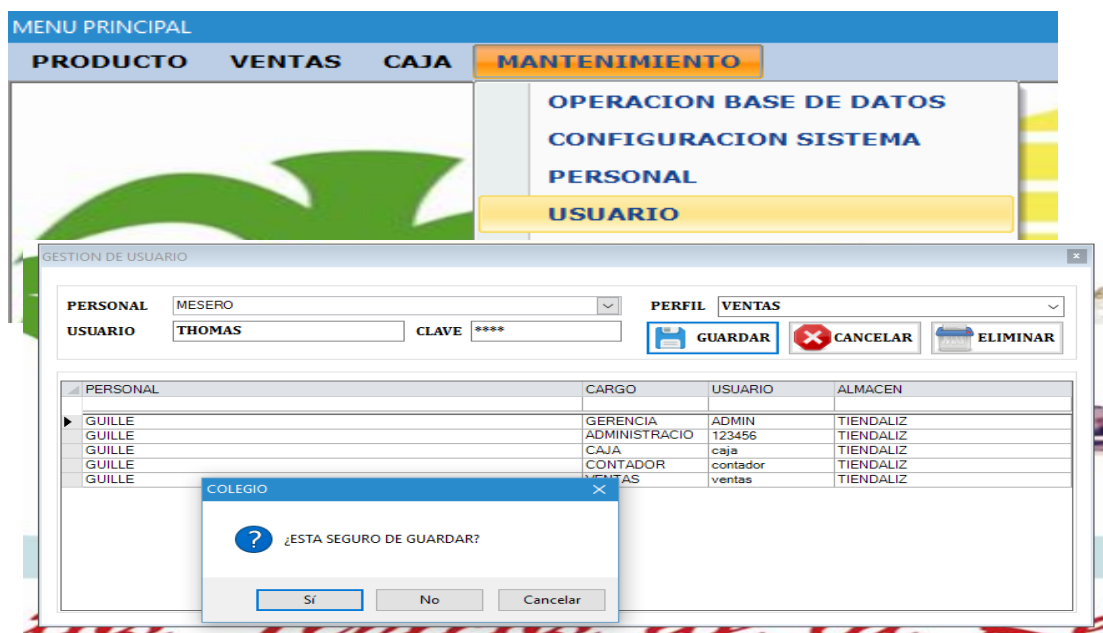
## MENÚ MANTENIMIENTO, SUB MENÚ PERSONAL

Esta opción nos permite registrar, editar, eliminar al personal que labora en la empresa.



## MENÚ MANTENIMIENTO, SUB MENÚ USUARIO

Esta opción nos permite guardar, eliminar al usuario que tiene acceso al sistema de la empresa.





## MENÚ MANTENIMIENTO, SUB MENÚ INICIALIZAR CAJA



Esta opción nos permite inicializar la caja, ingresando al usuario y la fecha del día.

## MENÚ MANTENIMIENTO, SUB MENÚ ADSOFT

The image shows two overlapping windows. The top window is the 'MENU PRINCIPAL' with the 'MANTENIMIENTO' tab active, showing the 'ADSOFT' option highlighted in yellow. The bottom window is titled 'EXPORTAR A EXCEL' and contains a search bar with 'FECHA INICIAL' (08/02/2016) and 'FECHA FINAL' (18/04/2016), and a 'BUSCAR' button. Below the search bar is a table with the following data:

SERIE	NUMERO	FECEAC	FECCVEN	NRO_RUC	NOMBRE	TIPDOC	TIPMON	DETRAC	IMPVTA	IMPINA	REGARC	IMPVIG	IMPOT	ST	SER_DQ	NUM_DQM
1	37	15/02/2016	15/02/2016			12	SOLES		3324		0	0	0			
1	40	15/02/2016	15/02/2016			12	SOLES		0		0	0	0			
1	49	08/03/2016	08/03/2016			12	SOLES		63		0	63				
1	89	20/03/2016	20/03/2016			12	SOLES		3612		0	3612				
1	90	20/03/2016	20/03/2016			12	SOLES		3578		0	3578				
1	91	20/03/2016	20/03/2016			12	SOLES		490		0	490				
1	92	04/04/2016	04/04/2016		ANULADO	12	SOLES		0		0	0	0	A		

Esta opción nos permite generar un reporte en Excel con las especificaciones correspondientes del contador ya que ellos usan el software ADSOFT

## **CONCLUSIONES**

El sistema yaudo a mejorar los procesos de:

- Con la implementación del sistema informático se puedo evidenciar la mejora en las ventas pues la atención era más rápida y sin errores en los pedidos.
- Con la implementación del sistema informático se puedo evidenciar la mejora en la gestión caja ya que saber que se gastó y que se vendió toma el menor tiempo posible.
- Con la implementación del sistema informático se puedo evidenciar la mejora en gestión de almacén pues se sabe con exactitud qué productos existen o no en el almacén.
- Ya que se mejoró en la gestión de ventas, caja y almacén es más que evidente que se mejoró la calidad en el servicio en el RECREO FALCON.

## **RECOMENDACIONES**

- Con respecto a la atención al cliente, ya que el sistema ha mejorado la velocidad de atención con el cliente se le recomienda al personal una pequeña capacitación de atención para así apoyar a que el sistema cumpla la función para la que ha sido creada.
- Se alienta realizar un mantenimiento preventivo y correctivo del sistema de manera periódica para evitar un mal funcionamiento.
- Se recomienda la implementación a futuro de un módulo de facturación electrónica.

## REFERENCIAS BIBLIOGRÁFICAS

- Bertalanffy, L. V. (1976). Teoría General de los Sistemas. México: Fondo de Cultura Económica.
- Davis, Gordon M y Olson, Margrethe H. (1994). Sistemas de Información Gerencial. McGraw-Hill.
- ITSA. (2008). Metodologías De Desarrollo De Software. Canada: Canada Pen.
- Márquez Redhead, J. (2013). Implementación de un sistema de información que apoye el proceso diario de elaboración de cronogramas del personal de medicina física y rehabilitación de un hospital (tesis de Grado). Universidad Pontificia Universidad Católica Del Perú.
- Sánchez Mercado, A. (2011). Análisis y diseño de un sistema informatizado para la Dinamización de los procesos y procedimientos practicados en la atención médico hospitalaria de los pacientes de oncología de un hospital público (tesis de Grado). Universidad Pontificia Universidad Católica Del Perú.
- Sánchez Carlessi H. y Reyes Meza C. (2006). Metodología y diseños en investigación científica. Edit. Visión Universitaria. Lima – Perú.

## ANEXOS

### MATRIZ DE CONSISTENCIA

PROBLEMAS	OBJETIVOS	VARIABLES	INDICADORES	TIPO DE INVESTIGACIÓN
¿ De que manera implementar un sistema informático de caja y facturación para el RECREO FALCON?	Implementar un sistema informático de caja y facturación para el RECREO FALCON	VARIABLE INDEPENDIENTE Sistema informático de caja y facturación.  VARIABLE DEPENDIENTE Gestión del RECREO FALCON.	<ul style="list-style-type: none"> <li>• Programación funcional</li> <li>• Interfaz adecuada</li> <li>• Tiempo de atención</li> <li>• Tiempo de búsqueda</li> <li>• Control de accesos</li> <li>• Respaldo de datos(Backup)</li>   <li>• Tiempo de respuesta de registros</li> <li>• Tiempo de respuesta de búsquedas</li> </ul>	Aplicada o Tecnológica
PROBLEMAS ESPECIFICOS	OBJETIVOS ESPECIFICOS			
¿Cómo implementar un Subsistema de gestión de caja en el RECREO FALCON?  ¿Cómo implementar un Subsistema de gestión de facturación en el RECREO FALCON?	Implementar un Subsistema de gestión de caja en el RECREO FALCON  Implementar un Subsistema de gestión de facturación en el RECREO FALCON			