

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR - JOSÉ MIGUEL CARRERA**

**SISTEMA INFORMÁTICO WEB PARA NEGOCIO
CON GIRO DE LIBRERÍA, PERFUMERÍA Y ARTÍCULOS DE PAQUETERÍA**

Trabajo de Titulación para optar al Título de
Técnico Universitario en INFORMÁTICA

Alumna:

María José Fajardo Vásquez

Profesora Guía:

Lais San Martín Navarro

2017

A Inés, Pedro, María de los Ángeles y Fernando, gracias por el apoyo incondicional.

A mi tía favorita Giannina ♥, mensaje recibido, nosotras también te amaremos por siempre.

"Brilla el sol, es un nuevo amanecer" – Wacry, Alma de Conquistador.

RESUMEN

KEYWORDS: Sistema informático, Ventas, Control de Existencias.

El presente sistema informático web fue desarrollado para Maybet, un negocio con giro comercial de librería, perfumería y artículos de paquetería.

El objetivo principal del sistema es permitir un control de los datos con que se maneja en el negocio, como lo es el nivel de existencias, el flujo de caja, control de las ventas, las devoluciones por parte del cliente, como así también, un registro de los proveedores, facturas de compra y empleados del local.

El sistema está implementado bajo la plataforma de tipo Web, usando PHP como lenguaje de programación, Apache como servidor y MySQL como motor de la base de datos. El sistema además, está orientado a que pueda crecer en el tiempo, dejando abierta la posibilidad de agregar nuevas funcionalidades.

El Trabajo de Título consta de tres capítulos. El primero de ellos se enfoca en los aspectos que son relevantes para el diseño lógico del sistema, en el que se describe la situación actual y los problemas que son detectados al momento de presentar el proyecto, para luego definir la solución planteada y orientada al local comercial.

En el segundo capítulo se describe el medio ambiente computacional en el que se realizó el sistema, es decir, la descripción del software y hardware utilizado. Posteriormente se presenta una descripción de archivos del sistema y las tablas utilizadas para la base de datos.

El tercer capítulo corresponde a la descripción de programas, con una breve descripción, su funcionalidad y diseño de interfaz.

Finalmente se exponen las conclusiones, las fuentes bibliográficas utilizadas a lo largo del desarrollo del proyecto y el anexo, que contiene el código completo del sistema.

ÍNDICE

RESUMEN

INTRODUCCIÓN	1
1 ASPECTOS RELEVANTES DEL DISEÑO LÓGICO	5
1.1 DESCRIPCIÓN DE LA ORGANIZACIÓN	5
1.2 DESCRIPCIÓN DE SITUACIÓN ACTUAL	6
1.2.1 Aspectos generales	6
1.2.2 Proceso de ventas	6
1.2.3 Proceso de compra al proveedor.....	6
1.2.4 Control de precios de los productos	7
1.2.5 Control de <i>stock</i>	7
1.2.6 Control de devoluciones	7
1.3 PROBLEMAS DETECTADOS.....	8
1.3.1 Proceso de ventas	8
1.3.2 Proceso de compra al proveedor.....	8
1.3.3 Control de precios de los productos	8
1.3.4 Control de <i>stock</i>	9
1.3.5 Control de devoluciones	9
1.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO.....	9
1.4.1 Objetivos y beneficios del sistema propuesto	9
1.4.1.1 Objetivo Principal	9
1.4.1.2 Objetivos Específicos.....	10
1.4.1.3 Beneficios del sistema propuesto	10
1.4.2 Descripción general de la situación propuesta	10
1.4.3 Diagramas de flujos administrativos.....	12
1.4.4 Estructura funcional del sistema	14
1.4.5 Descripción de fórmulas y cálculos empleados.....	15
1.4.6 Entradas y salidas	16
1.4.6.1 Entradas.....	16
1.4.6.2 Salidas.....	17
1.4.7 Entidades de información	18
1.4.7.1 Descripción de entidades.....	18
1.4.7.2 Modelo de datos relacional.....	20
1.4.8 Estructura de códigos.....	21
1.4.9 Condicionantes de diseño	21
2 MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE	
ARCHIVOS	25
2.1 DESCRIPCIÓN DE LAS CARACTERÍSTICAS DEL RECURSO	
COMPUTACIONAL.....	25

2.1.1	Configuración del sistema.....	25
2.1.2	Software utilizado.....	25
2.2	DESCRIPCIÓN DE ARCHIVOS.....	26
2.2.1	Listado de archivos.....	26
2.2.2	Descripción de tablas.....	27
2.2.2.1	Tabla Producto	27
2.2.2.2	Tabla Categoría.....	28
2.2.2.3	Tabla Subcategoría	28
2.2.2.4	Tabla Proveedor	28
2.2.2.5	Tabla Prod_prov	29
2.2.2.6	Tabla Ajustes	30
2.2.2.7	Tabla Usuario.....	30
2.2.2.8	Tabla Venta	31
2.2.2.9	Tabla Detalle_venta	31
2.2.2.10	Tabla Caja	32
2.2.2.11	Tabla Usuario_caja	33
2.2.2.12	Tabla Giros	33
2.2.2.13	Tabla Devoluciones	34
2.2.2.14	Tabla Pedido	34
2.2.2.15	Tabla Detalle_pedido.....	35
2.2.2.16	Tabla Factura	36
2.2.2.17	Tabla Detalle_factura	36
3	DESCRIPCIÓN DE PROGRAMAS	41
3.1	DIAGRAMA DE MENÚS	41
3.1.1	Menú administrativo	41
3.1.2	Menú vendedor	42
3.2	DIAGRAMA MODULAR	43
3.3	TABLA DE PROGRAMAS DEL SISTEMA	44
3.4	DESCRIPCIÓN DETALLADA DE PROGRAMAS.....	47
3.4.1	Identificación del usuario.....	47
3.4.2	Inicio	48
3.4.3	Mantenedor de productos	50
3.4.3.1	Ingresar producto.....	50
3.4.3.2	Listar productos	52
3.4.3.3	Modificar productos	56
3.4.3.4	Eliminar productos.....	58
3.4.4	Mantenedor de subcategorías	58
3.4.4.1	Ingresar subcategorías.....	58
3.4.4.2	Listar subcategorías.....	59
3.4.4.3	Modificar subcategorías.....	60
3.4.4.4	Eliminar subcategorías	61

3.4.5	Mantenedor de usuarios	62
3.4.5.1	Ingresar usuarios	62
3.4.5.2	Listar usuarios	63
3.4.5.3	Modificar usuarios.....	64
3.4.5.4	Eliminar usuarios.....	65
3.4.6	Mantenedor ajustes de existencias.....	66
3.4.6.1	Ingresar ajustes de existencias	67
3.4.6.2	Listar ajustes de existencias	68
3.4.6.3	Eliminar ajustes de existencias.....	69
3.4.7	Mantenedor de factura	70
3.4.7.1	Ingresar facturas.....	70
3.4.7.2	Listar facturas.....	71
3.4.7.3	Modificar facturas	75
3.4.7.4	Eliminar facturas	76
3.4.8	Ventas	70
3.4.8.1	Ingresar ventas.....	77
3.4.8.2	Listar ventas.....	79
3.4.9	Caja.....	81
3.4.9.1	Cierre de caja	81
3.4.9.2	Historial de movimiento de caja.....	82
	CONCLUSIONES	85
	BIBLIOGRAFÍA.....	87
	ANEXOS	89

ÍNDICE DE DIAGRAMAS

Diagrama 1-1: Diagrama de flujo administrativo del proceso de ventas	12
Diagrama 1-2: Diagrama de flujo administrativo de compra al proveedor.....	13
Diagrama 3-1: Diagrama de menú administrativo	41
Diagrama 3-2: Diagrama de menú de vendedor.....	42
Diagrama 3-3: Diagrama modular del sistema.....	43
Diagrama 3-4: Diagrama identificación del usuario	47
Diagrama 3-5: Diagrama inicio	48
Diagrama 3-6: Diagrama de bloque ingresar producto.....	51
Diagrama 3-7: Diagrama de bloque listar producto	53
Diagrama 3-8: Diagrama de bloque ver proveedores de producto	54
Diagrama 3-9: Diagrama de bloque eliminar proveedores de producto	55

Diagrama 3-10: Diagrama de bloque ver productos con bajo stock	56
Diagrama 3-11: Diagrama de bloque modificar producto	57
Diagrama 3-12: Diagrama de bloque eliminar producto	58
Diagrama 3-13: Diagrama de bloque ingresar subcategoría	59
Diagrama 3-14: Diagrama de bloque listar subcategorías	60
Diagrama 3-15: Diagrama de bloque modificar subcategoría.....	61
Diagrama 3-16: Diagrama de bloque eliminar categoría	62
Diagrama 3-17: Diagrama de bloque ingresar usuario.....	63
Diagrama 3-18: Diagrama de bloque listar usuarios	64
Diagrama 3-19: Diagrama de bloque modificar usuario	65
Diagrama 3-20: Diagrama de bloque eliminar usuario	66
Diagrama 3-21: Diagrama de bloque ingresar ajustes de existencias	67
Diagrama 3-22: Diagrama de bloque listar ajustes de existencias.....	68
Diagrama 3-23: Diagrama de bloque eliminar ajustes de existencia	69
Diagrama 3-24: Diagrama de bloque ingresar factura	70
Diagrama 3-25: Diagrama de bloque listar factura.....	72
Diagrama 3-26: Diagrama de bloque ver detalle de factura.....	73
Diagrama 3-27: Diagrama de bloque modificar detalle de una factura	74
Diagrama 3-28: Diagrama de bloque eliminar detalle de factura	75
Diagrama 3-29: Diagrama de bloque modificar factura	75
Diagrama 3-30: Diagrama de bloque eliminar factura.....	77
Diagrama 3-31: Diagrama de bloque ingresar venta	77
Diagrama 3-32: Diagrama de bloque listar venta	80
Diagrama 3-33: Diagrama de bloque ver detalle de venta.....	81
Diagrama 3-34: Diagrama cierre de caja	82
Diagrama 3-35: Diagrama historial de movimientos de caja	83

ÍNDICE DE FIGURAS

Figura 1-1: Modelo relacional del sistema.	20
Figura 3-1: Pantalla identificación del usuario.....	48
Figura 3-2: Pantalla inicio menú administrativo	48
Figura 3-3: Pantalla inicio menú vendedor	50
Figura 3-4: Pantalla número uno de ingreso de un producto.....	52

Figura 3-5: Pantalla número dos de ingreso de un producto	52
Figura 3-6: Pantalla de lista y administración de productos	53
Figura 3-7: Pantalla de proveedores de cada producto.....	54
Figura 3-8: Pantalla de informe de <i>stock</i> crítico	56
Figura 3-9: Pantalla de modificación de un producto.....	57
Figura 3-10: Pantalla de ingreso de una subcategoría.....	59
Figura 3-11: Pantalla de lista y administración de una subcategoría	60
Figura 3-12: Pantalla de modificación de una subcategoría.....	61
Figura 3-13: Pantalla de ingreso de un usuario	63
Figura 3-14: Pantalla de lista y administración de usuarios.....	64
Figura 3-15: Pantalla de modificación de usuarios	65
Figura 3-16: Pantalla de ingreso de ajustes de existencias	68
Figura 3-17: Pantalla de lista y administración de ajustes de existencias.....	69
Figura 3-18: Pantalla número uno de ingreso de una factura	71
Figura 3-19: Pantalla número dos de ingreso de una factura	71
Figura 3-20: Pantalla de lista y administración de facturas	72
Figura 3-21: Pantalla de lista y administración de detalle de factura	73
Figura 3-22: Pantalla de modificación de detalle de factura	74
Figura 3-23: Pantalla de modificación de factura	76
Figura 3-24: Pantalla número uno de ingreso de una venta	78
Figura 3-25: Pantalla número dos de ingreso de una venta	79
Figura 3-26: Pantalla número tres de ingreso de una venta.....	79
Figura 3-27: Pantalla número cuatro de ingreso de una venta	79
Figura 3-28: Pantalla de lista de venta	80
Figura 3-29: Pantalla de lista de detalle de venta.....	81
Figura 3-30: Pantalla de cierre de caja	82
Figura 3-31: Pantalla de historial de movimiento de caja	83

ÍNDICE DE TABLAS

Tabla 2-1: Descripción Tabla Producto	27
Tabla 2-2: Descripción Tabla Categoría	28
Tabla 2-3: Descripción Tabla Subcategoría	28
Tabla 2-4: Descripción Tabla Proveedor.....	29
Tabla 2-5: Descripción Tabla Prod_prov	29

Tabla 2-6: Descripción Tabla Ajustes	30
Tabla 2-7: Descripción Tabla Usuario	30
Tabla 2-8: Descripción Tabla Venta.....	31
Tabla 2-9: Descripción Tabla Detalle_venta	32
Tabla 2-10: Descripción Tabla Caja	32
Tabla 2-11: Descripción Tabla Usuario_caja.....	33
Tabla 2-12: Descripción Tabla Giros	33
Tabla 2-13: Descripción Tabla Devoluciones	34
Tabla 2-14: Descripción Tabla Pedido.....	34
Tabla 2-15: Descripción Tabla Detalle_pedido	35
Tabla 2-16: Descripción Tabla Factura	36
Tabla 2-17: Descripción Tabla Detalle_factura	37
Tabla 3-1: Descripción de programas y objetivos del Sistema.....	44

INTRODUCCIÓN

La informática, definida como la ciencia encargada de la automatización del manejo de la información, gracias a su rápido crecimiento en el último siglo, ha transformado rápidamente a la sociedad actual, al ser una herramienta necesaria en el día a día, ya que se encuentra en diferentes áreas como lo son gubernamental, comercio, medicina, educación, comunicaciones, entretenimiento, entre otras.

En Chile, en la década de los '90, surgió el crecimiento en la utilización de las computadoras, y cada vez son más los dueños de negocios, tanto grandes como pequeños, que han optado por la posibilidad de migrar a soluciones informáticas computarizadas, facilitando así el control y la toma de decisiones en la administración al momento de que surgen las diferentes problemáticas, ya que son más acertadas.

Maybet, como es llamado el pequeño local comercial del que trata este trabajo, fue fundado en 1988 y ubicado en la ciudad de San Antonio, en una multitudinaria zona de comercio en el sector de Barrancas. Ofrece productos de librería, perfumería y artículos de paquetería. Magaly Tobar y Berta Tobar, son dos hermanas que conformaron una sociedad de hecho, las dos son dueñas y por el momento, las únicas que atienden el local comercial.

El objetivo de este Trabajo de Título es diseñar y desarrollar un sistema informático Web para *Maybet*, que en estos momentos no tiene soporte informático alguno. El sistema pretende ser un apoyo en la solución de problemas, tales como en el manejo de existencias de productos, así como también en el apoyo en la compra y venta de productos realizadas en el negocio.

CAPÍTULO 1: ASPECTOS RELEVANTES DEL DISEÑO LÓGICO

1 ASPECTOS RELEVANTES DEL DISEÑO LÓGICO

1.1 DESCRIPCIÓN DE LA ORGANIZACIÓN

En el año 1988 dos hermanas, Magaly Tobar junto a Berta Tobar, conformaron una sociedad de hecho con razón social *Tobar Hermosilla, Magaly y otra*, con el fin de comenzar un negocio ubicado en la calle Antofagasta #123, Barrancas, San Antonio, sector comercial transitado especialmente por personas que viven a los alrededores.

El negocio es llamado *Maybet*, y su giro comercial es Librería, Perfumería y Artículos de Paquetería.

Todo el año el local vende diversos productos ligados a su rubro comercial. Ejemplos de éstos se listan a continuación, clasificándose en temporadas: en marzo lo son cuadernos, lápices, estuches, agendas, entre otros artículos escolares; cercano a San Valentín, Día de la madre, Día del padre, Día del niño y Navidad lo son perfumes, cremas, juguetes y peluches, sólo por nombrar algunos; en septiembre se venden variados artículos de paquetería como blondas, cintas, cintas tricolores, especialmente para adornar vestimentas típicas de bailes chilenos.

Maybet tiene sistemas de pago en efectivo y RedCompra.

Es Magaly Tobar la que, para abastecer el negocio, se comunica con los proveedores que se encuentran en la región Metropolitana. Los productos llegan en encomiendas que una empresa de transportes en particular se encarga de entregar en la dirección del local, anteriormente señalada.

Entre las hermanas se turnan la atención a público, no requiriendo de más personal. Magaly Tobar se encarga del primer turno, con horario de 10:00 a 14:00 horas, intervalo en donde llegan los productos desde Santiago. Así mismo Berta se encarga de la atención entre 16:00 a 20:00 horas. En los días previos a las festividades, en donde los negocios en general tienen un alza en las ventas, Magaly y Berta Tobar atienden juntas todo el día.

En el año 2010 se amplió el local, logrando así el espacio necesario para dos pasillos, cinco vitrinas, cuatro mostradores y tres estanterías. *Maybet* tiene poca competencia en los alrededores; por lo tanto, uno de sus principales objetivos es satisfacer a la comunidad con los productos que se podrían encontrar viajando al centro de San Antonio, el sector comercial más importante de la ciudad.

1.2 DESCRIPCIÓN DE SITUACIÓN ACTUAL

1.2.1 Aspectos generales

Maybet es un local que es manejado manualmente, no posee caja registradora, y tiene una serie de productos a su venta - cuadernos, libretas, agendas, lápices, destacadores, correctores, entre otros, en su área de librería; perfumes, tinturas, cremas, desodorantes, sólo por nombrar algunos, en su categoría de perfumería y en paquetería se destacan productos como cintas, blondas o cierres. En su administración no se lleva un control de *stock* o un control de precios de los productos.

El funcionamiento del local comercial, como también lo es su administración, implica una serie de procesos que se detallarán a continuación; éstos incluyen proceso de ventas, proceso de compra al proveedor, control de precios de los productos, control de *stock*, control de devoluciones y proceso de registro de orden de pedidos al proveedor.

1.2.2 Proceso de ventas

El cliente le solicita el artículo a la vendedora, la que inmediatamente procede a buscar el producto deseado.

Si el cliente compra más de un artículo, el resultado de la suma de precios se hace en forma manual o mediante una calculadora, al igual que la determinación del vuelto, si es que fuese necesario.

El medio de pago utilizado en el local es efectivo y débito. Se debe pagar la totalidad de la venta, es decir, no aceptan pagos en cuotas, ya sean con tarjetas de créditos o con abonos de los clientes habituales.

Una vez concretada la venta, el registro de ella se realiza en la boleta con su fecha determinada, que se efectúa si la compra es superior a \$180; si no lo es, se anota en una agenda en el día de la compra. El local comercial no realiza ventas con factura.

Todos los días se lleva un registro manual del flujo de caja, pero no se lleva una lista detallada de los productos vendidos.

1.2.3 Proceso de compra al proveedor

La responsable del abastecimiento de *Maybet* es Magaly Tobar, quién, según su perspectiva, observa qué productos faltan. Procede a buscar catálogos impresos que dispone (las empresas proveedoras los envían cuando se acerca el ingreso a

clases, cuando se acercan las festividades o según la empresa los renueve) o busca en el catálogo en línea que los proveedores poseen en sus sitios web.

Los proveedores se encuentran en Santiago; por esta razón, el proceso lo hace llamando por teléfono o mandando un correo electrónico, enviando el código del producto que se encuentran en el catálogo de cada empresa y la cantidad deseada, sin haber ningún documento que registre tal orden de pedido por parte de Magaly Tobar. Si la empresa proveedora tiene *stock* para el pedido, se llega a un acuerdo de compra y despacha los productos.

Para la posterior verificación de los productos a entregar por el proveedor, si se realizan pedidos a través de un mail, la orden de compra queda guardada en los mensajes enviados, pero si el pedido se hace a través de una llamada telefónica, el registro de compra sólo queda en la mente de la dueña.

La entrega del pedido se hace en el mismo local. La encomienda llega por una reconocida empresa de transportes, junto con la factura. Se revisa la mercancía y, si todo está en orden, se acepta el pedido.

1.2.4 Control de precios de los productos

Cada producto lleva su precio respectivo en un papel adhesivo, el que se coloca uno a uno con una pistola etiquetadora de precios.

El precio de venta se calcula tomando como base el precio de compra y sumándole un porcentaje de ganancia definido por las vendedoras.

1.2.5 Control de *stock*

Actualmente, no existe un proceso automatizado que controle la cantidad de productos vendidos por el local o adquiridos para su pronta venta.

1.2.6 Control de devoluciones

Tal como el SERNAC¹ lo impone, el cliente debe acercarse al local comercial presentando la boleta que acredite la compra.

Las vendedoras deciden si el producto está en buenas condiciones para su devolución de dinero o el cambio por el mismo producto. El producto a devolver debe cumplir con requisitos tales como:

- No debe estar usado, por ejemplo, si es una agenda.
- Los productos abiertos deben estar con embalajes originales.
- Productos de perfumería no deben tener sus sellos originales rotos.

¹ Servicio Nacional del Consumidor.

1.3 PROBLEMAS DETECTADOS

En relación a los procesos anteriormente señalados, se detectan los siguientes problemas.

1.3.1 Proceso de ventas

Si el cliente efectúa una compra inferior a \$180, el registro se debe hacer en una agenda, pero generalmente se pasa por alto ese procedimiento.

En un momento del día al azar, no se sabe con certeza cuánto dinero se ha acumulado por las ventas, ya sea en efectivo y en comprobantes de pago de débito. Se debe sacar cálculos para saberlo.

Al no llevar una lista detallada de los productos vendidos, si se desea saber con certeza, por ejemplo, los productos con mayor demanda de hace un año atrás, con el objetivo de abastecer el local con tales productos, no se puede lograr.

1.3.2 Proceso de compra al proveedor

No se sabe exactamente la cantidad de artículos que hay en el negocio, más si se trata de artículos como cantidad de lápices o metros de alguna cinta en particular. Por tal motivo se atrasa el pedido al proveedor de ciertos artículos, ya que hay que actualizar manualmente el control de existencias que lleva la contadora del local comercial, o simplemente el pedido se hace al momento de percatarse que existe una cifra baja de cierto producto, y como el despacho toma su tiempo para llegar a la región, el local queda sin stock.

Cuando un pedido se efectúa a través de una llamada telefónica no existe un registro de orden de pedido por parte de la dueña. Al momento de que llegan los pedidos, sólo se confía en su memoria para la verificación de la compra.

1.3.3 Control de precios de los productos

Al ser dos personas las que atienden el local comercial, pueden variar los precios de los productos, si es que por alguna razón se pierde el papel adhesivo en éste, arriesgándose a una posible queja del cliente.

Si se desea hacer una rebaja de un producto en particular se debe cambiar el precio artículo por artículo. Al igual, si se compra el mismo producto al siguiente mes y al por mayor cuesta más, en el local deben subir los precios.

1.3.4 Control de stock

Al no llevar un registro de los productos vendidos en el día a día, el inventario no es exacto y esto complica el proceso de compra a los proveedores al no tener la certeza de qué productos tienen un stock crítico.

1.3.5 Control de devoluciones

El principal problema es que no se lleva un registro de los productos devueltos al local. Si están en buen estado, se deben regresar al stock y si no, es pérdida para el local. No muchas empresas proveedoras hacen devoluciones de pedidos luego que pasa cierta cantidad de tiempo.

Un problema en particular es el hecho que la boleta de Maybet no contiene el detalle de los productos comprados, sólo el total de la venta, y si es que el cliente compró varios artículos puede pasar lo siguiente: La mercadería que ofrece Maybet se puede encontrar en muchos locales con el mismo giro comercial, y si el cliente lleva una boleta con una suma igual o que supere el precio del producto y lo desea cambiar, puede que lo haya comprado en otro lado, lo que genera una pérdida para el local.

1.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO

Como resultado de los problemas anteriormente señalados (punto 1.3), causados por la administración actual del negocio, se ha propuesto un sistema informático Web para automatizar la compra y venta de productos.

1.4.1 Objetivos y beneficios del sistema propuesto

1.4.1.1 Objetivo Principal

El objetivo principal es desarrollar un sistema informático web que por medio de una interfaz de fácil uso y una base de datos, permita un control lo más exacto posible y autónomo de la información manejada en el local comercial, como lo es el control de existencias y el control de ventas, para que la atención al cliente sea de forma rápida y adecuada. Otros objetivos son el cálculo de ventas diarias, y el informe de ventas (historia), ya sea en efectivo o débito y el control de las salidas de dinero,

tanto compras a los proveedores como posibles adelantos de remuneraciones de las vendedoras.

1.4.1.2 Objetivos Específicos

- Control y orden al identificar cada producto existente en *Maybet* mediante códigos únicos.
- Modificación y consulta de los precios de los productos.
- Control sobre el flujo de caja diario tanto en entradas de dinero, efectivo o débito, como en salidas de dinero, sea para pagar por los productos recibidos o para pagar el sueldo de las trabajadoras.
- Control de ventas diarias, mensuales y anuales para poder fijar metas de ventas en los años posteriores.
- Llevar un registro formal de proveedores tanto frecuentes como nuevos.
- Crear un sistema de tipos de usuarios que permita filtrar las distintas funcionalidades del sistema.
- Crear un sistema que se ejecute conjuntamente con una base de datos y que permita generar informes con los datos obtenidos de ésta.
- Crear una interfaz gráfica simple, para una fácil utilización por parte del usuario y un correcto ingreso, modificación o eliminación de datos por parte de éste.

1.4.1.3 Beneficios del sistema propuesto

La labor de las vendedoras se agilizará, ya sea en la atención del cliente como en el momento de realizar el conteo de los productos, la información manejada será consistente. También se agilizará el proceso de realizar cálculos necesarios para la cuadratura, solventando así las debilidades que tiene *Maybet*.

Se tomarán decisiones más acertadas en cuanto a la compra de productos.

1.4.2 Descripción general de la situación propuesta

El sistema contará con dos tipos de usuarios: usuario-administrador y usuario-vendedor, cada uno con sus privilegios respectivos. El usuario-administrador podrá ser capaz de manipular atributos en la base de datos, mientras un usuario-vendedor sólo podrá consultarlos.

Si el cliente desea conocer detalles de los productos, como es su precio o su stock, sólo se deberá hacer una búsqueda del producto en el sistema. La vendedora tendrá la opción de buscarlo por código o nombre.

Si la persona decide comprar el producto, se efectúa el proceso de venta, en que el cliente podrá comprar una cantidad de uno o más artículos, ya sea en efectivo

o en débito. Se concretará una venta pagando con dinero en efectivo o débito y se realizará el registro pertinente. Si el débito resultara rechazado, se le preguntará a la persona si lo desea pagar con efectivo; si no es el caso, se pondrá fin al proceso.

El proceso para efectuar la compra al proveedor será como la administradora lo maneja actualmente, sólo con la diferencia que se podrá ver un informe del *stock* crítico, y en él, cada producto con su cantidad y asociado con su(s) proveedor(es), lo que facilitará la tarea para contactar a la empresa requerida.

Al llegar las encomiendas con los productos, el usuario-administrador llevará a cabo la tarea de llenar la base de datos con los nuevos productos, agregándolos a los ya existentes, modificando los atributos o creando un registro nuevo con sus propios atributos.

Al terminar el día, el sistema tendrá una opción de cierre de caja donde se generará un registro detallado de éste, lo que facilitará el trabajo de cuadratura de caja.

1.4.3 Diagramas de flujos administrativos

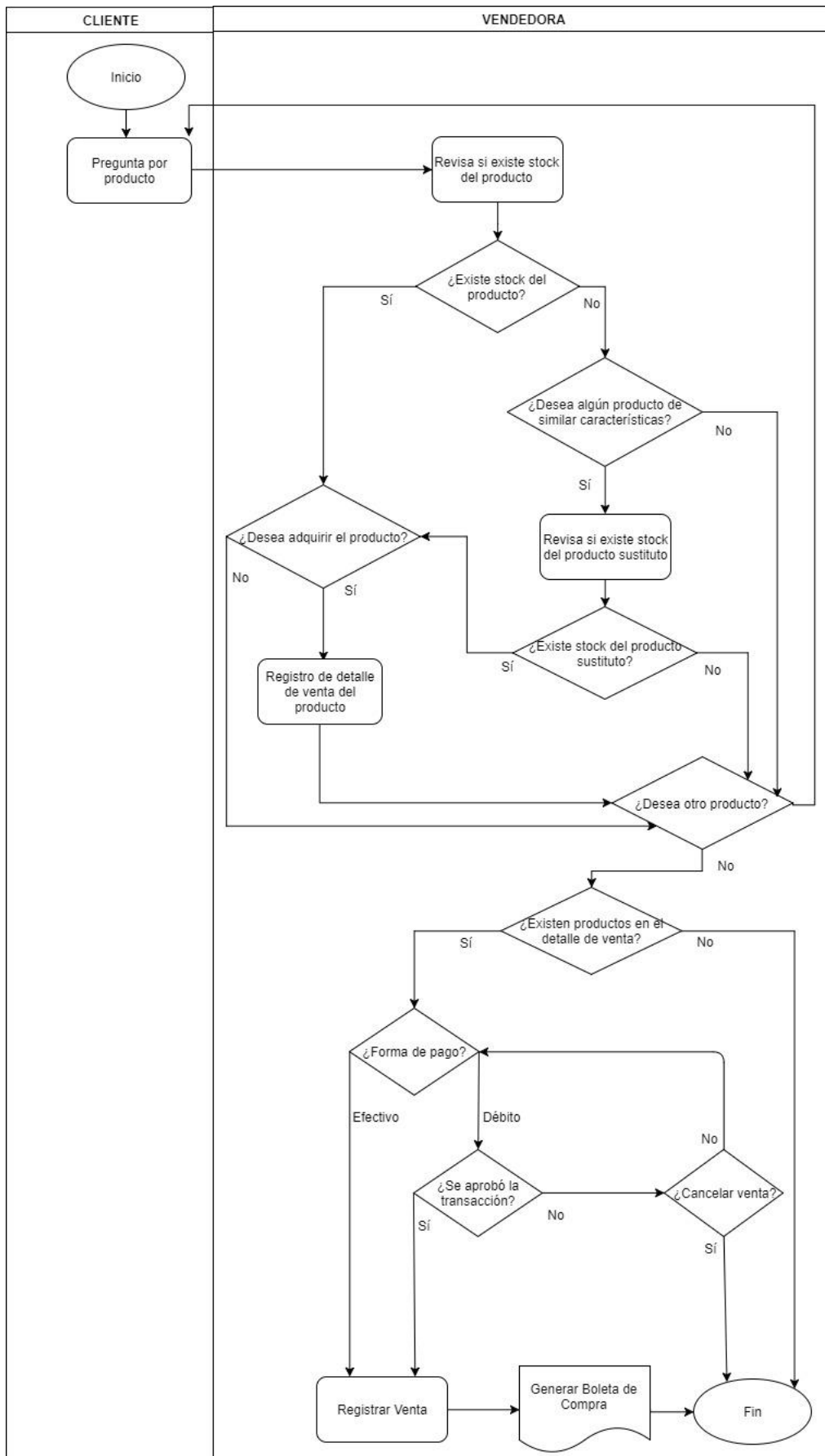


Diagrama 1-1: Diagrama de flujo administrativo del proceso de ventas

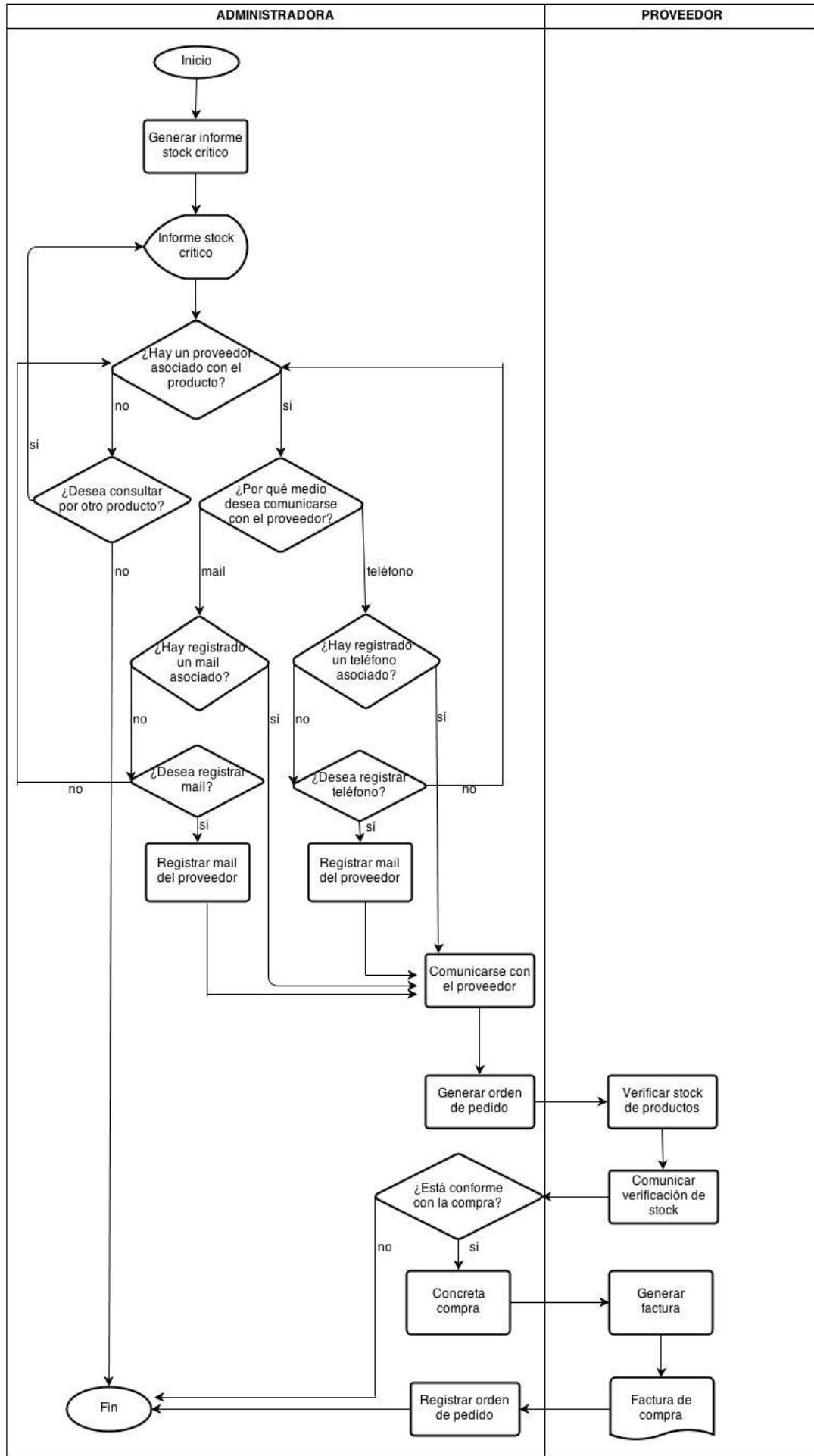


Diagrama 1-2: Diagrama de flujo administrativo de compra al proveedor

1.4.4 Estructura funcional del sistema

A continuación, se describen las funcionalidades del sistema a desarrollar:

- a) Mantenedor de productos: Tanto para agregar, modificar o consultar artículos en la base de datos. Eliminar un artículo será posible siempre y cuando no esté asociada tanto a una venta en proceso como a una realizada. Sólo usuarios administradores tendrán el acceso total a esta funcionalidad, en cambio un usuario con acceso limitado sólo podrá consultar artículos en la base de datos.
- b) Mantenedor de categorías: Agregar, modificar o consultar las categorías en que se clasifican los productos, ya sea librería, perfumería o paquetería. La eliminación sólo será posible si no está asociada a una subcategoría. Sólo usuarios administradores tendrán acceso.
- c) Mantenedor de subcategorías: Agregar, modificar o consultar las subcategorías en que se clasifican los productos. Si la subcategoría no está asociada a un producto será posible su eliminación. Un ejemplo de subcategoría es Cremas en la categoría Perfumería. Sólo usuarios administradores tendrán acceso.
- d) Mantenedor de proveedores: Agregar, modificar o consultar proveedores. Eliminar un proveedor será posible siempre y cuando no esté asociado a un producto de una venta en proceso o de una factura de compra registrada. Tendrán acceso los usuarios administradores.
- e) Mantenedor de usuarios: Los usuarios administradores tendrán la opción de agregar nuevos usuarios con los permisos que se deseen. También podrán eliminar usuarios, pero siempre que se cumplan las siguientes condiciones: debe quedar por lo menos un usuario administrador, el usuario que iniciará sesión no podrá eliminarse a sí mismo, y que el usuario a eliminar no esté asociado a un giro, un ajuste de existencias, un pedido, a la caja, a una venta o a una devolución. Así también, podrán tener la opción de modificar permisos, contraseñas o campos no claves y de generar informes de los perfiles actuales.
- f) Mantenedor de devoluciones: Los usuarios vendedores podrán agregar y consultar devoluciones. Sólo los usuarios administradores tendrán la opción de eliminación.
- g) Mantenedor de ajustes en el inventario (sólo en el rango de control de existencias): Tendrá la función de agregar modificaciones a las cantidades de los productos, tanto para añadir como para eliminar (mermas). El acceso será restringido para usuarios administradores.

- h) Mantenedor de giros: Agregar, consultar o eliminar giros de dinero. El acceso será restringido para usuarios administradores.
- i) Mantenedor de pedidos: Agregar, modificar, consultar o eliminar orden de pedidos a proveedores. Sólo usuarios administradores tendrán acceso.
- j) Mantenedor de facturas: Se podrán agregar facturas de compra, modificar sus atributos, y consultar las facturas en la base de datos. Para eliminar una factura será posible sólo si no hay un producto o proveedor, existentes en el sistema, asociado a ella. Sólo usuarios administradores tendrán acceso.
- k) Controlador de ventas: Cuando se realiza una venta se modifica el *stock* de los productos en la base de datos. También se registran los nuevos datos que identifiquen la venta, como lo son el código de venta, el número de boleta, la fecha, el total de venta, entre otros. Además, se podrán consultar ventas por fechas o número de venta.
- l) Controlador de caja: Al cerrar la caja, se agrega el total de dinero recaudado en el día, junto con la fecha y el usuario que realizó la labor, es decir, se agrega el movimiento diario de caja. La opción de cierre de caja la puede realizar un usuario administrador y un usuario vendedor. Sólo el usuario administrador tendrá la opción de consultar el historial de movimiento de caja.

1.4.5 Descripción de fórmulas y cálculos empleados

En este punto, se describirán las fórmulas y cálculos que se emplearán en el sistema a desarrollar.

- a) Total de la venta: Σ total_detalle_venta
- b) Total del detalle de una venta: cantidad_prod_venta*precio_unitario
- c) Total de la factura: Σ total_detalle_factura
- d) Total_detalle_factura: cantidad_prod_fact* precio_comprado
- e) Total del pedido: Σ total_detalle_pedido
- f) Total_detalle_pedido: cantidad_prod_pedido * precio_pedido
- g) Obtención del vuelto: efectivo_pagado – total_venta

h) Total caja: $((\text{ventas_sin_boleta} + \text{ventas_con_boleta}) - \text{total_giro}) + \text{efectivo_inicial}$

1.4.6 Entradas y salidas

A continuación, se registran las entradas y salidas del sistema.

1.4.6.1 Entradas

- Datos del producto: Código del producto, código de barras, nombre del producto, cantidad, precio unitario de venta, y código del proveedor.
- Datos de categoría: Su código y el nombre de la categoría.
- Datos de subcategoría: Su código, nombre de la subcategoría y código de la categoría a la cual pertenece.
- Datos del proveedor: Código del proveedor, nombre de la empresa, como también su dirección, la ciudad en donde se encuentra, su teléfono y su mail para contacto. También entre los datos está el nombre y apellido del encargado de la venta en dicha empresa.
- Datos del usuario: Su código respectivo, el tipo de usuario, ya sea usuario administrador o usuario común, nombre, dirección, teléfono y su *password* de autenticación.
- Datos de devoluciones: Código de la devolución, el código del producto devuelto, el código del usuario responsable, el precio histórico del producto y un comentario del porqué de la devolución.
- Datos de ajuste de existencias: Código de ajuste, código de producto para hacer el ajuste, código de usuario que realizó el ajuste, la fecha, la nueva cantidad, el tipo de ajuste, ya sea positivo o negativo en el *stock* y de productos, y un comentario de porqué se hizo el cambio.
- Datos de giro: Código de giro, código de la caja activa, el código del usuario responsable del efectúo del giro, el monto de dinero y el motivo por el cual realizó el giro.
- Datos de orden de pedidos: Código del pedido, código del usuario encargado, el código del proveedor, códigos de los productos, la cantidad pedida, precios de los

productos, la fecha en que se realizó el pedido, el total, el estado en que se encuentra y un comentario.

- Datos de la factura: Código de la factura, código del proveedor, códigos de los productos, sus cantidades respectivas, el precio en el cual se compraron, la fecha de la compra, el IVA², el total de la factura y el estado de pago.
- Datos de venta: Código de venta como clave única, código de la caja activa, número de boleta autorizada por el SII³, el código de usuario que realizó la venta, códigos de los productos vendidos, cantidades de los productos, precios de los productos, el tipo de pago, ya sea en efectivo o débito, y el total a pagar.

1.4.6.2 Salidas

- Informe de control de existencias: Informe de todos los productos que tiene el local, filtrados por categoría y subcategoría, mencionando también la cantidad existente de cada uno de ellos; o buscados por código o palabra clave, se listarán con su respectiva información: código interno, código de barras, categoría y subcategoría a la que pertenece, nombre del producto, cantidad existente, cantidad crítica, su precio y sus proveedores.
- Informe de stock crítico: Lista de productos con el *stock* crítico o sin *stock*. Se visualizarán cuando la cantidad del producto sea menor o igual a la cantidad crítica del producto.
- Informe de categorías: Listado de las categorías disponibles con sus respectivos datos.
- Informe de las subcategorías: Listado de las subcategorías disponibles con sus respectivos datos.
- Informe de proveedores: Es un listado de los proveedores con todos sus datos asociados.
- Informe de usuarios registrados: Lista de los usuarios, con sus respectivos datos, menos el *password*.
- Informe de devoluciones: Listado con las devoluciones con sus respectivos datos.

² Impuesto sobre el valor agregado.

³ Servicio de Impuestos Internos.

- Informe de ajustes: Listado de los ajustes realizados en las existencias de los productos, filtrado por fechas.
- Informe de giros: Listado de los giros realizados con sus respectivos datos.
- Informe de órdenes de pedido: Un listado con los pedidos hechos al proveedor por fecha.
- Informe de facturas: Listado de las facturas recibidas del proveedor.
- Informes de ventas: Listado de ventas en la fecha mencionada por el usuario o por el número de venta.
- Informe de movimientos diarios de caja: Listado del flujo de caja.
- Informes de cuadratura de caja: Totales de caja, seleccionados por fecha.

1.4.7 Entidades de información

A continuación, se describen las entidades y se muestra el modelo de datos relacional utilizado en el sistema.

1.4.7.1 Descripción de entidades

Breve descripción de entidades que intervienen en el sistema:

- **Producto**: Entidad que contiene los datos de los productos existentes en el local.
- **Categoría**: Contiene las categorías en las que se puede clasificar un producto, por ejemplo, Paquetería.
- **Subcategoría**: Contiene las subcategorías en las que se puede clasificar un producto. Por ejemplo, la categoría Paquetería se podría subdividir en Blondas.
- **Proveedor**: Contiene los datos de las diferentes empresas de proveedores a los que el local ha comprado sus productos.
- **Prod_prov**: Asocia cada producto con un proveedor. Un producto en particular puede tener varios proveedores, así como un proveedor puede tener varios productos en el sistema.
- **Usuario**: Almacena los datos de las personas que utilizarán el sistema.
- **Usuario_caja**: Asocia cada usuario con un movimiento de caja. Un usuario puede realizar varios movimientos, así como los movimientos de caja pueden ser realizados por varios usuarios.

- Devoluciones: Almacena los datos de las devoluciones de productos.
- Ajustes: Entidad que almacena los ajustes en el inventario.
- Giros: Entidad que almacena información de salida de dinero de caja.
- Pedido: Entidad que almacena las órdenes de pedido hacia los proveedores.
- Detalle_pedido: Entidad que almacena los detalles de cada pedido.
- Factura: Entidad que contiene los datos de cada factura.
- Detalle_factura: Entidad que almacena los detalles de las facturas.
- Venta: Entidad que contiene las ventas realizadas por el local.
- Detalle_venta: Entidad que almacena los detalles de cada venta.
- Caja: Contiene los datos del movimiento de caja. Incluye ventas con boleta, así como las ventas menores a \$180 pesos; el dinero en efectivo y en comprobantes de débito; las salidas de dinero de caja y el total diario de dinero en el de cierre de caja.

1.4.7.2 Modelo de datos relacional

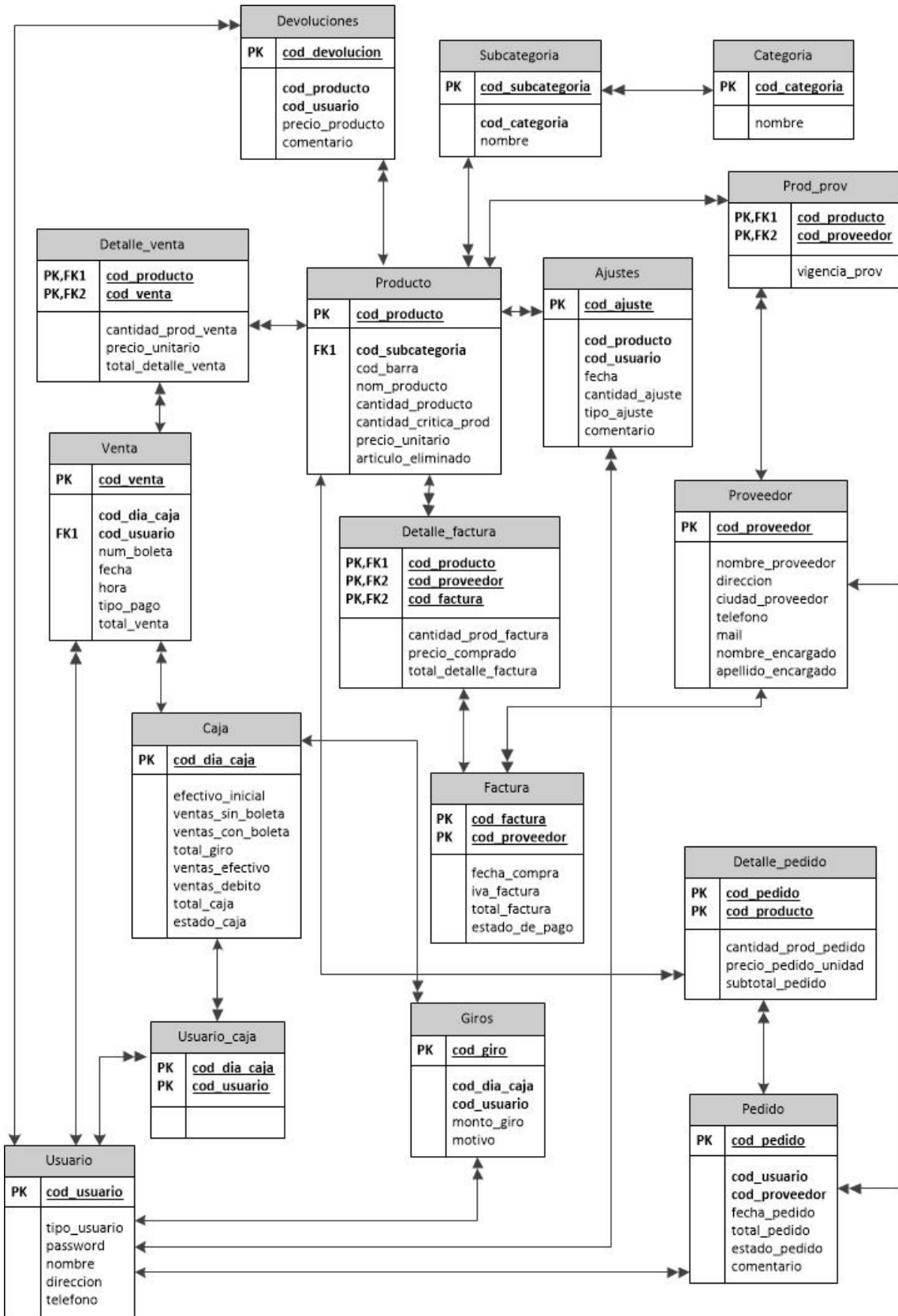


Figura 1-1: Modelo relacional del sistema.

1.4.8 Estructura de códigos

- Código del producto: Será un número correlativo que identifica un producto en particular.
- Producto eliminado: 0: Producto existente; 1: Producto eliminado lógicamente.
- Código de la categoría: Será un número correlativo que identifique a cada categoría.
- Código de la subcategoría: Será un número correlativo que identifica una subcategoría.
- Código de proveedor: Rut de la empresa proveedora.
- Código de usuario: Será el rut del usuario.
- Tipo de usuario: 'ADM': Cuenta usuario de tipo administrador que posee todos los privilegios; 'VEND': Cuenta usuario vendedor.
- Código de devoluciones: Será un número correlativo que identifique a cada devolución.
- Código de ajuste: Será un número correlativo que identifique a cada ajuste realizado.
- Tipo de ajuste: 0: Ajuste negativo (pérdida física, o merma, de algún producto); 1: Ajuste positivo (Ingreso de productos al sistema).
- Código de giro: Número correlativo que identifica a cada salida de dinero.
- Código de pedidos: Será un número correlativo.
- Estado del pedido: 'REC' es pedido recibido, 'PEND' es pendiente y 'RECH' es un pedido rechazado.
- Código de la factura: Será el número de factura entregada por el proveedor.
- Estado de pago de la factura: 0 es pagada y 1 es que aún se debe el dinero.
- Código de venta: Será un número correlativo.
- Tipo de venta: 'EF': Venta pagada en efectivo; 'DB': Venta pagada con débito.
- Código de caja: Código proporcionado por la fecha en el sistema.
- Estado de caja: 'A': Caja activa; 'D': Caja desactivada o cerrada.

1.4.9 Condicionantes de diseño

Para desarrollar un sistema computacional para Maybet, se optó por el lenguaje de programación interpretado PHP que se utiliza especialmente para el desarrollo web y su uso es gratuito. Para la base de datos se utilizará MySQL. Como servidor se utilizará Apache.

El local deberá contar con un computador de escritorio conexión a Internet.

CAPÍTULO 2: MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS

2 MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS

2.1 DESCRIPCIÓN DE LAS CARACTERÍSTICAS DEL RECURSO COMPUTACIONAL

A continuación, se detallarán las características y requerimientos del recurso computacional que se utilizará para el desarrollo del Sistema Informático Web para el local *Maybet*.

2.1.1 Configuración del sistema

Equipo de desarrollo.

- Notebook SAMSUNG, modelo R430.
- Procesador Intel Pentium T4400 (2,20 GHz).
- Memoria RAM de 2 GB.
- Disco duro de 250 GB.
- Conectividad: Interfaz de red alámbrica 10/100; Inalámbrica 802.11 bg/n.
- Pantalla LED HD de 14,0" (1.366 x 768) 16:9 "

Equipo donde será implantado el sistema.

- Computador de escritorio COMPAQ, modelo CQ2302LA.
- Procesador AMD Athlon 2650e (1,6 GHz).
- Memoria RAM de 2 GB.
- Disco duro de 320 GB.
- Red (LAN): Interfaz de red 10/100.
- Monitor LCD COMPAQ Widescreen CQ1569 de 15,6"

2.1.2 Software utilizado

A continuación, se describirá el software utilizado en el equipo de desarrollo del sistema.

- a) Sistema Operativo: Microsoft Windows 7 Professional, ya que es el más usado a nivel mundial, proporciona un ambiente gráfico de fácil uso, además de ser multitarea y multiusuario.

- b) Lenguaje de Programación: PHP, siglas de *Hypertext Preprocessor* (originalmente de *Personal Home Page*) es el lenguaje más popular para el servidor Apache y algunas ventajas que tiene son:
- Multiplataforma, para ser usado tanto en Windows como Linux.
 - Licencia libre.
- c) Sistema administrador de la base de datos: MySQL, software Open Source, es decir, libre de licencia y multiplataforma.
- d) Xampp: XAMPP, es un servidor de multiplataforma y libre. Integra en una sola aplicación un servidor web Apache, intérpretes de lenguaje de scripts PHP, un servidor de base de datos MySQL, entre otros módulos.
- e) Dreamweaver: Utiliza tecnologías web como CSS y Java Script. No es primordial tener conocimientos de código HTML para diseñar páginas web. Dreamweaver permite previsualizar las páginas en casi todos los navegadores.

2.2 DESCRIPCIÓN DE ARCHIVOS

A continuación, se listarán los archivos y se describirán las tablas presentes en la base de datos.

2.2.1 Listado de archivos

A continuación, se listarán todos los archivos presentes en la base de datos.

- Producto: Almacena los datos de los productos.
- Categoría: Almacena los datos de la categoría.
- Subcategoría: Almacena los datos de la subcategoría.
- Proveedor: Almacena los datos de los proveedores.
- Prod_prov: Almacena las claves primarias para asociar cada producto con un proveedor.
- Usuario: Almacena los datos de las personas que utilizarán el sistema.
- Devoluciones: Almacena los datos de las devoluciones de productos.
- Ajustes: Almacena los ajustes para el control de existencias.
- Giros: Almacena información de salidas de dinero de caja.
- Pedido: Almacena las órdenes de pedido hacia los proveedores.
- Detalle_pedido: Almacena los detalles de cada pedido.
- Factura: Almacena los datos de cada factura.

- Detalle_factura: Almacena los detalles de las facturas.
- Venta: Almacena las ventas realizadas en el local.
- Detalle_venta: Almacena las los detalles de cada venta.
- Caja: Almacena los datos del movimiento de caja diario.
- Usuario_caja: Almacena las claves primarias para asociar cada usuario con un movimiento de caja específico.

2.2.2 Descripción de tablas

A continuación, se describirán todas las tablas contenidas en la base de datos.

2.2.2.1 Tabla Producto

Nombre: Producto

Descripción: Tabla que almacenará los datos de los productos presentes en el local.

Clave primaria: cod_producto

Claves foráneas:

cod_subcategoria, referencia a tabla Subcategoria(cod_subcategoria)

Tabla 2-1: Descripción Tabla Producto

Nombre	Tipo	Descripción
cod_producto	SmallInt(5)	Corresponde a un número correlativo que identifica un producto.
cod_subcategoria	TinyInt (2)	Identificador de una subcategoría.
cod_barra	VarChar(13)	Corresponde al código de barras, proporcionado de fábrica, presente en los artículos.
nom_producto	VarChar(10)	Nombre del producto.
cantidad_producto	SmallInt(5)	Cantidad disponible de productos en el momento en que se consulta.
cantidad_critica_prod	TinyInt(2)	Stock crítico del producto.
precio_unitario	Int (6)	Corresponde al precio de venta del producto que se registra en ese momento (ya posee el IVA).
articulo_eliminado	TinyInt(1)	Indica si el producto fue eliminado lógicamente del sistema. 0 = Existe

Nombre	Tipo	Descripción
		1 = Eliminado

2.2.2.2 Tabla Categoría

Nombre: Categoría

Descripción: Tabla que almacenará los datos de las categorías.

Clave primaria: cod_categoria

Tabla 2-2: Descripción Tabla Categoría

Nombre	Tipo	Descripción
cod_categoria	TinyInt(2)	Número correlativo que identifica una categoría.
nombre	VarChar(30)	Nombre de la categoría.

2.2.2.3 Tabla Subcategoría

Nombre: Subcategoría

Descripción: Tabla que almacenará los datos de las subcategorías.

Clave primaria: cod_subcategoría

Claves foráneas: cod_categoria, referencia a tabla Categoría(cod_categoria)

Tabla 2-3: Descripción Tabla Subcategoría

Nombre	Tipo	Descripción
cod_subcategoría	TinyInt (2)	Número correlativo que identifica una subcategoría.
cod_categoria	TinyInt(2)	Número correlativo que identifica una categoría.
nombre	VarChar(30)	Nombre de la subcategoría.

2.2.2.4 Tabla Proveedor

Nombre: Proveedor

Descripción: Tabla que almacenará los datos de proveedores de la empresa.

Clave primaria: cod_proveedor

Tabla 2-4: Descripción Tabla Proveedor

Nombre	Tipo	Descripción
cod_proveedor	VarChar(10)	Rut de la empresa proveedora.
nombre_proveedor	VarChar(30)	Nombre de la empresa proveedora.
direccion	VarChar(40)	Dirección de la empresa proveedora.
ciudad_proveedor	VarChar(30)	Ciudad de la empresa proveedora.
telefono	VarChar(12)	Teléfono de contacto del proveedor.
mail	VarChar(40)	Mail de contacto del proveedor
nombre_encargado	VarChar(30)	Nombre del encargado de ventas de la empresa.
apellido_encargado	VarChar(30)	Apellido del encargado de ventas de la empresa.

2.2.2.5 Tabla Prod_prov

Nombre: Prod_prov

Descripción: Tabla que permite la vinculación de cada proveedor con sus productos.

Clave primaria: cod_producto + cod_proveedor

Claves foráneas: cod_producto, referencia a tabla Producto(cod_producto);
cod_proveedor, referencia a tabla Proveedor(cod_proveedor).

Tabla 2-5: Descripción Tabla Prod_prov

Nombre	Tipo	Descripción
cod_producto	SmallInt(5)	Corresponde a un número correlativo que identifica un producto.
cod_proveedor	VarChar(10)	Rut de la empresa proveedora.
vigencia_prov	TinyInt(1)	Indica la vigencia de un proveedor asociado a un producto, es decir, si ya no provee el producto se elimina lógicamente. 0 = Vigente

Nombre	Tipo	Descripción
		1 = Eliminado

2.2.2.6 Tabla Ajustes

Nombre: Ajustes

Descripción: Tabla que almacena ajustes al *stock* de los productos.

Clave primaria: cod_ajuste

Claves foráneas: cod_producto, referencia a tabla Producto(cod_producto);
cod_usuario, referencia a tabla Usuario(cod_usuario).

Tabla 2-6: Descripción Tabla Ajustes

Nombre	Tipo	Descripción
cod_ajuste	SmallInt(5)	Número correlativo que identifica cada ajuste realizado.
cod_producto	SmallInt(5)	Corresponde a un número correlativo que identifica un producto.
cod_usuario	VarChar(9)	Corresponde al rut del usuario.
fecha	Date	Fecha en la que se realizó el ajuste.
cantidad_ajuste	TintInt(3)	Cantidad del ajuste.
tipo_ajuste	TinyInt(1)	Indica si el ajuste fue negativo (pérdida física de algún producto, o sea, merma) o positivo (aumento de productos). 0 = Negativo 1 = Positivo
comentario	VarChar(40)	Comentario en que se indica el porqué del cambio de ajuste

2.2.2.7 Tabla Usuario

Nombre: Usuario

Descripción: Tabla que almacenará los datos de los usuarios registrados en el sistema, indicando también sus privilegios.

Clave primaria: cod_usuario

Tabla 2-7: Descripción Tabla Usuario

Nombre	Tipo	Descripción
cod_usuario	VarChar(9)	Corresponde al rut del usuario.
tipo_usuario	Char(3)	Corresponde a los tipos de usuario en el sistema. 'ADM' = Cuenta usuario de tipo administrador. 'ASI' = Cuenta usuario asistente.
password	VarChar(10)	Contraseña que valida el ingreso al sistema.
nombre	VarChar(20)	Nombre del usuario.
direccion	VarChar(40)	Dirección del usuario.
telefono	VarChar(12)	Teléfono de contacto del usuario.

2.2.2.8 Tabla Venta

Nombre: Venta

Descripción: Tabla que almacenará los datos de las ventas.

Clave primaria: cod_venta

Claves foráneas: cod_usuario, referencia a tabla Usuario(cod_usuario)

Tabla 2-8: Descripción Tabla Venta

Nombre	Tipo	Descripción
cod_venta	Int	Número correlativo único que identifica a una venta en forma lógica.
cod_dia_caja	Date	Fecha de la venta.
cod_usuario	VarChar(9)	Corresponde al rut del usuario.
num_boleta	Int	Número correlativo único que identifica a una venta en forma física, es decir, una boleta sólo cuando las ventas son superiores a \$180 pesos.
hora	Date	Hora de la venta.
tipo_pago	Char(2)	Corresponde al tipo de pago por los productos. 'EF' = Efectivo. 'DB' = Débito.
total_venta	Int(7)	Monto total de la venta.

2.2.2.9 Tabla Detalle venta

Nombre: Detalle_venta

Descripción: Tabla que almacenará los datos del detalle de una venta.

Clave primaria: cod_producto + cod_venta

Claves foráneas: cod_producto, referencia a tabla Producto(cod_producto);
cod_venta, referencia a tabla Venta(cod_venta).

Tabla 2-9: Descripción Tabla Detalle_venta

Nombre	Tipo	Descripción
cod_producto	SmallInt(5)	Corresponde a un número correlativo que identifica un producto.
cod_venta	Int	Número correlativo único que identifica a una venta en forma lógica.
cantidad_prod_venta	SmallInt(3)	Cantidad solicitada del producto.
precio_prod	Int (6)	Precio unitario del producto.
total_detalle_venta	Int(7)	Corresponde al total del detalle de la venta.

2.2.2.10 Tabla Caja

Nombre: Caja

Descripción: Tabla que almacenará los datos del movimiento de caja diario.

Clave primaria: cod_dia_caja

Claves foráneas: cod_venta, referencia a tabla Venta(cod_venta).

Tabla 2-10: Descripción Tabla Caja

Nombre	Tipo	Descripción
cod_dia_caja	Date	Fecha trabajada.
efectivo_inicial	Int(7)	Cantidad solicitada del producto.
ventas_sin_boleta	Int(7)	Total de ventas menores a \$180 pesos.
ventas_con_boleta	Int(7)	Total de ventas superiores a \$180 pesos.
total_giro	Int(8)	Total de salidas de dinero de caja hechos en el día.
ventas_efectivo	Int(7)	Total de dinero en efectivo.
ventas_debito	Int(7)	Total de dinero en débito.
total_caja	Int(8)	Total de dinero en caja, en efectivo y débito.
estado_caja	VarChar(1)	Corresponde al estado en que se encuentra la caja

Nombre	Tipo	Descripción
		'A' = Caja activa para realizar ventas. 'D' = Caja desactivada o cerrada.

2.2.2.11 Tabla Usuario_caja

Nombre: Usuario_caja

Descripción: Tabla que permite la vinculación de cada usuario con la caja.

Clave primaria: cod_dia_caja + cod_usuario

Claves foráneas: cod_dia_caja, referencia a tabla Caja(cod_dia_caja);
cod_usuario, referencia a tabla Usuario(cod_usuario)

Tabla 2-11: Descripción Tabla Usuario_caja

Nombre	Tipo	Descripción
cod_dia_caja	Date	Fecha trabajada.
cod_usuario	VarChar(9)	Corresponde al rut del usuario.

2.2.2.12 Tabla Giros

Nombre: Giros

Descripción: Tabla que almacenará información de salidas de dinero de caja.

Clave primaria: cod_giro

Claves foráneas: cod_dia_caja, referencia a tabla Caja(cod_dia_caja);
cod_usuario, referencia a tabla Usuario(cod_usuario)

Tabla 2-12: Descripción Tabla Giros

Nombre	Tipo	Descripción
cod_giro	Int	Número correlativo que identifica cada salida de dinero.
cod_dia_caja	Date	Fecha trabajada.
cod_usuario	VarChar(9)	Corresponde al rut del usuario.
monto_giro	Int(7)	Monto de dinero.
motivo	VarChar(40)	Comentario en que se indica el porqué del giro de dinero.

2.2.2.13 Tabla Devoluciones

Nombre: Devoluciones

Descripción: Tabla que almacenará información de las devoluciones, registradas sólo por usuarios vendedores y por esa razón no implican actualización en la tabla Productos.

Clave primaria: cod_devolucion

Claves foráneas: cod_producto, referencia a tabla Producto(cod_producto);
cod_usuario, referencia a tabla Usuario(cod_usuario).

Tabla 2-13: Descripción Tabla Devoluciones

Nombre	Tipo	Descripción
cod_devolucion	Int	Número correlativo que identifica a una devolución.
cod_producto	SmallInt(5)	Corresponde a un número correlativo que identifica un producto.
cod_usuario	VarChar(9)	Corresponde al rut del usuario.
precio_producto	Int (6)	Corresponde al precio del producto que se registró al momento de la venta, o si es merma, está registrado en ese momento.
comentario	VarChar(40)	Comentario del motivo de la devolución o merma.

2.2.2.14 Tabla Pedido

Nombre: Pedido

Descripción: Tabla que almacenará los datos de los pedidos hacia los proveedores.

Clave primaria: cod_pedido

Claves foráneas: cod_usuario, referencia a tabla Usuario(cod_usuario);
cod_proveedor, referencia a tabla Proveedor(cod_proveedor).

Tabla 2-14: Descripción Tabla Pedido

Nombre	Tipo	Descripción
cod_pedido	Int	Número correlativo que identifica un pedido.
cod_usuario	VarChar(9)	Corresponde al rut del usuario.
cod_proveedor	VarChar(10)	Rut de la empresa proveedora.
fecha_pedido	Date	Fecha en que se hizo el pedido.
total_pedido	Int(8)	Monto total del pedido.
estado_pedido	Char(4)	Corresponde al estado en que se encuentra el pedido. Recibido = RECI Pendiente = PEND Rechazado = RECH
comentario	VarChar(40)	Usado para documentar el medio del pedido, ya sea por teléfono o mail, o para agregar una nota por parte de la administradora.

2.2.2.15 Tabla Detalle_pedido

Nombre: Detalle_pedido

Descripción: Tabla que almacena los detalles de cada pedido.

Clave primaria: cod_pedido + cod_producto

Claves foráneas: cod_pedido, referencia a tabla Pedido(cod_pedido);
cod_producto, referencia a tabla Producto(cod_producto).

Tabla 2-15: Descripción Tabla Detalle_pedido

Nombre	Tipo	Descripción
cod_pedido	Int	Número correlativo que identifica un pedido.
cod_producto	SmallInt(5)	Corresponde a un número correlativo que identifica un producto.
cantidad_prod_pedido	TinyInt (3)	Cantidad solicitada del producto.

Nombre	Tipo	Descripción
precio_pedido_unidad	Int(6)	Precio unitario del producto.
subtotal_pedido	Int(7)	Total del detalle del pedido al proveedor.

2.2.2.16 Tabla Factura

Nombre: Factura

Descripción: Tabla que contiene los datos de las facturas.

Clave primaria: cod_factura + cod_proveedor

Claves foráneas:

cod_proveedor referencia a la tabla Proveedor (cod_proveedor).

Tabla 2-16: Descripción Tabla Factura

Nombre	Tipo	Descripción
cod_factura	Int(9)	Número correspondiente a la factura.
cod_proveedor	VarChar(10)	Rut de la empresa proveedora.
fecha_compra	Date	Fecha de compra.
iva_factura	Int(3)	IVA cobrado.
total_factura	Int(7)	Monto total de la factura.
estado_de_pago	TinyInt(1)	Indica el estado de pago de la factura. 0 = Pagada 1 = Por pagar.

2.2.2.17 Tabla Detalle factura

Nombre: Detalle_factura

Descripción: Tabla que almacena los detalles de la factura.

Clave primaria: cod_producto + cod_proveedor + cod_factura

Claves foráneas: cod_producto referencia a tabla Producto(cod_producto);
cod_proveedor referencia a tabla Proveedor(cod_proveedor);
cod_factura referencia a tabla Factura(cod_factura).

Tabla 2-17: Descripción Tabla Detalle_factura

Nombre	Tipo	Descripción
cod_producto	SmallInt(5)	Corresponde a un número correlativo que identifica un producto.
cod_factura	Int(9)	Número correspondiente a la factura.
cantidad_prod_factura	TinyInt(3)	Cantidad de unidades del producto.
precio_comprado	Int(6)	Precio por unidad del producto.
total_detalle_factura	Int(6)	Valor total de un producto

CAPÍTULO 3: DESCRIPCIÓN DE PROGRAMAS

3 DESCRIPCIÓN DE PROGRAMAS

3.1 DIAGRAMA DE MENÚS

A continuación se presentan los diagramas de menús del sistema. Se aprecian dos menús, uno para cada tipo de usuario, con sus distintas funcionalidades.

3.1.1 Menú administrativo

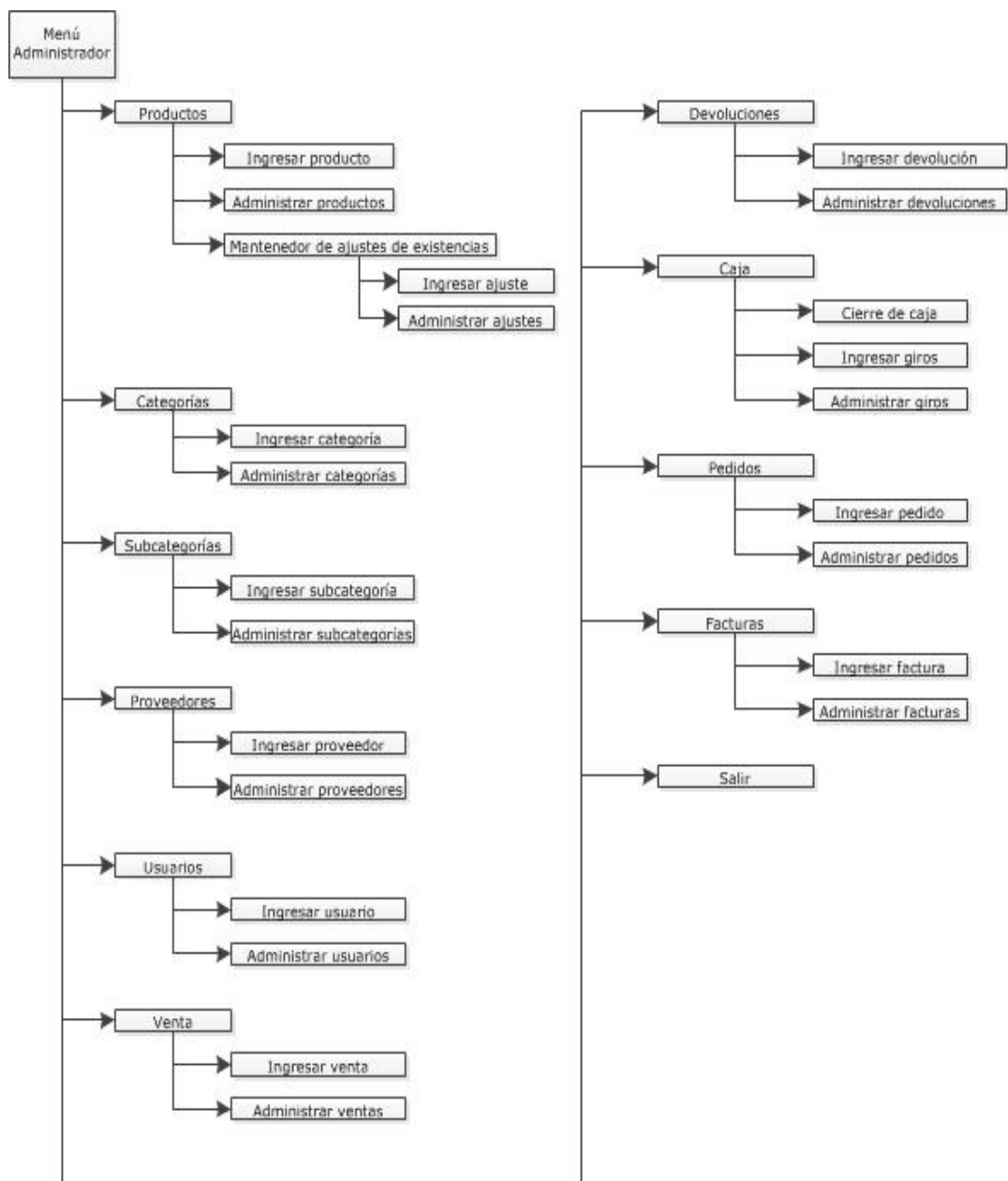


Diagrama 3-1: Diagrama de menú administrativo

Las opciones Administrar presentes en el Diagrama de menú administrativo (Diagrama 3-1) adjunta las funcionalidades de consulta, modificación y eliminación.

3.1.2 Menú vendedor

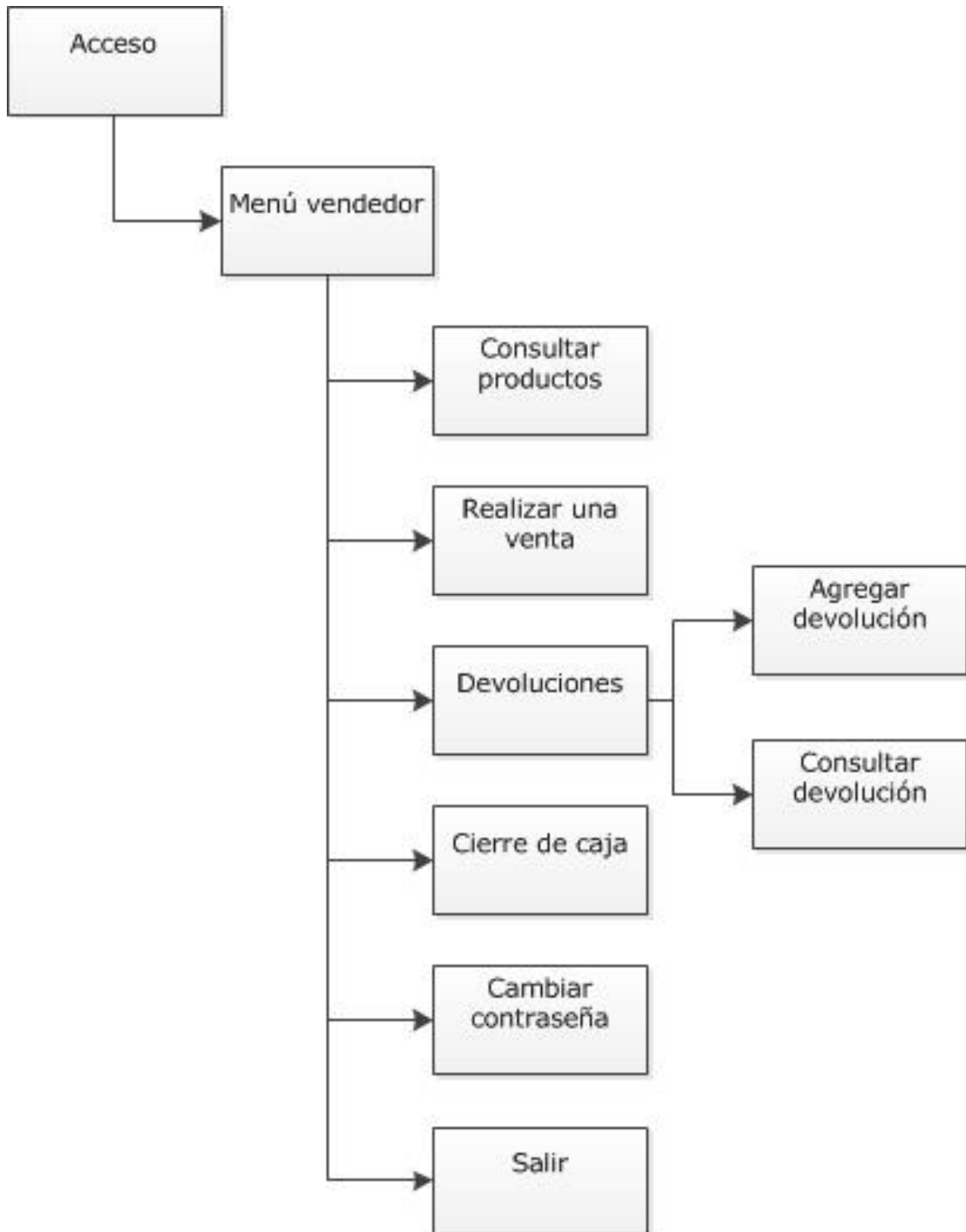


Diagrama 3-2: Diagrama de menú de vendedor

3.2 DIAGRAMA MODULAR

A continuación se presenta el diagrama modular.

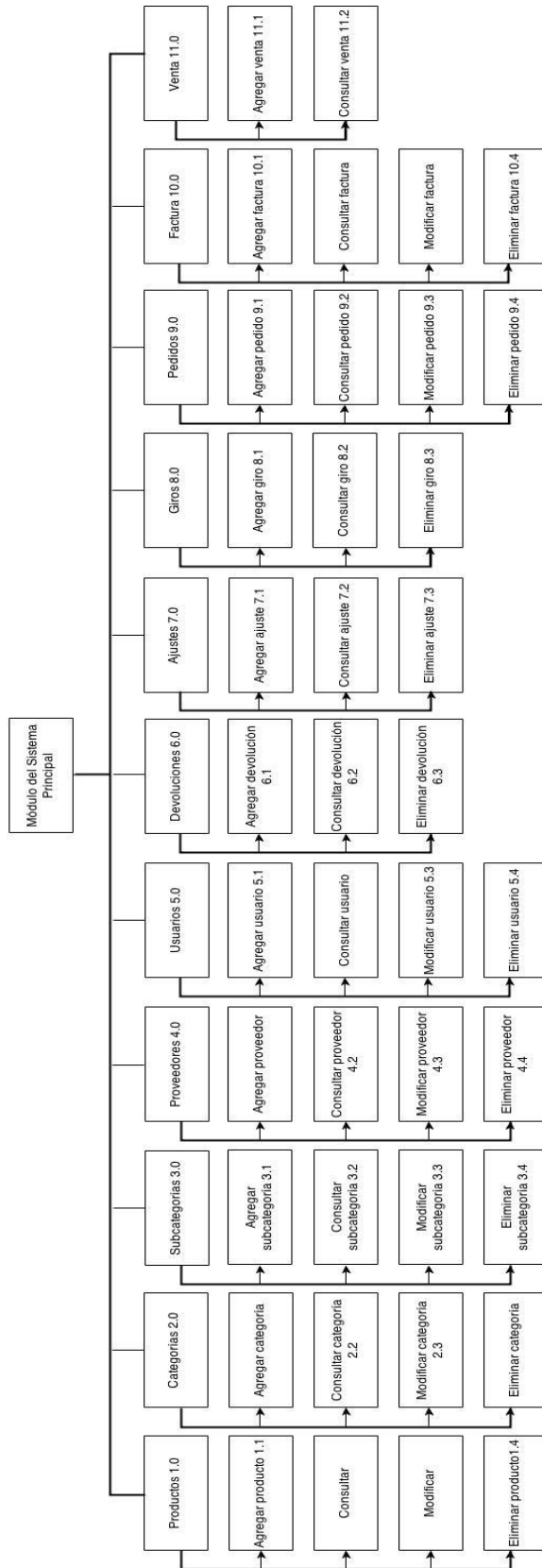


Diagrama 3-3: Diagrama modular del sistema

3.3 TABLA DE PROGRAMAS DEL SISTEMA

A continuación, se describirán los programas existentes dentro del sistema informático, con sus objetivos principales.

Tabla 3-1: Descripción de programas y objetivos del Sistema.

Nombre de Programa	Objetivo
Identificación del usuario	Restringir las distintas funcionalidades administrativas del sistema
Inicio	Permite seleccionar las distintas opciones de la estructura funcional del sistema
Ingresar producto	Agregar productos al sistema
Listar productos	Permite emitir la lista completa o filtrada de los productos en el sistema, también empleada para administrar productos
Ver proveedores	Permite generar la lista de los proveedores asociados al producto seleccionado en la pantalla de administrador de productos
Eliminar proveedores de producto	Eliminar los proveedores asociados al producto seleccionado
Ver informe de productos con bajo <i>stock</i>	Permite emitir la lista de productos con bajo stock existentes en el local
Modificar productos	Permitir la modificación de los datos de productos registrados
Eliminar productos	Eliminar un producto en el sistema, que haya sido seleccionado por el usuario al momento de listar los productos
Ingresar categoría	Agregar nuevas categorías al sistema
Listar categorías	Permite generar una lista completa, o filtrada, de las categorías en el sistema, además de ser usada para la administración de categorías
Modificar categorías	Permite la modificación de las categorías existentes por parte del usuario-administrador
Eliminar categorías	Elimina una categoría, seleccionada al momento de listarlas, si el usuario así lo requiere, siempre y cuando no esté asociada a una subcategoría
Ingresar subcategoría	Agregar nuevas subcategorías al sistema

Nombre de Programa	Objetivo
Listar subcategorías	Permite emitir una lista completa, o filtrada, las subcategorías en el sistema, además de ser usada para la administración de subcategorías
Modificar subcategorías	Permite la modificación de las subcategorías registradas
Eliminar subcategorías	Elimina una subcategoría, seleccionada por el usuario, del sistema
Ingresar proveedor	Agregar proveedores al sistema
Listar proveedores	Permite listar, o filtrar, los proveedores en el sistema, además de ser usada para la administración de éstos.
Modificar proveedores	La modificación de los proveedores registrados en el sistema
Eliminar proveedores	Elimina un proveedor del sistema, seleccionado por el usuario, siempre y cuando no esté asociado a una factura, un pedido o a un producto
Ingresar usuario	Agregar usuarios al sistema
Listar usuarios	Permite emitir la lista completa, o filtrada, los usuarios en el sistema, además de ser usada para la administración, es decir, la modificación y eliminación, de los usuarios
Modificar usuarios	La modificación de los usuarios ya agregados en el sistema
Eliminar usuarios	Elimina un usuario del sistema
Ingresar devolución	Agregar devoluciones en el sistema
Listar devoluciones	Permite la emisión de la lista completa o filtrada de devoluciones existentes en el sistema, además de ser usada para la eliminación de las devoluciones
Eliminar devoluciones	Elimina una devolución del sistema, seleccionada por el usuario administrador
Ingresar ajuste de existencias	Agregar nuevos ajustes de existencia al sistema
Listar ajustes de existencias	Permite emitir la lista completa o filtrada de los ajustes de existencias en el sistema, además de ser usada para la eliminación de algún ajuste en específico
Eliminar ajustes de existencias	Permite eliminar un ajuste de existencia en el sistema, que haya sido seleccionado en la lista generada

Nombre de Programa	Objetivo
Ingresar giro	Agregar giros al sistema
Listar giros	Permite generar la lista completa o filtrada de los giros existentes en el sistema, también empleada para la eliminación de los giros
Eliminar giros	Eliminar un giro existente en el sistema
Ingresar pedido	Agregar nuevos pedidos al sistema
Listar pedido	Permite generar una lista del completa o filtrada de los pedidos registrados en el sistema, también empleada para administrar los pedidos
Ver detalle de pedido	Visualización del detalle del pedido seleccionado
Modificar detalle de pedido	Permite la modificación del detalle del pedido seleccionado
Eliminar detalle de pedido	Elimina un detalle de pedido del sistema
Modificar pedido	Permite la modificación de los pedidos ya agregados en el sistema
Eliminar pedido	Permite eliminar un pedido del sistema
Ingresar factura	Agregar facturas al sistema
Listar facturas	Permite emitir la lista completa o filtrada de todas las facturas en el sistema, también empleada para administrar facturas
Ver detalle de factura	Permite generar una lista del detalle de la factura seleccionada al momento de emitir la lista de las facturas
Modificar detalle de factura	Permitir la modificación del detalle de la factura
Eliminar detalle de factura	Permite la eliminación de un detalle de factura del sistema
Modificar facturas	Permite la modificación de las facturas ya ingresadas en el sistema
Eliminar facturas	Permite la eliminación de una factura del sistema
Ingresar ventas	Agregar ventas al sistema
Listar ventas	Permite emitir la lista completa o filtrada de las ventas registradas en el sistema
Ver detalle de venta	Permite emitir una lista del detalle de una venta seleccionada por el usuario
Cierre de caja	Permite cerrar el día de caja
Historial de movimiento de caja	Permite la consulta del historial del movimiento de caja

3.4 **DESCRIPCIÓN DETALLADA DE PROGRAMAS**

A continuación, se presenta una descripción detallada de los programas esenciales que componen el sistema.

3.4.1 Identificación del usuario

- Nombre físico: index.php, index_proc.php
- Objetivo: Permitir el acceso a las diferentes funciones del sistema, de acuerdo al perfil que tiene el usuario
- Descripción: Permite el acceso al sistema de los usuarios previamente registrados.
 - Validaciones: El ingreso de un rut válido, y los dos campos no deben quedar vacíos.
- Diagrama de bloque:

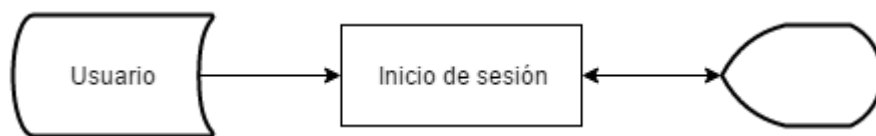


Diagrama 3-4: Diagrama identificación del usuario

- Algoritmo: Al iniciar el sistema, se presenta una pantalla en que se pedirá ingresar los campos Usuario y Contraseña (Figura 3-1). Si el usuario no está registrado (no existe el registro en la tabla *Usuario* de la base de datos), se muestra en un mensaje que informa el error y no podrá acceder al sistema; de lo contrario, dependiendo de los datos ingresados, se accederá como usuario administrador o vendedor.
- Código fuente: Anexo 1, página 89.
- Diseño de pantalla:

Ingreso al sistema

Usuario:

Contraseña:

Figura 3-1: Pantalla identificación del usuario

3.4.2 Inicio

- Nombre físico: inicio.php
- Objetivo: Permite seleccionar las distintas opciones de la estructura funcional del sistema.
- Descripción: Pantalla de inicio en que se observa, e interactúa, con el menú, dependiente del privilegio de cada usuario, el cual permite acceder a las diferentes opciones.
- Diagrama de bloque:

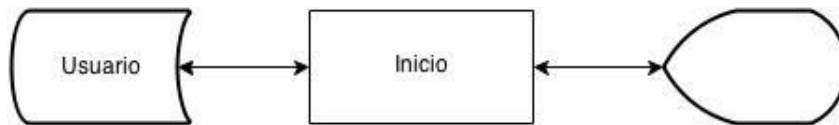


Diagrama 3-5: Diagrama inicio

- Algoritmo: Si un usuario administrador ingresa al sistema, se carga una página con las funcionalidades administrativas correspondientes, desplegadas en un menú (Figura 3-2). Si un usuario vendedor ingresa al sistema, se carga una página con un menú que contiene las funcionalidades para el vendedor (Figura 3-3).
- Código fuente: Anexo 2, página 92.
- Diseño de pantalla:

Menú administrativo:

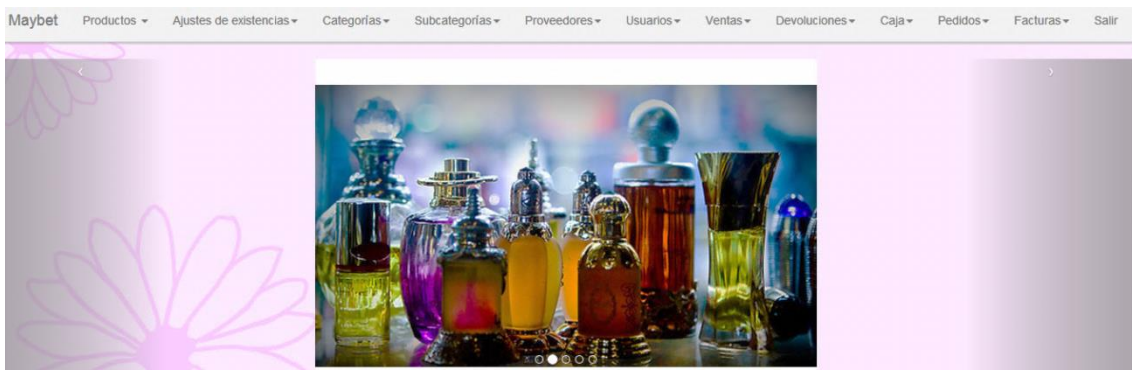


Figura 3-2: Pantalla inicio menú administrativo

Menú vendedor:

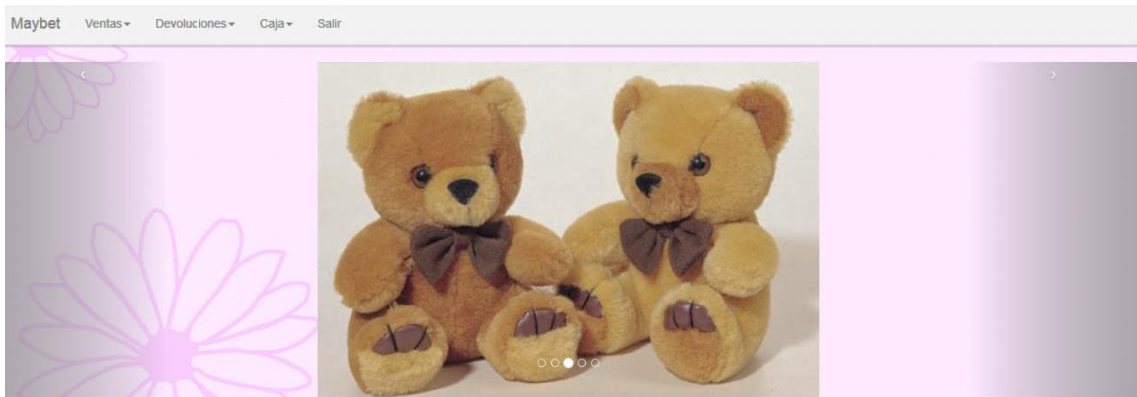


Figura 3-3: Pantalla inicio menú vendedor

3.4.3 Mantenedor de productos

Mantenedor que permite ingresar nuevos productos al sistema. Permite además, mostrar una lista con productos existentes, y en cada uno de estos últimos, mostrar opciones para ver los proveedores asociados, modificar datos y eliminar productos del sistema.

3.4.3.1 Ingresar producto

- Nombre físico: p_prod_agrega.php, prod_agrega_guardar.php
- Objetivo: Agregar productos al sistema.
- Descripción: Personas de tipo usuario administrador podrán ingresar productos al sistema, mediante una pantalla que contiene los campos necesarios para realizar la tarea.
 - Validaciones: Sólo el campo del código de barras puede ser dejado en blanco, los demás se completan obligatoriamente. El código de barras no puede existir en la base de datos, no puede superar los trece caracteres, y debe ser un dato numérico. El nombre del producto no debe superar los cuarenta caracteres. La cantidad y la cantidad crítica del producto deben ser sólo numéricas, superiores a cero; la cantidad no debe superar los cinco caracteres, a diferencia de los dos caracteres en el caso de la cantidad crítica. El precio unitario debe ser sólo numérico positivo superior a cero y no debe superar los seis caracteres.
- Diagrama de bloque:

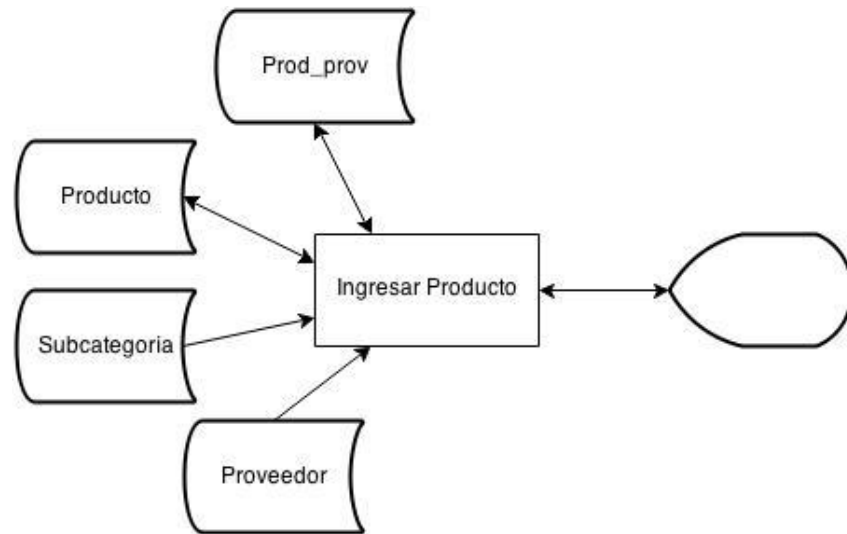


Diagrama 3-6: Diagrama de bloque ingresar producto

- Algoritmo: El usuario administrador, al ingresar sus datos correctamente, logrará apreciar en la parte superior de la pantalla, una opción de menú, que al posicionarse y entrar a *Productos*, y luego *Ingresar Producto*, se redireccionará a un formulario (Figura 3-4). Se ingresa el código de barras (no obligatoriamente), se selecciona el nombre de la subcategoría a la que pertenece el producto (lista generada con datos leídos de la tabla *Subcategoría*), el nombre del producto, la cantidad y la cantidad crítica, y el precio de venta. Si al pulsar el botón *Siguiente* hay datos no ingresados, o así mismo, mal ingresados, se apreciarán mensajes de error en cada ítem del formulario, y en donde se especifica el error cometido. Si se cumple con la validación, se redirige a la segunda pantalla (Figura 3-5), en la que se pueden agregar el proveedor asociado al producto (lista generada con datos leídos de la tabla *Proveedor*); si fuese más de uno, se debe apretar el botón *Agregar más proveedores*. Con el botón *Agregar producto* se finaliza la operación y se registra el producto en la tabla *Producto* y se registra el (los) proveedor (es) asociado (s) al producto en la tabla *Prod_prov* en la base de datos.
- Código fuente: Anexo 3.1, página 94.
- Diseño de pantalla:

The screenshot shows a web interface for adding a product. At the top, there is a navigation menu with items: Maybet, Productos, Ajustes de existencias, Categorías, Subcategorías, Proveedores, Usuarios, Ventas, Devoluciones, Caja, Pedidos, Facturas, and Salir. The main heading is 'Agrega un producto'. Below this, there are several input fields: 'Código de barra' (with a placeholder 'Ingrese el código de barra'), 'Subcategoría' (a dropdown menu currently showing 'AGENDAS'), 'Nombre' (with a placeholder 'Ingrese el nombre del producto'), 'Cantidad' (with a placeholder 'XXXX'), 'Cantidad crítica' (with a placeholder 'XX'), and 'Precio de venta' (with a placeholder '\$'). A green button labeled 'Siguiente' is positioned at the bottom left of the form area.

Figura 3-4: Pantalla número uno de ingreso de un producto

The screenshot shows a web interface for product entry. At the top, there is a navigation menu with items: Maybet, Productos, Ajustes de existencias, Categorías, Subcategorías, Proveedores, Usuarios, Ventas, Devoluciones, Caja, Pedidos, Facturas, and Salir. The main heading is 'Ingreso de productos', with a sub-heading 'Ingreso de proveedores'. Below this, there are two input fields: 'Nombre del producto' (with the value 'Estuche') and 'Proveedor' (a dropdown menu showing '1'). At the bottom left, there are two buttons: a blue button with a plus sign and the text 'Agrega más proveedores', and a green button with a checkmark and the text 'Agregar pedido'.

Figura 3-5: Pantalla número dos de ingreso de un producto

3.4.3.2 Listar productos

- Nombre físico: prod_admin2.php
- Objetivo: Permite emitir la lista completa o filtrada de los productos en el sistema, también empleada para administrar productos.
- Descripción: El usuario puede buscar o filtrar productos por código de barra, por nombre del producto, por categoría o por subcategoría, además de generar un informe de los productos con *stock* crítico.
- Diagrama de bloque:

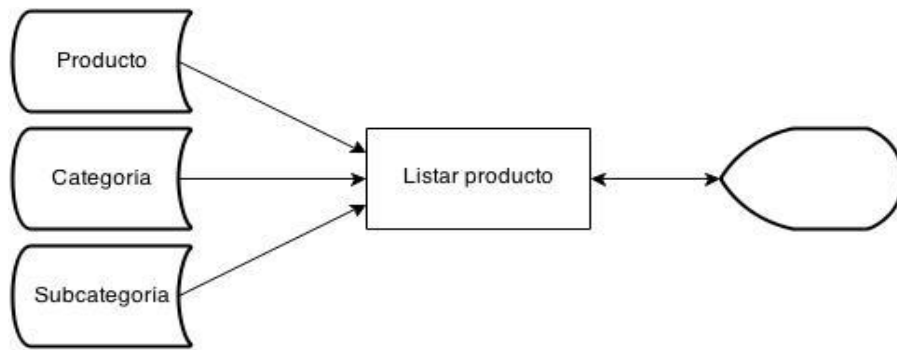


Diagrama 3-7: Diagrama de bloque listar producto

- Algoritmo: El usuario administrador al acceder a *Productos* y luego *Administrar Productos*, opciones disponibles en la barra de menú situada en el extremo superior de la pantalla, visualizará la pantalla (Figura 3-6) que permitirá generar una lista de los productos disponibles en la *tabla Producto*, los datos son extraídos de las tablas *Producto*, *Categoría* y *Subcategoría*. En cada línea de producto se mostrarán tres botones: *Proveedor* (Punto 3.4.3.2.1), *Modificar* (Punto 3.4.3.3) y por último, *Eliminar* (Punto 3.4.3.4). También se puede observar un botón que tiene una funcionalidad de listar los productos con bajo *stock* registrados en el sistema (Punto 3.4.3.2.3)
- Código fuente: Anexo 3.2, página 98.
- Diseño de pantalla:

Maybet Productos ▾ Ajustes de existencias ▾ Categorías ▾ Subcategorías ▾ Proveedores ▾ Usuarios ▾ Ventas ▾ Devoluciones ▾ Caja ▾ Pedidos ▾ Facturas ▾ Salir

Administrador de productos

Productos con bajo stock

Buscador:
 1) Por Código de barra
 2) Por nombre del producto
 3) Por categoría
 4) Por subcategoría
 Presione espacio para mostrar la lista completa

Código	Código barra	Categoría	Subcategoría	Nombre	Cantidad	Cantidad crítica	Precio	Administración
61	1234	LIBRERIA	AGENDAS	aa	61	1	1000	Proveedores Modificar Eliminar
32	4444444	LIBRERIA	AGENDAS	AGENDA 2014	0	3	5000	Proveedores Modificar Eliminar
40	1	PERFUMERIA	PERFUMES	AGUA BRAVA	1	4	6000	Proveedores Modificar Eliminar
39	77	LIBRERIA	AGENDAS	CHANNEL NRO 5	25	1	40000	Proveedores Modificar Eliminar
17	97989786	PERFUMERIA	PERFUMES	CORAL	6	2	3000	Proveedores Modificar Eliminar
57	8	LIBRERIA	GOMAS	DE PAN	18	4	200	Proveedores Modificar Eliminar
54	0	LIBRERIA	GOMAS	Gomita	-6	2	2000	Proveedores Modificar Eliminar
18	9887776	LIBRERIA	AGENDAS	PASCUALINA	7	3	3	Proveedores Modificar Eliminar

Figura 3-6: Pantalla de lista y administración de productos

3.4.3.2.1 Ver proveedores

- Nombre físico: prod_det.php
- Objetivo: Permite generar la lista de los proveedores asociados al producto seleccionado en la pantalla de administrador de productos (Figura 3-6).
- Descripción: El usuario al seleccionar la opción *Proveedores* puede visualizar los proveedores de cada producto.
- Diagrama de bloque:

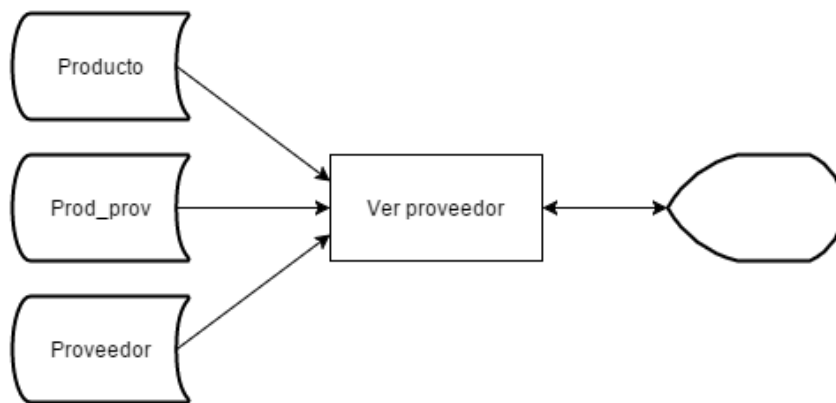


Diagrama 3-8: Diagrama de bloque ver proveedores de producto

- Algoritmo: Si se accede a *Productos*, luego *Administrar Productos*, se generará una lista con productos en el sistema. Finalmente, si se presiona el botón *Proveedor*, en un producto de la lista, se apreciará una pantalla (Figura 3-7) en donde se podrá ver los proveedores del producto, datos extraídos de las tablas *Prod_prov*, *Producto* y *Proveedor*. Además, si fuese necesario, se podrán eliminar proveedores asociados (Punto 3.4.3.2.2).
- Código fuente: Anexo 3.2.2, página 100.
- Diseño de pantalla:

Maybet Productos - Ajustes de existencias - Categorías - Subcategorías - Proveedores - Usuarios - Ventas - Devoluciones - Caja - Pedidos - Facturas - Salir				
Detalle del producto Proveedores				
Código del producto	Nombre del producto	Código del proveedor	Nombre del proveedor	Administración
988776	PASCUALINA	167593674	Empresa N	Eliminar

Figura 3-7: Pantalla de proveedores de cada producto

3.4.3.2.2 Eliminar proveedores de producto

- Nombre físico: prod_det_elim.php
- Objetivo: Eliminar los proveedores asociados al producto seleccionado.
- Descripción: El usuario administrador, al seleccionar la opción *Eliminar* (Figura 3-7) puede eliminar un proveedor de la lista de un producto seleccionado con anterioridad (Punto 3.4.3.2).
- Diagrama de bloque:

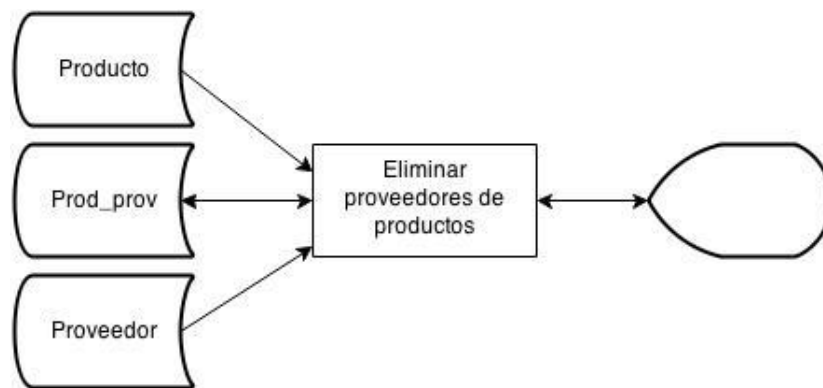


Diagrama 3-9: Diagrama de bloque eliminar proveedores de producto

- Algoritmo: En la lista de proveedores de productos anteriormente mencionada (Punto 3.3.3.2.1) que se genera con datos de las tablas *Producto*, *Prod_prov* y *Proveedor*, si el usuario administrador selecciona el botón *Eliminar*, se desplegará un mensaje que pedirá la confirmación de dicha acción. Si la respuesta del usuario es verdadera, se logrará eliminar proveedores, del producto seleccionado, satisfactoriamente. Esta eliminación lógica se lleva a cabo cambiando el estado del campo *vigencia_prov* de la tabla *Prod_prov*.
- Código fuente: Anexo 3.2.3, página 101.

3.4.3.2.3 Ver informe de productos con bajo Stock

- Nombre físico: bajoStock.php
- Objetivo: Permite emitir la lista de productos con bajo *stock* existentes en el local.
- Descripción: El usuario, al seleccionar el botón *Productos con bajo stock*, puede observar un informe con los productos que cumplen ese requisito.
- Diagrama de bloque:

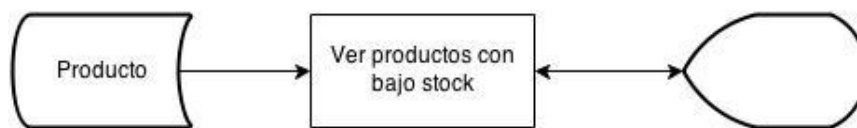


Diagrama 3-10: Diagrama de bloque ver productos con bajo stock

- Algoritmo: En el menú administrativo se debe seleccionar la opción *Productos*, luego *Administrar Productos*. Se generará una pantalla con una lista con productos en el sistema y con un botón *Productos con bajo stock* en la parte superior. Si se presiona el botón, se apreciará una pantalla (Figura 3-8) en que se listarán sólo los productos que estén con la cantidad de existencia baja, es decir, sólo se muestran los datos que, al leer la tabla *Producto*, el campo *cantidad_critica_prod* sea menor a *cantidad_producto*.
- Código fuente: Anexo 3.2.4, página 101.
- Diseño de pantalla:

Productos con bajo Stock		
Volver Atrás		
Código	Nombre	Cantidad disponible
32	AGENDA 2014	0
40	AGUA BRAVA	1
54	Gomita	-6

Figura 3-8: Pantalla de informe de *stock* crítico

3.4.3.3 Modificar productos

- Nombre físico: `prod_mod.php`, `prod_mod_guardar.php`
- Objetivo: Permitir la modificación de los datos de productos registrados.
- Descripción: El usuario podrá modificar un producto existente en el sistema.
- Diagrama de bloque:

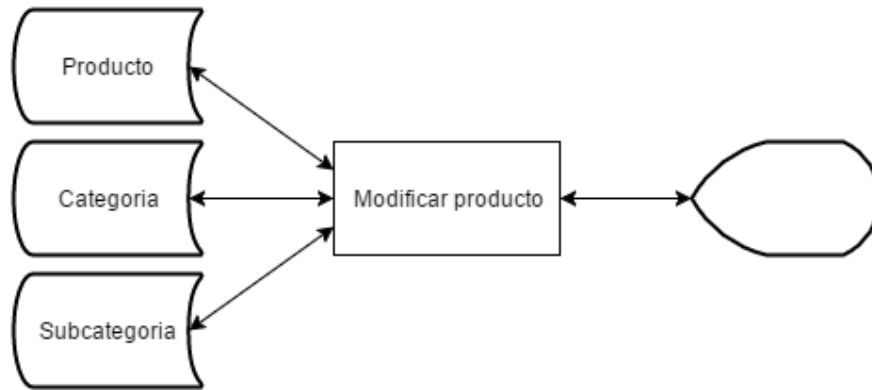


Diagrama 3-11: Diagrama de bloque modificar producto

- Algoritmo: El usuario administrador al listar los productos (Punto 3.4.3.2) con los datos leídos de las tablas *Producto*, *Categoria* y *Subcategoria*, y seleccionar el botón *Modificar* de un producto en particular (Figura 3-6), permitirá que se dirija a un formulario (Figura 3-9) en el que podrá cambiar los datos del producto seleccionado, menos su código único y lógico que lo identifica, su código de barra, o como también así, su cantidad, ya que esta última se realiza por ajuste de existencia (Punto 3.4.6). Al seleccionar el botón *Modificar*, que se encuentra al final del formulario, se actualizarán la tablas *Producto*, *Categoria* y *Subcategoria* con los datos modificados.
- Código fuente: Anexo 3.3, página 103.
- Diseño de pantalla:

Figura 3-9: Pantalla de modificación de un producto

3.4.3.4 Eliminar productos

- Nombre físico: prod_elim.php
- Objetivo: Eliminar un producto en el sistema, que haya sido seleccionado por el usuario al momento de listar los productos (Punto 3.3.3.2).
- Descripción: Eliminar un artículo será posible siempre y cuando no esté asociado tanto a una venta en proceso como a una realizada.
- Diagrama de bloque:

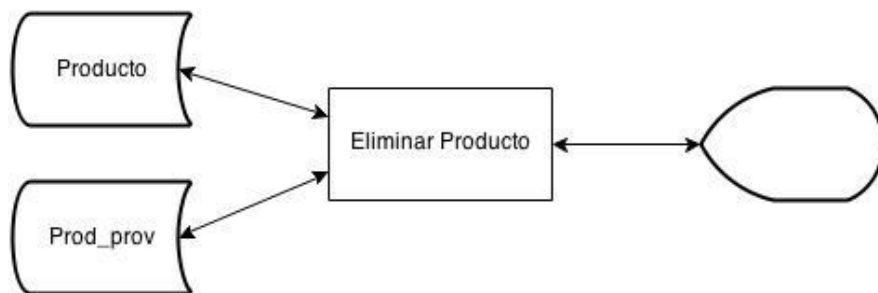


Diagrama 3-12: Diagrama de bloque eliminar producto

- Algoritmo: Si se presiona el botón *Eliminar*, sobre la sección de Administración de la tabla, generada al listar los productos (Figura 3-6), el usuario recibe un mensaje de confirmación de la operación. Se elimina físicamente de la tabla *Producto* si el producto no está asociado a una venta en proceso o realizada, datos que se encuentran en la tabla *Prod_prov*, de lo contrario se oculta de la lista de los productos (eliminación lógica).
- Código fuente: Anexo 3.4, página 106.

3.4.4 Mantenedor de subcategorías

Permite crear nuevas subcategorías al sistema. Otras funcionalidades son listar, modificar y eliminar subcategorías.

3.4.4.1 Ingresar subcategorías

- Nombre físico: p_subcat_agrega.php, subcat_agrega_guardar.php
- Objetivo: Agregar nuevas subcategorías al sistema.
- Descripción: Sólo el usuario administrador podrá ingresar nuevas subcategorías, mediante el ingreso de datos en un formulario.
 - Validaciones: El campo del nombre de la nueva subcategoría no puede estar vacío y no debe superar los treinta caracteres.
- Diagrama de bloques:

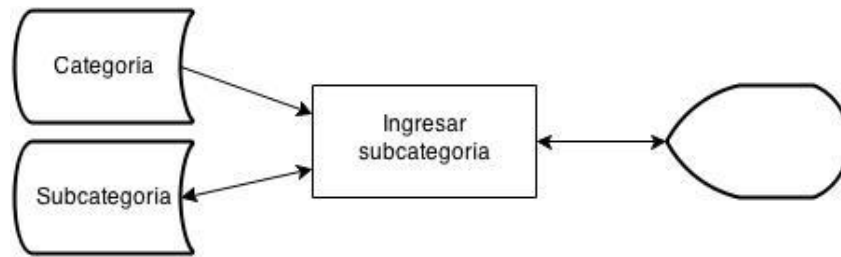


Diagrama 3-13: Diagrama de bloque ingresar subcategoría

- Algoritmo: Si la sesión se inicia como usuario administrador, y si en la barra de menú se selecciona *Subcategorías*, y luego *Ingresar Subcategoría*, se verá un formulario (Figura 3-10) en el que puede ingresar el nombre de una nueva subcategoría. Además se debe seleccionar a una categoría, ya registrada previamente en la base de datos, específicamente en la tabla *Categoria*, a la cual pertenece. Al seleccionar el botón *Ingresar*, se verifica que los datos estén ingresados correctamente, si es así, se ingresa la nueva subcategoría a la tabla *Subcategoría*, disponible en la base de datos.
- Código fuente: Anexo 4.1, página 106.
- Diseño de pantalla:

Figura 3-10: Pantalla de ingreso de una subcategoría

3.4.4.2 Listar subcategorías

- Nombre físico: `subcat_admin2.php`
- Objetivo: Permite emitir una lista completa, o filtrada, las subcategorías en el sistema, además de ser usada para la administración de subcategorías.
- Descripción: Un usuario administrador puede buscar o filtrar subcategorías por su nombre.

- Diagrama de bloque:

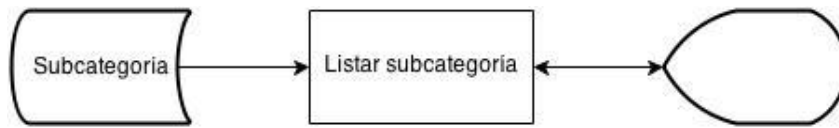


Diagrama 3-14: Diagrama de bloque listar subcategorías

- Algoritmo: En la barra de menú se selecciona *Subcategorías* y luego *Administrar Subcategorías*. La pantalla (Figura 3-11) mostrará una lista, ordenada por código, con los nombres de las subcategorías que están registradas en la *tabla Subcategoria* y cada una tendrá dos botones de administración: *Modificar* (Punto 3.4.4.3) y *Eliminar* (Punto 3.4.4.4).
- Código fuente: Anexo 4.2, página 109.
- Diseño de pantalla:

Código	Nombre subcategoria	Administración
1	AGENDAS	Modificar Eliminar
2	CUADERNOS	Modificar Eliminar
3	CREMAS	Modificar Eliminar
4	PERFUMES	Modificar Eliminar
5	LANAS	Modificar Eliminar
6	HILOS	Modificar Eliminar

Figura 3-11: Pantalla de lista y administración de una subcategoría

3.4.4.3 Modificar subcategorías

- Nombre físico: `subcat_mod.php`, `subcat_mod_guardar.php`
- Objetivo: Permite la modificación de las subcategorías registradas.
- Descripción: Se logrará modificar subcategorías existente en el sistema por parte del usuario administrador.
- Diagrama de bloques:

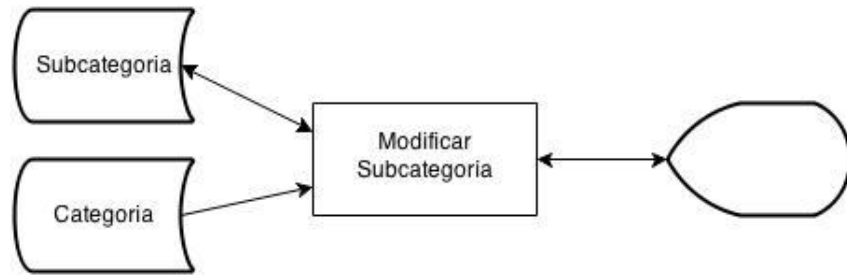


Diagrama 3-15: Diagrama de bloque modificar subcategoría

- Algoritmo: Luego de listar (Punto 3.4.4.2) extrayendo los datos de las tablas *Subcategoría* y *Categoría*, y seleccionar el botón *Modificar* de alguna subcategoría, la pantalla que le procede (Figura 3-12) presenta un formulario con los campos Código de la subcategoría, Categoría a la que pertenece, Categoría nueva y Nombre, dando opción de modificación sólo a las dos últimas. Al seleccionar el botón *Modificar*, que se encuentra al final del formulario, se permitirá realizar exitosamente la modificación en la tabla *Subcategoría* en la base de datos.
- Código fuente: Anexo 4.3, página 111.
- Diseño de pantalla:

Figura 3-12: Pantalla de modificación de una subcategoría

3.4.4.4 Eliminar subcategorías

- Nombre físico: subcat_elim.php
- Objetivo: Elimina una subcategoría, seleccionada por el usuario, del sistema.
- Descripción: Si el usuario administrador desea eliminar una subcategoría, seleccionada al momento de listar las subcategorías (Punto 3.4.4.2), realizará la acción exitosamente, siempre y cuando no esté asociada a un producto.

- Diagrama de bloque:

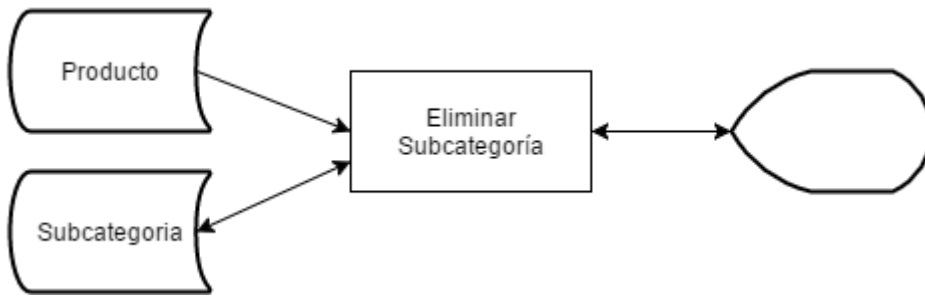


Diagrama 3-16: Diagrama de bloque eliminar categoría

- Algoritmo: Al listar las subcategorías, y presionar el botón *Eliminar* (Figura 3-11) en una de ellas, el usuario administrador recibe un mensaje de confirmación de la operación. Si el usuario acepta la operación de eliminación, el paso siguiente que se efectúa, es la validación de que la subcategoría que desea eliminar no se encuentre ligada a un producto, esto se realiza extrayendo datos de las tablas *Producto* y *Subcategoría*, si es correcta, se elimina físicamente, modificando la tabla *Subcategoría*. Si la validación es falsa, se genera un mensaje de error explicando que no se puede eliminar la subcategoría.
- Código fuente: Anexo 4.4, página 113.

3.4.5 Mantenedor de usuarios

Mantenedor que permite ingresar nuevos usuarios al sistema. Otras funcionalidades son listar, modificar y eliminar usuarios.

3.4.5.1 Ingresar usuarios

- Nombre físico: p_usu_agrega.php, usu_agrega_guardar.php
- Objetivo: Agregar usuarios al sistema.
- Descripción: Sólo usuarios administradores podrán crear usuarios al sistema, mediante un formulario.
 - Validaciones: Todos los campos son requeridos. El rut del usuario debe ser válido, no debe superar los nueve caracteres. El nombre no debe superar los dos caracteres, la contraseña no debe superar diez, y así mismo, la dirección no debe superar los cuarenta caracteres. Los datos que se ingresan en el campo del teléfono del usuario debe ser sólo de tipo numérico y no debe superar los doce caracteres.

- Diagrama de bloques:

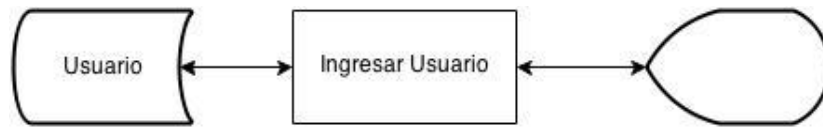


Diagrama 3-17: Diagrama de bloque ingresar usuario

- Algoritmo: Sólo usuarios que hayan iniciado sesión con cuenta tipo administrador pueden ingresar nuevos usuarios. En el menú de administrador se ingresa a Usuarios y posteriormente Ingresar usuario. Se deben completar los campos del formulario (Figura 3-13) Rut de usuario, Nombre, Contraseña, Dirección, Teléfono, y Tipo de usuario. Posteriormente, si se pulsa el botón Agregar que está ubicado al final del formulario, se permitirá agregar exitosamente, a la tabla *Usuario*, sólo si la validación de los datos ingresados es exitosa.
- Código fuente: 5.1, página 113.
- Diseño de pantalla:

Figura 3-13: Pantalla de ingreso de un usuario

3.4.5.2 Listar usuarios

- Nombre físico: usu_admin2.php
- Objetivo: Permite emitir la lista completa, o filtrada, los usuarios en el sistema, además de ser usada para la administración, es decir, la modificación y eliminación, de los usuarios.

- Descripción: En la lista de usuarios, los administradores en el sistema pueden, además de ver todos los usuarios registrados en el sistema, buscar o filtrar usuarios por Rut o por nombre, y además la administración de éstos (modificación y eliminación).
- Diagrama de bloque:



Diagrama 3-18: Diagrama de bloque listar usuarios

- Algoritmo: En el menú se selecciona *Usuarios* y luego *Administrar usuarios*. La siguiente pantalla que se visualiza, es una lista de los usuarios registrados en la base de datos, en la tabla *Usuario* (Figura 3-14). La lista puede ser completa o filtrada por rut o nombre del usuario. En la lista se muestran datos como el código de usuario (rut), el tipo de usuario, el nombre, la dirección y el teléfono de contacto, por seguridad la contraseña queda oculta, además de dos botones: *Modificar* (Punto 3.4.5.3) y *Eliminar* (Punto 3.4.5.4)
- Código fuente: 5.2, página 118.
- Diseño de pantalla:

Código	Tipo de usuario	Nombre	Dirección	Teléfono	
111	VEN	Vendedor	Local	00	Modificar Eliminar
123	ADM	Administrador	Local	0	Modificar Eliminar

Figura 3-14: Pantalla de lista y administración de usuarios

3.4.5.3 Modificar usuarios

- Nombre físico: usu_mod.php, usu_mod_guardar.php
- Objetivo: La modificación de los usuarios ya agregados en el sistema.
- Descripción: El usuario administrador será capaz de modificar algún usuario existente en el sistema.

- Diagrama de bloques:



Diagrama 3-19: Diagrama de bloque modificar usuario

- Algoritmo: Previamente se deben listar los usuarios (Punto 3.4.5.2) y elegir un usuario en específico para apretar su botón *Modificar*, a continuación, en la pantalla (Figura 3-15) se observa un formulario con los campos Rut del usuario, Nombre, Contraseña, Dirección, Teléfono, y Tipo de usuario, ya sea Administrador o Vendedor. Solamente no es posible cambiar el campo del rut del usuario. Al seleccionar el botón *Modificar* que se encuentra al final del formulario, permitirá realizar exitosamente la modificación en la tabla *Usuario*.
- Código fuente: 5.3, página 120.
- Diseño de pantalla:

Figura 3-15: Pantalla de modificación de usuarios

3.4.5.4 Eliminar usuarios

- Nombre físico: usu_elim.php
- Objetivo: Elimina un usuario del sistema.

- Descripción: Si un usuario administrador desea eliminar un usuario, realizará la acción exitosamente, siempre y cuando no esté asociado a un giro, ajuste, pedido, a la caja, a una venta o a una devolución, o que no se elimine mientras la sesión está siendo utilizada.
- Diagrama de bloque:

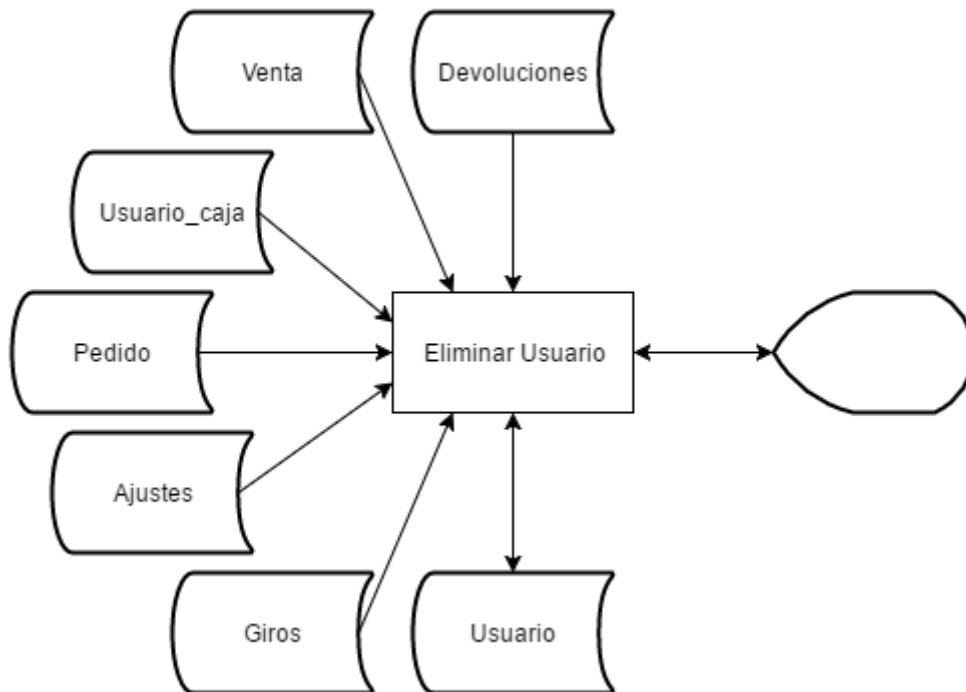


Diagrama 3-20: Diagrama de bloque eliminar usuario

- Algoritmo: Un usuario administrador debe apretar el botón *Eliminar* que se encuentra en la pantalla en que se listan los usuarios (Figura 3-14) en un usuario en específico, luego de esto, se visualiza un mensaje de confirmación de la operación. Si el usuario acepta la operación de eliminación, se realiza la validación de que el usuario a eliminar no esté asociado a un giro, ajuste, pedido, a la caja, a una venta o a una devolución, o que no se *autoelimine* mientras la sesión está siendo utilizada. Para lo anterior, se extraen datos de las tablas *Usuario*, *Giros*, *Ajustes*, *Pedido*, *Usuario_caja*, *Venta*, *Giros*, *Ajustes*, *Pedido*, *Usuario_caja*, *Venta* y *Devoluciones*. Si es correcta la validación, se elimina físicamente, de lo contrario, se genera un mensaje de error explicando que no se puede eliminar el usuario.
- Código fuente: 5.4, página 122.

3.4.6 Mantenedor ajustes de existencias

Permite agregar nuevos ajustes a las existencias del sistema. Además de listar, modificar y eliminar ajustes.

3.4.6.1 Ingresar ajustes de existencias

- Nombre físico: p_aj_agrega.php, aj_agrega_guardar.php
- Objetivo: Agregar nuevos ajustes de existencia al sistema.
- Descripción: Se podrán ingresar ajustes de existencias mediante un formulario mostrado en pantalla.
 - Validaciones: Todos los campos son requeridos. La cantidad del ajuste debe ser sólo numérico positivo y no debe superar los tres caracteres. El comentario de ajuste no debe superar los cuarenta caracteres.
- Diagrama de bloques:

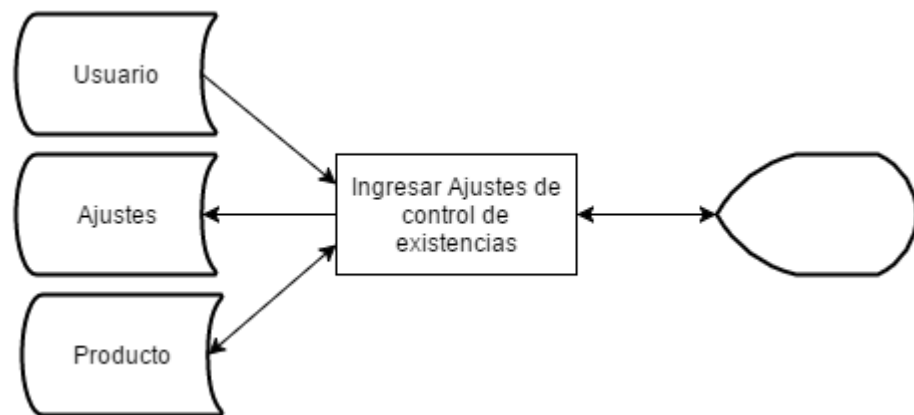


Diagrama 3-21: Diagrama de bloque ingresar ajustes de existencias

- Algoritmo: Luego de extraer los datos de las tablas *Usuario* y *Producto*, el usuario administrador, al entrar al menú *Ajustes de existencias*, y luego *Ingresar ajuste*, observará un formulario (Figura 3-16) en que se ingresa el producto al cual se le hará el ajuste, la cantidad, si el ajuste es positivo o negativo (merma) y un campo comentario para escribir algún motivo de ajuste. El registro se realizará si se valida correctamente el ingreso, acción que se permite realizar al pulsar el botón *Agregar*, si no, se le informa al usuario mediante un mensaje de error, para que corrija los datos erróneos. Lo ingresado se agrega a la tabla *Ajustes*, y se modifica el campo *cantidad_producto* de la tabla *Producto*.
- Código fuente: Anexo 6.1, página 123.

- Diseño de pantalla:

Figura 3-16: Pantalla de ingreso de ajustes de existencias

3.4.6.2 Listar ajustes de existencias

- Nombre físico: aj_admin2.php
- Objetivo: Permite emitir la lista completa o filtrada de los ajustes de existencias en el sistema, además de ser usada para la eliminación de algún ajuste en específico.
- Descripción: Un usuario administrador puede mostrar todos los ajustes realizados o filtrar por fecha del ajuste.
- Diagrama de bloques:

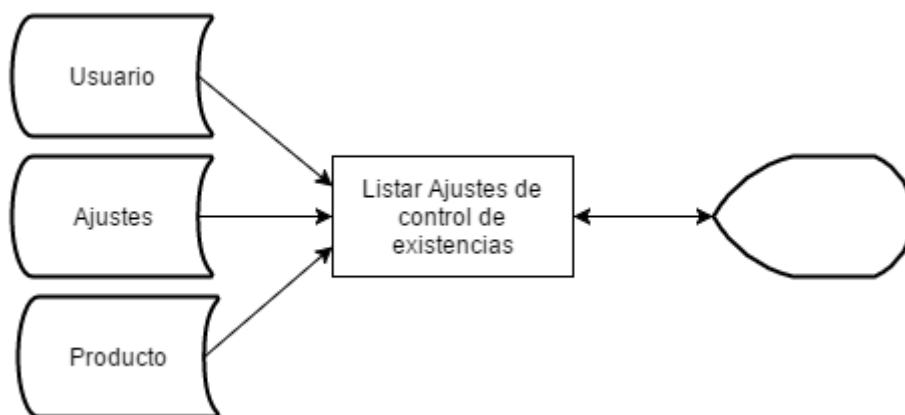


Diagrama 3-22: Diagrama de bloque listar ajustes de existencias

- Algoritmo: En la barra de menú se selecciona *Ajustes de existencias* y luego *Administrar ajustes*. La pantalla (Figura 3-17) mostrará una lista de los ajustes de existencias registrados en tabla *Ajustes*, y para obtener más detalles,

también debe leer datos de las tablas *Usuario* y *Producto*, y cada uno contendrá un botón para la eliminación (Punto 3.4.6.3), la modificación no es necesaria porque si el ajuste sufre de algún cambio, simplemente se ingresa un nuevo ajuste de existencias (Punto 3.4.6.1).

- Código fuente: Anexo 6.2, página 126.
- Diseño de pantalla:

Codigo del producto	Nombre del producto	Nombre del usuario	Fecha	Cantidad	Tipo de ajuste	Comentario	Administración
988776	PASCUALINA	Vendedor	2013-12-11	30	Positivo	Ingreso de 30	Eliminar
97989786	CORAL	Administrador	2013-12-15	2	Negativo	Regalo a familiar	Eliminar

Figura 3-17: Pantalla de lista y administración de ajustes de existencias

3.4.6.3 Eliminar ajustes de existencias

- Nombre físico: aj_elim.php
- Objetivo: Permite eliminar un ajuste de existencia en el sistema, que haya sido seleccionado en la lista generada en el punto 3.4.6.2.
- Descripción: Eliminar un ajuste de existencia seleccionado al momento de listar los ajustes si el usuario así lo requiere.
- Diagrama de bloque:

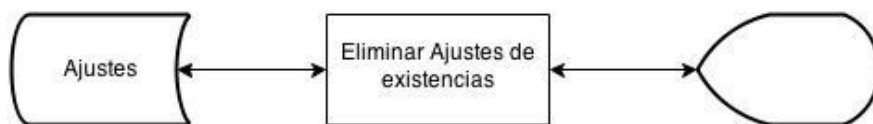


Diagrama 3-23: Diagrama de bloque eliminar ajustes de existencia

- Algoritmo: Si se presiona el botón *Eliminar* (Figura 3-17), de un registro ajuste de existencia seleccionado, el usuario deberá confirmar que desea borrar el ajuste en un mensaje que se genera. Si el usuario acepta la eliminación, se procede a eliminar físicamente el ajuste.
- Código fuente: Anexo 6.3, página 128.

3.4.7 Mantenedor de factura

Mantenedor que permite agregar nuevas facturas al sistema. Además de listar, modificar y eliminar facturas.

3.4.7.1 Ingresar facturas

- Nombre físico: p_fact_agrega.php, fact_agrega_guardar.php
- Objetivo: Agregar facturas al sistema.
- Descripción: Función restringida para usuarios administradores, podrán ingresar facturas al sistema, mediante un formulario que contiene los campos para el ingreso de datos y realizar la tarea.
 - Validaciones: Todos los campos son requeridos. Los datos en el campo del IVA deben ser numéricos positivos y con un máximo de tres caracteres. El total de la factura también debe ser numérico positivo, pero con un máximo de siete caracteres.
- Diagrama de bloque:

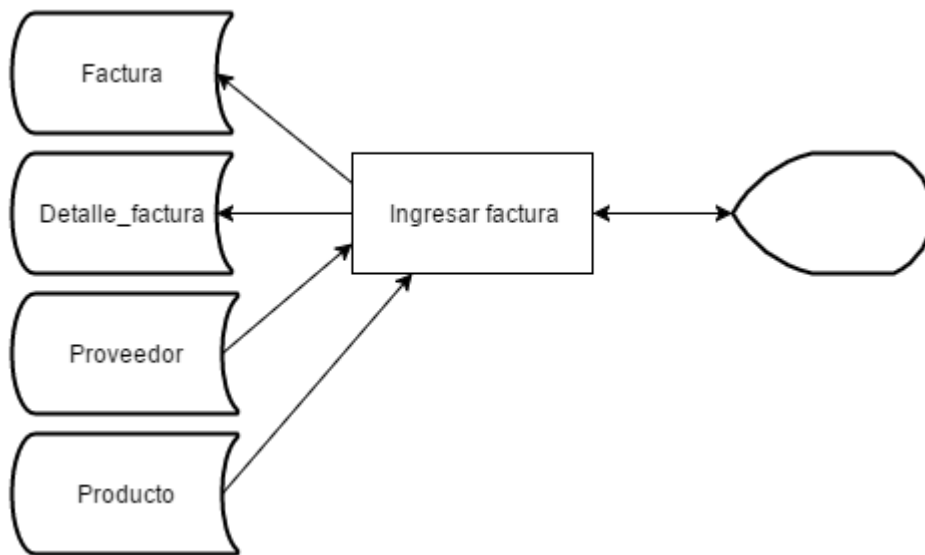


Diagrama 3-24: Diagrama de bloque ingresar factura

- Algoritmo: El usuario administrador, al ingresar al menú y entrar a *Facturas*, y luego *Ingresar Factura*, visualizará un formulario (Figura 3-18) en que se permite ingresar datos, los cuales son Código de proveedor extraído de tabla *Proveedor*, en que se selecciona el nombre del proveedor, IVA, Total factura, que son cajas de texto para que el usuario complete con datos, y Estado de pago, en que el usuario debe seleccionar una de las dos opciones, pagada o

debe. Se mostrarán mensajes de error si es que al momento de apretar el botón *Siguiente*, hay datos no ingresados, o así mismo, mal ingresados. Si la validación se cumple, es decir, los datos son correctos, se redirige a la segunda pantalla (Figura 3-19), en la que se muestra el número de la factura que se está efectuando, y se agrega el detalle de la factura, ingresando producto (selección de datos en tabla *Producto*), la cantidad y el precio por unidad. Para ingresar más productos, se debe apretar el botón *Agregar más productos*. Al pulsar el botón *Agregar factura* se permite el fin de operación y se registra la factura en la base de datos, específicamente en las tablas *Factura* y *Detalle_factura*.

- Código fuente: Anexo 7.1, página 128.
- Diseño de pantalla:

Figura 3-18: Pantalla número uno de ingreso de una factura

Figura 3-19: Pantalla número dos de ingreso de una factura

3.4.7.2 Listar facturas

- Nombre físico: fact_admin2.php

- Objetivo: Permite emitir la lista completa o filtrada de todas las facturas en el sistema, también empleada para administrar facturas.
- Descripción: El usuario puede buscar o filtrar facturas por número de factura o por fecha de compra.
- Diagrama de bloque:



Diagrama 3-25: Diagrama de bloque listar factura

- Algoritmo: El usuario administrador debe posicionarse en el menú y acceder a *Facturas* y luego *Administrar facturas*. La siguiente pantalla (Figura 3-20) permitirá generar una lista de las facturas disponibles en la tabla *Facturas*, además se pueden filtrar los resultados. En cada registro de factura, se observarán tres botones: *Detalle* (Punto 3.4.7.2.1), *Modificar* (Punto 3.4.7.3) y por último, *Eliminar* (Punto 3.4.7.4)
- Código fuente: Anexo 7.2, página 132.
- Diseño de pantalla:

Maybet Productos - Ajustes de existencias - Categorías - Subcategorías - Proveedores - Usuarios - Ventas - Devoluciones - Caja - Pedidos - Facturas - Salir

Administrador de facturas

Buscador:
 1) Por N° de Factura
 2) Por fecha de compra (yyyy-mm-dd)

Presione espacio para mostrar la lista completa

Código	Proveedor	Fecha de compra	IVA	Total	Estado de pago	Acción
10	174731330	2013-12-15	0	0	1	Detalle Modificar Eliminar
11	1234567	2013-12-15	10	30000	0	Detalle Modificar Eliminar
12	1234567	2013-12-15	10	1313	0	Detalle Modificar Eliminar

Figura 3-20: Pantalla de lista y administración de facturas

3.4.7.2.1 Ver detalle de factura

- Nombre físico: fact_det.php
- Objetivo: Permite generar una lista del detalle de la factura seleccionada al momento de emitir la lista de las facturas (Punto 3.4.7.2).
- Descripción: El usuario al seleccionar la opción *Detalle* (Figura 3-20) puede visualizar el detalle de cada factura.

- Diagrama de bloque:

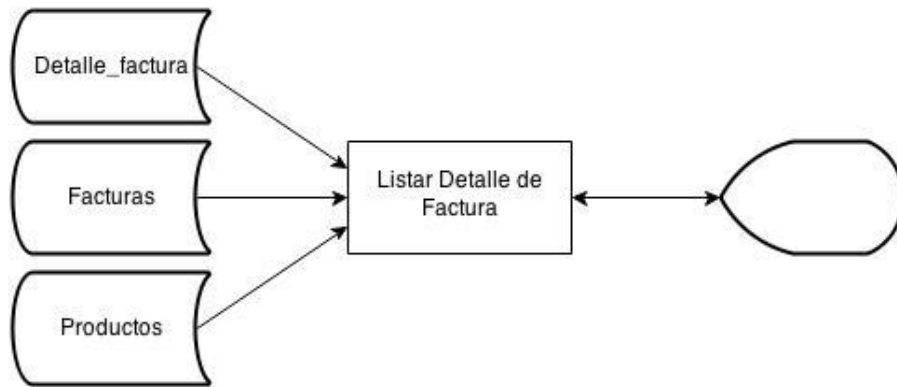


Diagrama 3-26: Diagrama de bloque ver detalle de factura

- Algoritmo: Si se accede a *Facturas*, luego *Administrar facturas*, se generará una lista con las facturas en el sistema, con datos extraídos de las tablas *Facturas*, *Detalle_factura* y *Producto*. Finalmente, si se presiona el botón *Detalle* en una factura de la lista, se generará una pantalla (Figura 3-21) en donde se podrá ver el detalle de la factura, es decir, los productos, dando cabida para la modificación (Punto 3.4.7.2.2) y la eliminación si lo desea (Punto 3.4.7.2.3).
- Código fuente: Anexo 7.2.2, página 133.
- Diseño de pantalla:

Maybet Productos ▾ Ajustes de existencias ▾ Categorías ▾ Subcategorías ▾ Proveedores ▾ Usuarios ▾ Ventas ▾ Devoluciones ▾ Caja ▾ Pedidos ▾ Facturas ▾ Salir

Administrador de facturas

Buscador:
 1) Por N° de Factura
 2) Por fecha de compra (yyyy-mm-dd)

Presione espacio para mostrar la lista completa

Detalle de facturas

Código factura	Código producto	Cantidad	Precio por unidad	Subtotal
12	1	13	1313	17069

[Modificar](#) [Eliminar](#)

Figura 3-21: Pantalla de lista y administración de detalle de factura

3.4.7.2.2 Modificar detalle de factura

- Nombre físico: fact_det_mod.php, fact_det_mod_guardar.php
- Objetivo: Permitir la modificación del detalle de la factura.

- Descripción: El usuario administrador logrará modificar algún producto del detalle de la factura existente en el sistema.
- Diagrama de bloque:

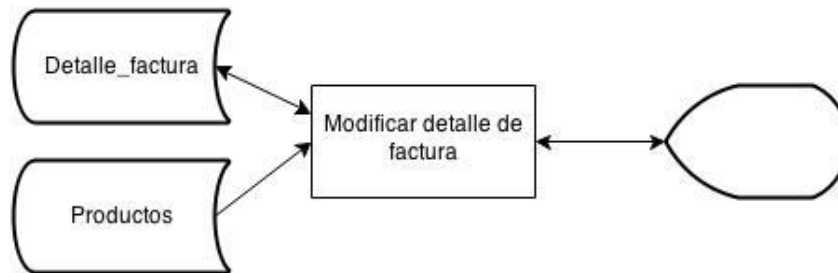


Diagrama 3-27: Diagrama de bloque modificar detalle de una factura

- Algoritmo: Listar los detalles de factura (Punto 3.4.7.2.1), y seleccionar el botón *Modificar* en de algún producto (Figura 3-21). Luego se cambian datos como Cantidad y Precio de producto en el formulario (Figura 3-22), se encuentran bloqueados de cambios los campos Código de factura, Código de proveedor, y el Subtotal de factura, el último calculado automáticamente. Al seleccionar el botón *Modificar* que se encuentra al final del formulario, se realizará exitosamente y se actualizará la base de datos, específicamente la tabla *Detalle_factura*.
- Código fuente: Anexo 7.2.3, página 135.
- Diseño de pantalla:

Maybet Productos - Ajustes de existencias - Categorías - Subcategorías - Proveedores - Usuarios - Ventas - Devoluciones - Caja - Pedidos - Facturas - Salir

Administrador de facturas

Buscador:
 1) Por N° de Factura
 2) Por fecha de compra (yyyy-mm-dd)

Presione espacio para mostrar la lista

Modifica un detalle de factura

Código de factura:

Código de producto:

(*)Cantidad:

(*)Precio producto (unidad):

Subtotal factura:

Figura 3-22: Pantalla de modificación de detalle de factura

3.4.7.2.3 Eliminar detalle de factura

- Nombre físico: fact_det_elim.php
- Objetivo: Permite la eliminación de un detalle de factura del sistema.
- Descripción: Un usuario administrador puede realizar exitosamente la acción de eliminar un producto en el detalle de factura.
- Diagrama de bloque:



Diagrama 3-28: Diagrama de bloque eliminar detalle de factura

- Algoritmo: Un usuario administrador debe apretar el botón *Eliminar* que se encuentra en la pantalla en que se listan los productos en el detalle de factura (Figura 3-21), luego de ésto, se visualiza un mensaje de confirmación de la operación. Si el usuario acepta la operación de eliminación, se elimina físicamente.
- Código fuente: Anexo 7.2.4, página 137.

3.4.7.3 Modificar facturas

- Nombre físico: fact_mod.php, fact_mod_guardar.php
- Objetivo: Permite la modificación de las facturas ya ingresadas en el sistema.
- Descripción: El usuario administrador será capaz de modificar alguna factura existente en el sistema.
- Diagrama de bloques:

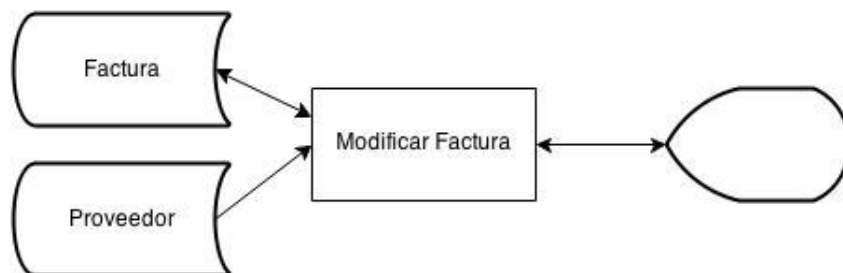


Diagrama 3-29: Diagrama de bloque modificar factura

- Algoritmo: Previo a la modificación, se deben listar las facturas (Punto 3.3.7.2) y elegir una factura en particular, y luego apretar su botón *Modificar*, a continuación, en la pantalla (Figura 3-23) se observa un formulario con los campos Código de factura, Proveedor ingresado, Proveedor nuevo, Fecha de compra, IVA, total factura, Estado de pago. Es posible cambiar solamente los campos seleccionando un Proveedor nuevo, ingresar el IVA, el total e la factura y el estado de pago, pagada o se debe. Al seleccionar el botón *Modificar* que se encuentra al final del formulario se permite la realización exitosa de la modificación, de la tabla *Factura*, en la base de datos.
- Código fuente: Anexo 7.3, página 138.
- Diseño de pantalla:

Maybet Productos - Ajustes de existencias - Categorías - Subcategorías - Proveedores - Usuarios - Ventas - Devoluciones - Caja - Pedidos - Facturas - Salir

Administrador de facturas

Buscador:
 1)Por N° de Factura
 2)Por fecha de compra (yyyy-mm-dd)

Presione espacio para mostrar la lista

Modifica factura

Código de factura: 12

Proveedor ingresado: 1234567

(*Proveedor nuevo: Empresa Libreria

Fecha de compra: 2013-12-15

IVA: 10

Total factura: 1313

Estado de pago: Pagada Se debe

Modificar

Figura 3-23: Pantalla de modificación de factura

3.4.7.4 Eliminar facturas

- Nombre físico: fact_elim.php
- Objetivo: Permite la eliminación de una factura del sistema.
- Descripción: Un usuario administrador puede realizar exitosamente la acción de eliminar una factura del sistema.
- Diagrama de bloque:

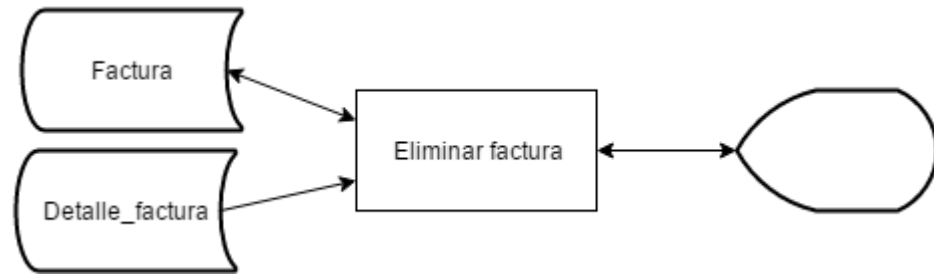


Diagrama 3-30: Diagrama de bloque eliminar factura

- Algoritmo: Un usuario administrador debe apretar el botón *Eliminar* que se encuentra en la pantalla en que se listan las facturas (Figura 3-20), paso siguiente, se observará un mensaje de confirmación de la operación. Si el usuario acepta la operación de eliminación, se elimina físicamente, siempre y cuando no existan detalles de facturas asociados a ésta.
- Código fuente: Anexo 7.4, página 140.

3.4.8 Ventas

Permite el control en las ventas y la consulta del historial de ventas.

3.4.8.1 Ingresar ventas

- Nombre físico: venta.php, ven_3.php, ven_3_agrega_guardar.php, ven_vender.php, venDetProd_agrega_guardar.php, ven_det_det_mod.php, ven_det_det_elim.php, ven_det_det_mod_guardar.php
- Objetivo: Agregar ventas al sistema.
- Descripción: Se podrán ingresar ventas al sistema, usuarios administradores y usuarios vendedores tendrán acceso a esta función.
- Diagrama de bloques:

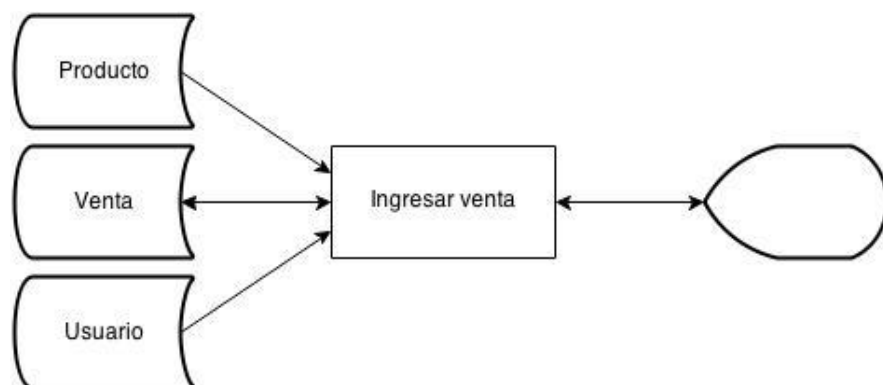


Diagrama 3-31: Diagrama de bloque ingresar venta

- Algoritmo: Se extraen datos de las tablas *Producto* y *Usuario* para obtener más detalles de qué se desea vender y quién está a cargo de la venta. El usuario administrador o vendedor visualizarán en su respectivo menú la opción de *Ventas*, y luego *Ingresar Venta*, se redireccionará a una pantalla en donde se permitirá efectuar la venta respectiva (Figura 3-24) en que busca el producto a vender, y se muestra una tabla que contiene información del producto buscado, si la cantidad en el *Stock* es mayor a la que se venderá, se aprieta el botón *Vender*. Paso siguiente se mostrará una pantalla (Figura 3-25) en que se registra la cantidad a vender del producto, luego de ésto se selecciona el botón *Agregar al detalle*, lo que activa un mensaje de información hacia el usuario de que el producto se agregó con éxito al detalle. Si son más productos para la venta, se repite el paso de búsqueda del producto. Si se desea revisar el detalle de la venta, se debe seleccionar el botón *Detalle de venta*. El Detalle de la venta (Figura 3-26) muestra los productos vendidos hasta el momento, la cantidad, el precio unitario y el subtotal de dinero, para volver a la venta, se selecciona el botón *Volver atrás*. Para seguir con la venta se pulsa el botón *Siguiente*, el cual conlleva a la visualización de la pantalla de medio de pago (Figura 3-27), y que contiene el total de dinero de la venta. Para realizar y registrar la venta, en la tabla *Venta*, se selecciona el botón *Registrar venta*.
- Código fuente: Anexo 8.1, página 141.
- Diseño de pantalla:

Código	Código barra	Categoría	Subcategoría	Nombre	Precio	Stock	Vender
17	97989786	PERFUMERIA	PERFUMES	CORAL	3000	8	Vender

Figura 3-24: Pantalla número uno de ingreso de una venta

Figura 3-25: Pantalla número dos de ingreso de una venta

Código venta	Código producto	Cantidad	Precio por unidad	Subtotal	
60	17	2	3000	6000	Modificar Eliminar
60	39	2	40000	80000	Modificar Eliminar

Figura 3-26: Pantalla número tres de ingreso de una venta

Figura 3-27: Pantalla número cuatro de ingreso de una venta

3.4.8.2 Listar ventas

- Nombre físico: vent_admin2.php
- Objetivo: Permite emitir la lista completa o filtrada de las ventas registradas en el sistema.
- Descripción: El usuario administrador puede buscar o filtrar por fecha las ventas registradas.
- Diagrama de bloque:

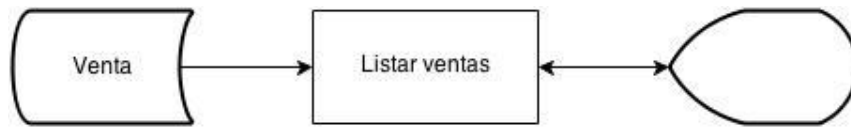


Diagrama 3-32: Diagrama de bloque listar venta

- Algoritmo: El usuario administrador debe acceder a *Ventas* y luego *Administrar ventas*. La siguiente pantalla (Figura 3-28) generará una lista de las ventas que se encuentran registradas en la base de datos, en la tabla *Venta*, dando opción para filtrar los resultados por fecha de venta. En cada venta se observa un botón *Detalle* (Punto 3.4.8.2.1)
- Código fuente: Anexo 8.2, página 150.
- Diseño de pantalla:

Captura de pantalla de la interfaz de usuario 'Consulta de ventas'. La barra de navegación superior incluye: Maybet, Productos, Ajustes de existencias, Categorías, Subcategorías, Proveedores, Usuarios, Ventas, Devoluciones, Caja, Pedidos, Facturas, Salir. El título de la sección es 'Consulta de ventas'. Hay un campo de búsqueda 'Buscador por fecha:' con un input 'Ingrese la fecha'. A continuación hay una tabla con las siguientes columnas: Código, Usuario, Boleta, Fecha, Hora, Tipo de pago, Total venta, Acción. Cada fila de datos tiene un botón 'Detalle'.

Código	Usuario	Boleta	Fecha	Hora	Tipo de pago	Total venta	Acción
52	123	20	2013-12-17	18:33:00	EF	43980	Detalle
53	123	21	2013-12-17	19:14:00	DB	5780	Detalle
54	123	22	2013-12-17	19:23:00	DB	9970	Detalle
55	123	23	2013-12-17	19:25:00	DB	3600	Detalle

Figura 3-28: Pantalla de lista de venta

3.4.8.2.1 Ver detalle de venta

- Nombre físico: `vent_det.php`
- Objetivo: Permite emitir una lista del detalle de una venta seleccionada por el usuario (Punto 3.4.8.2).
- Descripción: El usuario al seleccionar la opción *Detalle* (Figura 3-28) puede visualizar el detalle de cada venta, en la cual se listan los productos vendidos (Figura 3-29).
- Diagrama de bloque:

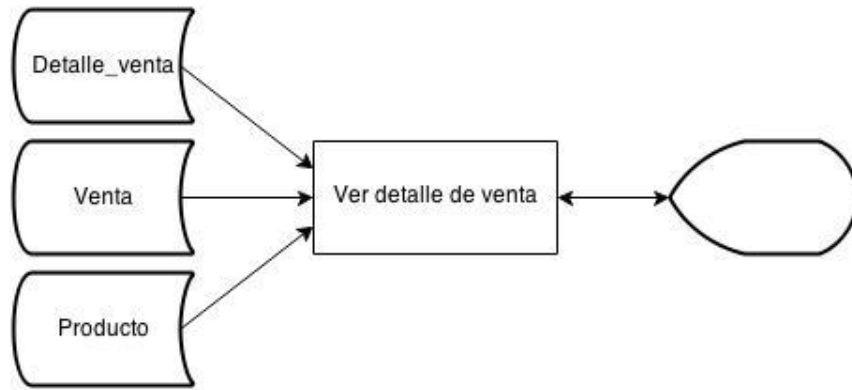


Diagrama 3-33: Diagrama de bloque ver detalle de venta

- Algoritmo: Si se accede a *Venta*, luego *Administrar ventas*, luego si, en la lista que se visualiza, se presiona el botón *Detalle*, en una venta en particular, se generará una lista (Figura 3-29) del detalle de la venta, es decir, los productos vendidos.
- Código fuente: Anexo 8.2.2, página 151.
- Diseño de pantalla:

Maybet Productos - Ajustes de existencias - Categorías - Subcategorías - Proveedores - Usuarios - Ventas - Devoluciones - Caja - Pedidos - Facturas - Salir				
Detalle de ventas				
Código venta	Nombre del producto	Cantidad	Precio por unidad	Subtotal
47	AGENDA 2014	2	5000	10000
47	CHANNEL NRO 5	1	40000	40000

Figura 3-29: Pantalla de lista de detalle de venta

3.4.9 Caja

Permite el control en el cierre de caja y la consulta del historial del movimiento de caja.

3.4.9.1 Cierre de caja

- Nombre físico: cierre.php
- Objetivo: Permite cerrar el día de caja.

- Descripción: Pantalla de cierre de caja en que se observa los totales del movimiento de caja diario.
- Diagrama de bloque:

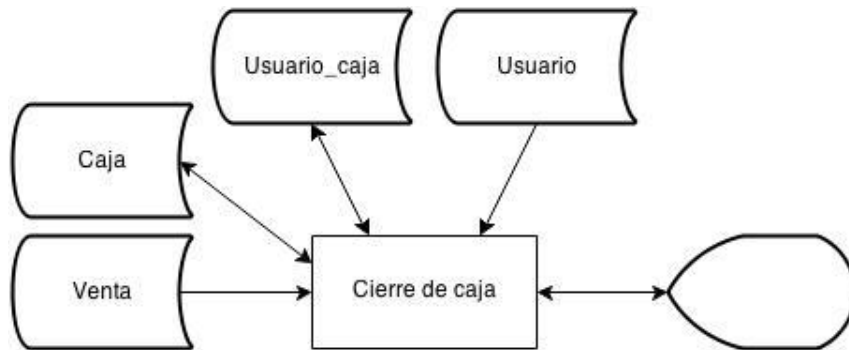


Diagrama 3-34: Diagrama cierre de caja

- Algoritmo: Si se ingresa a la opción *Caja* y luego *Cierre de caja*, se visualizará la pantalla con los totales del movimiento de caja diario. Además, el estado de la caja deja de ser activa, cerrando así el día de venta. Se observa un botón *Salir* para volver a la pantalla de inicio (Figura 3-30)
- Código fuente: Anexo 9.1, página 153.
- Diseño de pantalla:

Z							
Día de caja	Efectivo inicial	Ventas sin boleta	Ventas con boleta	Total giros	Ventas efectivos	Ventas débitos	Total efectivo
2013-12-17	10000	124	47692	4300	44316	3500	53516

Salir

Figura 3-30: Pantalla de cierre de caja

3.4.9.2 Historial de movimiento de caja

- Nombre físico: historial.php
- Objetivo: Permite la consulta del historial del movimiento de caja.
- Descripción: Pantalla en que se observan los totales del movimiento de caja, todos o filtrado por fecha.
- Diagrama de bloque:

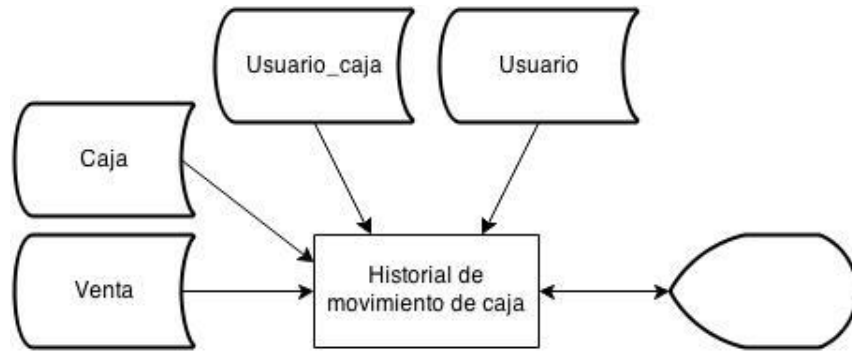


Diagrama 3-35: Diagrama historial de movimientos de caja

- Algoritmo: Ingresando a la opción *Caja* y luego *Historial*, se apreciará una pantalla con los totales del movimiento de caja, los que también pueden ser filtrados por fecha (Figura 3-31). Se observa un botón *Volver* para volver a la pantalla de inicio.
- Código fuente: Anexo 9.2, página 155.
- Diseño de pantalla:

Historial de movimiento de caja							
Buscador por fecha(dd-mm-aaaa): <input type="text" value="ingrese el dato"/>							
Día de caja	Efectivo inicial	Ventas sin boleta	Ventas con boleta	Total giros	Ventas efectivos	Ventas débitos	Total efectivo
2013-12-03	2000	1000	10000	1000	10000	100	1000000
2013-12-15	10000	1000	19000	1000	15000	5000	30000
2013-12-17	10000	124	47692	4300	44316	3500	53516

[Volver](#)

Figura 3-31: Pantalla de historial de movimiento de caja

CONCLUSIONES

Este trabajo de título surgió de la idea de implementar un sistema computarizado para un local comercial que, fundada hace más de dos décadas, tiene la necesidad de incorporar las nuevas tecnologías para permitir un control lo más exacto posible en la administración de la información manejada en el local.

Fueron aproximadamente 10 meses dedicados a la realización de este trabajo, tiempo que me sirvió para entender lo que hay detrás de la elaboración de un sistema informático completo, refiriéndome así al análisis y el desarrollo del software. Aprendí que si se efectúa un análisis meticuloso, la programación resulta más simple y rápida de elaborar, logrando así, una base sólida para que sean mínimos los inconvenientes que se encuentren en el curso del desarrollo.

La investigación personal y las personas que me apoyaron en el transcurso del desarrollo, contestándome dudas sobre las herramientas de software utilizadas, fueron una parte fundamental en el desarrollo del sistema. Además, al utilizar herramientas de software libre, es posible encontrar una extensa documentación disponible en línea, en donde usuarios alrededor del mundo comparten sus dudas y respuestas.

Aún la venta en el local comercial *Maybet* no abarca líneas de crédito y cheques, pero con miras hacia el futuro, este proyecto puede crecer hacia esos ámbitos, agregando estas funcionalidades, u otras que el usuario requiera, como por ejemplo, métodos que permitan generar reportes estadísticos con la información histórica sobre la tendencia de compra de los clientes, estadísticas de rentabilidad de los productos con los distintos proveedores, u otro método que permita ayudar a tomar decisiones estratégicas que mejoren el negocio.

También cabe destacar que, aplicando la misma metodología utilizada en este proyecto, se puede ajustar a cualquier local pequeño.

BIBLIOGRAFÍA

A continuación, se listan los sitios web que se visitaron para el desarrollo del proyecto:

- Manual de PHP: Sitio web creado con el objetivo del aprendizaje de PHP. Disponible en <http://www.phpya.com.ar/index.php>
- Tutorial de PHP y MySQL: Tutorial web creado por José Antonio Rodríguez. Disponible en <http://www.pablin.com.ar/computer/cursos/phpmysql/>
- Tutorial de programación: Sitio web, creado por Nelly Natalí Sánchez Rodríguez. Contiene documentación de CSS, Javascript, HTML, PHP, entre otros. Disponible en <http://blog.nnatali.com/category/php/>
- JQuery: Documentación en línea de JQuery, que contiene métodos de validación. Disponible en <http://jqueryvalidation.org/documentation/>

ANEXOS

ANEXOS

1. Identificación del usuario

1.1. index.php

```

<?php session_start(); ?>

<?php
    include("usu_admin1.php");
    $texto_tabla = new sistema;
    $texto_tabla->conectar();
?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>MAYBET</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
    <style type="text/css">
        body {
            background-image: url(img/fondo.png);
        }
    </style>
</head>

<body>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>
    <script type="text/javascript" src="js/jquery.validate.js"></script>
    <script type="text/javascript" src="js/jquery.numeric.js"></script>

<center>
    <div class="container">
        <div class="row-fluid">
            <div class="col-xs-12 col-sm-12 col-md-20 well well-sm">
                <h3>Ingreso al sistema</h3>
            </div>
        </div>
    </div>

</div>

<div class="container">
    <div class="row">
        <div class="span3">

```

```

<script type="text/javascript">
    $(function()
        {
            $('#formIndex').validate(
                {
                    rules :
                        {
                            txtIndexRut:
                                {
                                    required : true,
                                    maxlength : 9
                                },
                            txtIndexPass:
                                {
                                    required : true,
                                    maxlength : 10
                                }
                        },
                    messages :
                        {
                            txtIndexRut :
                                {
                                    required : "Debe ingresar un rut de usuario.",
                                    maxlength : "El largo del rut debe tener un máximo de 9 caracteres."
                                },
                            txtIndexPass:
                                {
                                    required : "Debe ingresar una contraseña.",
                                    maxlength : "El largo de la contraseña debe tener un máximo de 10 caracteres."
                                }
                        }
                });
    });
</script>
<form class="form-horizontal" role="form" name="formIndex" id="formIndex"
method="POST" action="index_proc.php">
    <div class="form-group"><label for="ing_usuario" class="col-lg-3
control-label">Usuario:</label>
        <div class="col-lg-3"><input type="text" class="form-control" name=
"txtIndexRut" placeholder="Ingrese su rut"></div>
    </div>
    <div class="form-group"><label for="ing_pass" class="col-lg-3 control-
label">Contraseña:</label>
        <div class="col-lg-3"><input type="password" class="form-control"
name="txtIndexPass" placeholder="Ingrese su contraseña"></div>
    </div>

    <div class="form-group">
        <div class="col-lg-offset-1 col-lg-5"><button type="submit" class="btn btn-
primary">Entrar</button></div>
    </div>

```

```

        </form>
    </div>
</div>
</center>

</body>
</html>

```

1.2. index_proc.php

```

<?php
include("usu_admin1.php");
$prim = new sistema;
$prim->conectar();

$ingRut = $_POST['txtIndexRut'];
$ingPass = $_POST['txtIndexPass'];

$consulta = mysql_query("SELECT * FROM usuario WHERE cod_usuario = '$ingRut' AND
password = '$ingPass'");
// $filas = mysql_fetch_assoc($consulta);
$resultado = mysql_fetch_array($consulta);
$sw = 0; //ambito, todo lo que tenga llave
if ($resultado['cod_usuario'] == $ingRut)
    {
        session_start(); // luego le asigno los $_SESSION para que queden almacenados
en las cookies

        $_SESSION['ingRut'] = $ingRut; //variable de entorno
        $_SESSION['ingPass'] = $ingPass;
        $_SESSION['nombre'] = $resultado['nombre'];
        $_SESSION['tipo_usuario'] = $resultado['tipo_usuario'];

        //now()
        $sw=1;
        $usuario = $resultado;
    }

if($sw == 0)
    {
        print "Usuario inválido";

        // echo $resultado['nombre']; Aquí dejo guardado el nombre de la persona que
inició sesión

    } else
    {
        $ingRut = $_POST['txtIndexRut'];
        date_default_timezone_set('UTC');
        $hoy = date("Y-m-d");
    }

```

```

VALUES ('0', '12')";

$query = "INSERT INTO usuario_caja (cod_dia_caja, cod_usuario)

$insertar = mysql_query($query);

if($insertar)
{ print "1";
  print $insertar;

  header('Location: http://localhost/TT/p_abrir_caja.php');
} else{
  print "2";
  print $query;
  print "--";
  print $hoy;
  print "--";
  print $insertar;
  header('Location: http://localhost/TT/inicio.php');
}
}
?>

```

2. Inicio

2.1. inicio.php

```

<?php
  session_start();
  if (isset($_SESSION['ingRut']))
  {
    ?>
    <label>Bienvenida <?php print $_SESSION['nombre'];?></label>
    <?php
  }
  if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']="ADM"))
  {
    include("b_menu_admin.php");
  }else
  {
    include("b_menu_vend.php");
  }
?>

```

```

<!DOCTYPE html>
<html>
  <head>
    <title>Inicio</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">
  </head>
  <body>

```

```

<script src="http://code.jquery.com/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<center>
  <div id="contenedor">
    <div id="myCarousel" class="carousel slide">
      <ol class="carousel-indicators">
        <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
        <li data-target="#myCarousel" data-slide-to="1"></li>
        <li data-target="#myCarousel" data-slide-to="2"></li>
        <li data-target="#myCarousel" data-slide-to="3"></li>
        <li data-target="#myCarousel" data-slide-to="4"></li>
      </ol>

      <div class="carousel-inner">
        <div class="active item"></div>
        <div class="item"></div>
        <div class="item"></div>
        <div class="item"></div>
        <div class="item"></div>
        <div class="item"></div>
        <div class="item"></div>

      </div>
      <!-- Carousel nav -->
      <a class="carousel-control left" href="#myCarousel" data-slide="prev">&lsaquo;</a>
      <a class="carousel-control right" href="#myCarousel" data-slide="next">&rsaquo;</a>
    </div>
  </div>

</center>
<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<script>
  $(document).ready(function(){
    $('.carousel').carousel()
  });
</script>
<script>
  $(document).ready(function(){
    $('.carousel').carousel({
      interval: 10000
    });
  });
</script>
</body>
</html>

```

3. Productos

3.1. Ingresar productos

3.1.1. p_prod_agrega.php

```

<?php
    session_start();
    include("prod_admin1.php");
    $texto_tabla = new sistema;
    $texto_tabla->conectar();

    if (isset($_SESSION['ingRut']))
    {
        ?>
        <label>Bienvenida <?php print $_SESSION['nombre'];?></label>
        <?php
    }
    if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
    {
        include("b_menu_admin.php");
    }else
    {
        header('Location: http://localhost/TT/inicio.php');
    }

?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Agrega productos</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
    <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>
    <script type="text/javascript" src="js/jquery.validate.js"></script>
    <script type="text/javascript" src="js/jquery.numeric.js"></script>

    <center>
    <div class="container">
        <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h3>Agrega
un producto</h3></div></div>
    </div>
</center>

```

```

<div class="container">
<div class="row">
<div class="span3">

<script type="text/javascript">
    $(function()
        {
        $('#form_agrega').validate(
            {
            rules :
                {
            txt_barra:
                {
                maxlength : 13,
                digits: true
            },
            txt_nom:
                {
            required : true,
            maxlength : 40
            },
            txt_cant:
                {
            required : true,
            maxlength : 5,
                digits: true,
                min: 1
            },
            txt_cant_cri:
                {
            required : true,
            maxlength : 2,
                digits: true,
                min: 1
            },
            txt_precio:
                {
            required : true,
            maxlength : 6,
                digits: true,
                min: 1
            }
            },
            messages :
                {
            txt_barra :
                {
                digits: "Debe ingresar sólo números.",
            maxlength : "El largo del código debe tener un máximo de 13 caracteres."
            },
            txt_nom:
                {

```

```

required : "Debe ingresar un nombre.",
maxlength : "El largo del nombre debe tener un máximo de 40 caracteres."
},
    txt_cant:
    {
required : "Debe ingresar una cantidad.",
maxlength : "La cantidad debe tener un máximo de 5 caracteres.",
        digits: "Debe ingresar sólo números.",
        min : "La cantidad no debe ser igual o menor
a 0."
    },
    txt_cant_cri:
    {
required : "Debe ingresar una cantidad",
maxlength : "La cantidad debe tener un maximo de 2 caracteres",
        digits: "Debe ingresar sólo números",
        min : "La cantidad no debe ser igual o menor
a 0"
    },
    txt_precio:
    {
required : "Debe ingresar una cantidad",
maxlength : "La cantidad debe tener un maximo de 6 caracteres",
        digits: "Debe ingresar solo números",
        min : "La cantidad no debe ser menor a 0"
    }
}
});
});
</script>

<form class="form-horizontal" id= "form_agrega" name= "form_agrega" role="form" method
= "GET">
    <div class="form-group">
        <label for="ing_barra" class="col-lg-3 control-label">Código
de barra:</label>
        <div class="col-lg-3"><input type="text" class="form-control"
name="txt_barra" placeholder="Ingrese el código de barra"></div>
        </div>
        <div class="form-group">
            <label for="ing_subcat" class="col-lg-3 control-label">Subcategoría:</label>
            <div class="col-lg-3">
                <select class="form-control" id="txt_sub" name="txt_sub">
                    <?php
                                $resultado = "SELECT * FROM
subcategoria ORDER BY nombre asc";
                                $query = mysql_query
($resultado);
                                while ($fila = mysql_fetch_array($query))
                                {
                                    echo '<option value=
'.".$fila["cod_subcategoria"].">'. $fila["nombre"].'</option>';

```

```

        }
        while ($fila = mysql_fetch_array($resultado));
        echo '</select>';
    ?>

    </select>
</div>
</div>
<div class="form-group">
    <label for="ing_nom" class="col-lg-3 control-label">Nombre:</label>
    <div class="col-lg-4"><input type="text" class="form-control" name="txt_nom"
placeholder="Ingrese el nombre del producto"></div>
</div>

<div class="form-group">
    <label for="ing_cant" class="col-lg-3 control-label">Cantidad:</label>
    <div class="col-lg-2"><input type="text" class="form-control" name="txt_cant"
placeholder="xxxx"></div>
</div>

<div class="form-group">
    <label for="ing_cant_cri" class="col-lg-3 control-label">Cantidad crítica:</label>
    <div class="col-lg-2"><input type="text" class="form-control" name="txt_cant_cri"
placeholder="xx"></div>
</div>

<div class="form-group">
    <label for="ing_precio" class="col-lg-3 control-label">Precio de venta:</label>
    <div class="col-lg-3"><input type="text" class="form-control" name="txt_precio"
placeholder="$"></div>
</div>

<div class="form-group">
    <div class="col-lg-offset-1 col-lg-8"><button type="submit" class="btn btn-success"
onClick="prod_agrega_guardar()">Siguiente</button></div>
</div>
</form>
</div>
</div>
</div>
</body>

```

3.1.2. prod_agrega_guardar.php

```

<?php
include("prod_admin1.php");
$prim = new sistema;
$prim->conectar();

$cb = $_GET['txt_barra'];
$subc = $_GET['txt_sub'];
$nom = $_GET['txt_nom'];
$cant = $_GET['txt_cant'];

```

```
$cri = $_GET['txt_cant_cri'];
$prec = $_GET['txt_precio'];
```

```
$insertar= mysql_query("INSERT INTO producto (cod_barra, cod_subcategoria, nom_producto,
cantidad_producto, cantidad_critica_prod, precio_unitario, articulo_eliminado) VALUES ('$cb', '$subc',
'$nom', '$cant', '$cri', '$prec', '0')");
```

```
if($insertar)
{
    print "Producto agregado correctamente";
} else
{
    print "Error al agregar producto";
}
?>
```

3.2. Listar productos

3.2.1. prod_admin2.php

```
<?php
    session_start();
    include("prod_admin1.php");
    $texto_tabla = new sistema;
    $texto_tabla->conectar();

    if (isset($_SESSION['ingRut']))
    {
        ?>
        <label>Bienvenida <?php print $_SESSION['nombre'];?></label>
        <?php
        }
        if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']="ADM"))
        {
            include("b_menu_admin.php");
        }else
        {
            header('Location: http://localhost/TT/inicio.php');
        }

    ?>
</html>
<!doctype html>
<html lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Administrador de productos</title>
```

```

<link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>
</html>

<body>
  <center>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <div class="container">
      <div class="row-fluid">
        <div class="col-xs-12 col-sm-12 col-md-12 well well-sm">
          <h4>Administrador de productos</h4>
        </div>
      </div>
    </div>
  </center>

  <script src="js/ajax.js" language="javascript"></script>

  <div class="container">
    <div class="col-lg-offset-1 col-lg-5">
      <button type="submit" class="btn btn-danger">Productos con bajo stock</button>
    </div>
  </div>

  <div class = "container">
    <label style="color: #009" style="font:bold" class="col-lg-4 control-label">Buscador:</label>
  </div>
  <div class = "container">
    <label style="color: #936" for="lbl_busca_ped" class="col-lg-4 control-label">1)Por Código de
    barra</label>
  </div>
  <div class = "container">
    <label style="color: #936" for="lbl_busca_ped" class="col-lg-4 control-label">2)Por nombre del
    producto</label>
  </div>
  <div class = "container">
    <label style="color: #936" for="lbl_busca_ped" class="col-lg-4 control-label">3)Por
    categoría</label>
  </div>
  <div class = "container">
    <label style="color: #936" for="lbl_busca_ped" class="col-lg-4 control-label">4)Por
    subcategoría</label>
  </div>
  <div class = "container">
    <label style="color: #009" for="lbl_busca_ped" class="col-lg-4 control-label">Presione espacio para
    mostrar la lista completa</label>

  <div class = "container">
    <div class="input"><input type="text" class="col-lg-4" class="text_buscador" name="text_buscador"
    id="valor" onKeyUp="busqueda()" placeholder="Ingrese el Nombre"></div>
  </div>
  <div class = "container">
    <div class="tabla_result" id= "tabla_result"></div>
  </div>

```

```
</div>
</body>
```

3.2.2. Ver proveedor

3.2.2.1. prod_det.php

```
<?php
    session_start();
    include("prod_admin1.php");
    $texto_tabla = new sistema;
    $texto_tabla->conectar();

    $texto_buscadore = $_GET['texto_buscadore'];

    print $texto_buscadore;

?>
<label>Bienvenida <?php print $_SESSION['nombre'];?></label>
<?php

if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
{
}else
{
    header('Location: http://localhost/TT/inicio.php');
}

?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Detalle productos</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
    <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>
    <script type="text/javascript" src="js/jquery.validate.js"></script>
    <script type="text/javascript" src="js/additional-methods.js"></script>

    <script type="text/javascript" src="js/jquery.numeric.js"></script>

    <center>
    <div class="container">
        <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h3>Detalle
del producto</h3></div></div>
```

```

        </div>
    </center> .

<?php
    $texto_buscador = trim($texto_buscador);
    $query = (mysql_query ("SELECT * FROM prod_prov WHERE cod_producto =
'$texto_buscador'"));

    if (mysql_num_rows($query) <=0)
    {
        print 'Sin resultados';
    } else
        {
            print '<table class="table table-hover">
            <td>Código producto</td>
                <td>Código proveedor</td>
                <td>Administración</td>
            <tr>';

            while ($fila = mysql_fetch_assoc($query))
            {

                print '<td>'.$fila['cod_producto'].'</td>
                    <td>'.$fila['cod_proveedor'].'</td>
                    <td>
                        <button type="button" class="btn btn-
danger btn-xs" onclick="ConfirmarDetProd('.$fila['cod_producto'].');" >Eliminar</button></td>
                    </tr>';

            }
            print '</table>';
        }

?>

```

3.2.3. Eliminar proveedores de producto

3.2.3.1. prod_det_elim.php

```

<?php
    include("prod_admin1.php");
    $texto_tabla = new sistema;
    $texto_tabla->conectar();
    $cod_proveedor = $_GET['texto_buscador'];
    $texto_tabla->EliminarDetProd($cod_proveedor);

?>

```

3.2.4. Ver informe de productos con bajo Stock

3.2.4.1. bajoStock.php

```

<?php

```

```

        session_start();
        include("prod_admin1.php");
        $tx_tbl_prod = new sistema;
        $tx_tbl_prod->conectar();

?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>lista detalle de venta</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
    <style type="text/css">
    body {
        background-image: url(img/fondo.png);
    }
    </style>
</head>

<body>
    <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>
    <script type="text/javascript" src="js/jquery.validate.js"></script>
    <script type="text/javascript" src="js/additional-methods.js"></script>

    <script type="text/javascript" src="js/jquery.numeric.js"></script>

    <center>
    <div class="container">
        <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-
sm"><h3>Productos con bajo Stock</h3></div></div>
        </div>
    </center>

    <button type="submit" class="btn btn-warning"><a href="javascript:history.back(1)">Volver
Atrás</a></button>
    <div class="tbl_res_det_ven" id= "tbl_res_ven_det"></div>

<?php
    $query = (mysql_query ("SELECT * FROM producto WHERE cantidad_producto <
cantidad_critica_prod"));

    if (mysql_num_rows($query) <=0)
    {
        print 'Sin resultados';
    } else
    {

        print '<table id="tbl_res_ven_det" class="table table-hover">
        <td>Código</td>
        <td>Nombre</td>

```

```

        <td>Cantidad disponible</td>
    <tr>';

    while ($fila = mysql_fetch_assoc($query))
    {

        print '<td>'.$fila['cod_producto'].'</td>
            <td>'.$fila['nom_producto'].'</td>

        <td>'.$fila['cantidad_producto'].'</td>

            <td>

        </tr>';
    }
    print '</table>';
}

```

3.3. Modificar productos

3.3.1. prod_mod.php

```

<?php
include("prod_admin1.php");
$texto_tabla = new sistema;
$texto_tabla->conectar();
$texto_buscador = $_GET['texto_buscador'];
$texto_tabla = mysql_fetch_assoc(mysql_query ("SELECT prod.cod_producto, prod.cod_barra,
cat.nombre as nombrecat, sub.nombre as nombresub,prod.nom_producto ,prod.cantidad_producto,
prod.cantidad_critica_prod,
                prod.precio_unitario
                FROM producto prod, subcategoria sub,
categoria cat
                WHERE sub.cod_subcategoria =
                AND sub.cod_categoria = cat.cod_categoria
                AND cod_producto = '$texto_buscador'"));
?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Modifica productos</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
    <center>
        <script type="text/javascript" src="js/bootstrap.js"></script>
        <script type="text/javascript" src="js/ajax.js"></script>

```

```

<div class="container">
  <div class="row-fluid">
    <div class="col-xs-12 col-sm-12 col-md-20 well well-sm">
      <h3>Modifica producto</h3>
    </div>
  </div>
</div>
</center>

<div class="container">
  <div class="row">
    <div class="span3">

      <form class="form-horizontal" role="form" method = "get" action = "prod_mod_guardar.php">
        <div class="form-group">
          <label for="lbl_cb" class="col-lg-3 control-label">Código del producto:</label>
          <div class="col-lg-2"><input type="text" class="form-control" id="cod_producto"
value="<?php print $texto_tabla['cod_producto'];?>" disabled></div>
        </div>

        <div class="form-group">
          <label for="lbl_cb" class="col-lg-3 control-label">Código de barra:</label>
          <div class="col-lg-2"><input type="text" class="form-control" id="cod_barra" value="<?php
print $texto_tabla['cod_barra'];?>" disabled></div>
        </div>

        <div class="form-group">
          <label for="ing_cat" class="col-lg-3 control-label">Categoría ingresada:</label>
          <div class="col-lg-3"><input type="text" class="form-control" id="nombrecat" disabled
value="<?php print $texto_tabla['nombrecat'];?>"></div>
        </div>

          <div class="form-group">
            <label for="ing_cat" class="col-lg-3 control-label">Subcategoría ingresada:</label>
            <div class="col-lg-3"><input type="text" id="nombresub" class="form-control"
name="txt_old_cat" value="<?php print $texto_tabla['nombresub'];?>" disabled></div>
          </div>

          <div class="form-group">
            <label for="ing_subcat" class="col-lg-3 control-label">Subcategoría nueva:</label>
            <div class="col-lg-3">
              <select class="form-control" id="nombre">
                <?php
ORDER BY nombre asc";

                $resultado = "SELECT * FROM subcategoria

                $query = mysql_query ($resultado);
                while ($fila = mysql_fetch_array($query))
                {
                  echo '<option value= "'. $fila["cod_subcategoria"]."'>'. $fila["nombre"].'</option>';
                }
                while ($fila = mysql_fetch_array($resultado));

                echo '</select>';
              ?>
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>

```

```

        </select>
    </div>
</div>

<div class="form-group">
    <label for="ing_nom" class="col-lg-3 control-label">Nombre:</label>
    <div class="col-lg-3"><input type="text" class="form-control" id="nom_producto"
value="<?php print $texto_tabla['nom_producto'];?>" ></div>
    </div>

<div class="form-group">
    <label for="ing_cant" class="col-lg-3 control-label">Cantidad:</label>
    <div class="col-lg-1"><input type="text" class="form-control" id="cantidad_producto"
value="<?php print $texto_tabla['cantidad_producto'];?>" disabled></div>
    </div>

<div class="form-group">
    <label for="ing_nom" class="col-lg-3 control-label">Cantidad crítica:</label>
    <div class="col-lg-1"><input type="text" class="form-control" id="cantidad_critica_prod"
value="<?php print $texto_tabla['cantidad_critica_prod'];?>"></div>
    </div>

<div class="form-group">
    <label for="ing_nom" class="col-lg-3 control-label">Precio de venta:</label>
    <div class="col-lg-3"><input type="text" class="form-control" id="precio_unitario" value=
"<?php print $texto_tabla['precio_unitario'];?>"></div>
    </div>
</div>

<div class="form-group">
    <div class="col-lg-offset-1 col-lg-5"><button type="submit" class="btn btn-success"
onClick="prod_mod_guardar(<?php print $texto_tabla['cod_producto'];?>)">Modificar</button></div>
    </div>
</form>
</div>
</div>
</div>
</body>

```

3.3.2. prod_mod_guardar.php

```

<?php
include("prod_admin1.php");
$texto_tabla = new sistema;
$texto_tabla->conectar();

$cod_producto = $_GET['cod_producto'];
$cod_barra = $_GET['barra'];
$cod_categoria = $_GET['cat'];
$cod_subcategoria = $_GET['subc'];
$nom_producto = $_GET['nom'];
$cantidad_producto = $_GET['cant'];

```

```

$cantidad_critica_prod = $_GET['cri'];
$precio_unitario = $_GET['prec'];

//$cod_categoria = 2;
//$actualizar = mysql_query ("UPDATE subcategoria SET cod_categoria = '$cod_categoria' WHERE
cod_subcategoria = //$cod_subcategoria");

$varCat = mysql_query("SELECT cod_categoria FROM sub_categoria where cod_subcategoria =
$cod_subcategoria");

                                if($varCat){
                                    $consulta =
mysql_fetch_assoc($varCat);
                                    $cod_categoria =
$consulta['cod_categoria'];
                                }

$actualizar = mysql_query ("UPDATE producto SET cod_producto = '$cod_producto', cod_barra =
'$cod_barra', cod_subcategoria = '$cod_subcategoria', nom_producto = '$nom_producto',
cantidad_producto = '$cantidad_producto', cantidad_critica_prod = '$cantidad_critica_prod',
precio_unitario = '$precio_unitario' WHERE cod_producto = '$cod_producto'");

if($actualizar)
    {
        print "Producto modificado correctamente";
    } else
    {
        print "Error al modificar producto";
    }
?>

```

3.4. Eliminar productos

```

prod_elim.php
<?php
include("prod_admin1.php");
$texto_tabla = new sistema;
$texto_tabla->conectar();
$cod_producto = $_GET['texto_buscadore'];
$texto_tabla->Eliminar($cod_producto);
?>

```

4. Subcategorías

4.1. Ingresar subcategorías

4.1.1. p_subcat_agrega.php

```

<?php
    session_start();

```

```

if (isset($_SESSION['ingRut']))
{
?>
<label>Bienvenida <?php print $_SESSION['nombre'];?></label>
<?php
}
if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
{
    include("b_menu_admin.php");
}else
{
    include("b_menu_vend.php");
}
include("subcat_admin1.php");
$texto_tabla = new sistema;
$texto_tabla->conectar();
?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Agrega Subcategoría</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
    <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>
    <script type="text/javascript" src="js/jquery.validate.js"></script>
    <script type="text/javascript" src="js/jquery.numeric.js"></script>

    <center>
    <div class="container">
        <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h3>Agrega
una subcategoría</h3></div></div>
    </div>
    </center>

    <div class="container">
    <div class="row">
        <div class="span3">

            <script type="text/javascript">
            $(function()
            {
                $('#form_agrega_subcat').validate(
                {
                    rules :
                    {
                        txtNomSubcat:
                        {

```

```

        required : true,
        maxlength : 30
    },

    txtCatSubcat:
    {
        required : true
    }

    },
    messages :
    {
        txtNomSubcat :
        {
            required : "Debe ingresar un nombre de subcategoría.",
            maxlength : "El largo del nombre debe tener un máximo de 30 caracteres."
        },

        txtCatSubcat:
        {
            required : "Debe ingresar
una de categoría.",
        }

    }

});
});
</script>

<form class="form-horizontal" id= "form_agrega_subcat" name= "form_agrega_subcat"
role="form" method = "GET">
    <div class="form-group">
        <label for="ingSubcNom" class="col-lg-3 control-label">Nombre:</label>
        <div class="col-lg-4"><input type="text" class="form-control" name="txtNomSubcat"
placeholder="Ingrese el nombre de la categoría"></div>
    </div>

    <div class="form-group">
        <label for="ingSubcCat" class="col-lg-3 control-label">Categoría a la que
pertenece:</label>
        <div class="col-lg-3">
            <select class="form-control" id="txtCatSubcat" name="txtCatSubcat">
                <?php
                    $resultado = "SELECT * FROM
categoria ORDER BY nombre asc";
                    $query = mysql_query
($resultado);
                    while ($fila = mysql_fetch_array($query))
                    {
                        echo '<option value=
"'. $fila["cod_categoria"]. "'>'. $fila["nombre"]. '</option>';
                    }
            </select>
        </div>
    </div>
</form>

```

```

        while ($fila = mysql_fetch_array($resultado));
                                echo '</select>';
                                ?>

        </select>
    </div>
</div>

    <div class="form-group">
        <div class="col-lg-offset-1 col-lg-5">
            <button                type="submit"                class="btn                btn-primary"
onClick="subcat_agrega_guardar()">Ingresar</button>
        </div>
    </div>
</form>
</div>
</div>
</body>

```

4.1.2. subcat_agrega_guardar.php

```

<?php
include("subcat_admin1.php");
$prim = new sistema;
$prim->conectar();

$sbnom = $_GET['txtNomSubcat'];
$sbc = $_GET['txtCatSubcat'];

$insertar= mysql_query("INSERT INTO subcategoria (cod_categoria, nombre) VALUES ('$sbc',
'$sbnom')");

if($insertar)
    {
        print "Subcategoría agregada correctamente";
    } else
    {
        print "Error al agregar categoría";
    }
?>

```

4.2. Listar subcategorías

4.2.1. subcat_admin2.php

```

<?php
    session_start();
    include("usu_admin1.php");
    $tx_tbl_usu = new sistema;
    $tx_tbl_usu->conectar();

```

```

if (isset($_SESSION['ingRut']))
{
?>
<label>Bienvenida <?php print $_SESSION['nombre'];?></label>
<?php
}
if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
{
    include("b_menu_admin.php");
}else
{
    header('Location: http://localhost/TT/inicio.php');
}

?>

<html>
<!doctype html>
<html lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Administrador de proveedores</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>
</html>

<body>
<center>
<script type="text/javascript" src="js/bootstrap.js"></script>
<div class="container">
<div class="row-fluid">
<div class="col-xs-12 col-sm-12 col-md-20 well well-sm">
<h4>Administrador de subcategorías</h4>
</div>
</div>
</div>
</center>

<script src="js/ajax.js" language="javascript"></script>

<div class = "container">
<label for="lbl_busca_subcat" class="col-lg-4 control-label">Buscador por nombre de
subcategoría:</label>
<div class="input"><input type="col-lg-4" class="text_buscador_subcat"
name="text_buscador_subcat" id="valorSubcat" onKeyUp="busca_subcat()"
placeholder="Ingrese la Subcategoría"></div>
<div class="tbl_res_subcat" id= "tbl_res_subcat"></div>

</div>
</body>

```

4.3. Modificar subcategorías

4.3.1. subcat_mod.php

```

<?php
    session_start();
    include("subcat_admin1.php");
    $tx_tbl_subcat = new sistema;
    $tx_tbl_subcat->conectar();
    $tx_busca_subcat = $_GET['tx_busca_subcat'];
    $tx_tbl_subcat = mysql_fetch_assoc(mysql_query ("SELECT cat.cod_categoria, cat.nombre as
nombrecat, sub.cod_subcategoria, sub.cod_categoria, sub.nombre as nombresubcat
FROM subcategoria sub, categoria
cat
WHERE cat.cod_categoria =
sub.cod_categoria
AND cod_subcategoria =
'$tx_busca_subcat'"));

    if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']="ADM"))
    {

    }else
    {
        header('Location: http://localhost/TT/inicio.php');
    }
?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Modifica subcategorías</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
    <center>
        <script type="text/javascript" src="js/bootstrap.js"></script>
        <script type="text/javascript" src="js/ajax.js"></script>

        <div class="container"><div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-12 well well-
sm">
            <h3>Modifica subcategorías</h3>
        </div></div></div>
    </center>

    <div class="container"><div class="row"><div class="span3">

        <form class="form-horizontal" role="form" method = "GET">

```

```

    <div class="form-group">
        <label for="ing_rut_usu" class="col-lg-3 control-label">Código de la
subcategoria:</label>
        <div class="col-lg-2"><input type="text" class="form-control" id="cod_subcategoria"
value="<?php print $tx_tbl_subcat['cod_subcategoria'];?>" disabled></div>
    </div>

    <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">Categoría a la que pertenece:</label>
        <div class="col-lg-3"><input type="text" class="form-control" id="nombrecat" disabled
value="<?php print $tx_tbl_subcat['nombrecat'];?>"></div>
    </div>

    <div class="form-group">
        <label for="ing_subcat" class="col-lg-3 control-label">Categoría nueva:</label>
        <div class="col-lg-3">
            <select class="form-control" id="nombrecatnueva">
                <?php
                    $resultado = "SELECT * FROM categoria
ORDER BY nombre asc";
                    $query = mysql_query ($resultado);
                    while ($fila = mysql_fetch_array($query))
                    {
                        echo '<option value= "'. $fila["cod_categoria"]."'>'. $fila["nombre"].'</option>';
                    }
                    while ($fila = mysql_fetch_array($resultado));
                </select>';
            </div>
        </div>

        <div class="form-group">
            <label for="ing_cat" class="col-lg-3 control-label">Nombre:</label>
            <div class="col-lg-3"><input type="text" id="nombresub" class="form-control" value="<?php
print $tx_tbl_subcat['nombresubcat'];?>"></div>
        </div>

        <div class="form-group">
            <div class="col-lg-offset-1 col-lg-5"><button type="submit" class="btn btn-success"
onClick="subcat_mod_guardar(<?php
print $tx_tbl_subcat['cod_subcategoria'];?>)">Modificar</button></div>
        </div>
    </form>
</div>
</div>
</div>
</body>

```

4.3.2. subcat_mod_guardar.php

```
<?php
```

```

include("subcat_admin1.php");
$tx_tbl_subcat = new sistema;
$tx_tbl_subcat->conectar();

$cod_subcategoria = $_GET['cod_subcategoria'];
$cod_categoria = $_GET['nombrecatnueva'];
$nombre = $_GET['nombresub'];

$actualizar = mysql_query ("UPDATE subcategoria SET cod_subcategoria = '$cod_subcategoria',
cod_categoria = '$cod_categoria', nombre = '$nombre'");

if($actualizar)
{
    print "Subcategoria modificada correctamente";
} else
{
    print "Error al modificar subcategoria";
}
?>

```

4.4. Eliminar subcategorías

4.4.1. subcat_elim.php

```

<?php
include("subcat_admin1.php");
$tx_tbl_subcat = new sistema;
$tx_tbl_subcat->conectar();
$cod_subcategoria = $_GET['tx_busca_subcat'];
$tx_tbl_subcat->EliminarSubcat($cod_subcategoria);
?>

```

5. Usuarios

5.1. Ingresar usuarios

5.1.1. p_usu_agrega.php

```

<?php
    session_start();
    include("usu_admin1.php");
    $tx_tbl_usu = new sistema;
    $tx_tbl_usu->conectar();

    if (isset($_SESSION['ingRut']))
    {
        ?>
        <label>Bienvenida <?php print $_SESSION['nombre'];?></label>
        <?php

```



```

    {
      required : true,
      maxlength : 20
    },
    txtPassUsu:
    {
      required : true,
      maxlength : 10,
    },
    txtDirUsu:
    {
      required : true,
      maxlength : 40,
    },
    txtFonoUsu:
    {
      required : true,
      maxlength : 12,
      digits: true,
    },
  },
  messages :
  {
    txtRutUsu :
    {
      required : "Debe ingresar un rut sin '-'.",
      maxlength : "El largo del rut debe tener un
máximo de 9 caracteres."
    },
    txtNomUsu:
    {
      required : "Debe ingresar un nombre.",
      maxlength : "El largo del nombre debe tener
un máximo de 20 caracteres."
    },
    txtPassUsu:
    {
      required : "Debe ingresar una contraseña.",
      maxlength : "La cantidad debe tener un
máximo de 10 caracteres.",
    },
    txtDirUsu:
    {
      required : "Debe ingresar una dirección",
      maxlength : "La cantidad debe tener un
maximo de 40 caracteres",
      digits: "Debe ingresar sólo números",
    },
    txtFonoUsu:

```

```

        {
            required : "Debe ingresar un teléfono",
            maxlength : "La cantidad debe tener un
maximo de 12 caracteres",
            digits: "Debe ingresar sólo números",
        },
    },
});
});
</script>
<script>
function validaRut(Objeto)
{
    var tmpstr = "";
    var intlargo = Objeto.value
    if (intlargo.length > 0)
    {
        crut = Objeto.value
        largo = crut.length;
        if ( largo < 2 )
        {
            alert('Debe ingresar un rut válido.')
            Objeto.focus()
            return false;
        }
        for ( i=0; i < crut.length ; i++ )
        if ( crut.charAt(i) != '.' && crut.charAt(i) != '-' && crut.charAt(i) != '-' )
        {
            tmpstr = tmpstr + crut.charAt(i);
        }
        rut = tmpstr;
        crut=tmpstr;
        largo = crut.length;
        if ( largo > 2 )
            rut = crut.substring(0, largo - 1);
        else
            rut = crut.charAt(0);
        dv = crut.charAt(largo-1);
        if ( rut == null || dv == null )
            return 0;
        var dvr = '0';
        suma = 0;
        mul = 2;
        for (i= rut.length-1 ; i >= 0; i--)
        {
            suma = suma + rut.charAt(i) * mul;
            if (mul == 7)
                mul = 2;
            else
                mul++;
        }
        res = suma % 11;
        if (res==1)

```

```

        dvr = 'k';
    else if (res==0)
        dvr = '0';
    else
    {
        dvi = 11-res;
        dvr = dvi + "";
    }
    if ( dvr != dv.toLowerCase() )
    {
        alert('Error, rut inválido')
        Objeto.focus()
        return false;
    }
    return true;
}
}
</script>

```

```

<form class="form-horizontal" id= "formUsuAgrega" name= "formUsuAgrega" role="form" method
= "GET">

```

```

<div class="form-group">

```

```

<label for="ing_rut_usu" class="col-lg-3 control-label">Rut del usuario:</label>

```

```

<div class="col-lg-2"><input type="text" class="form-control" name="txtRutUsu"
placeholder="123456789" onBlur="validaRut(txtRutUsu)"></div>

```

```

</div>

```

```

<div class="form-group">

```

```

<label for="ing_nom_u" class="col-lg-3 control-label">Nombre:</label>

```

```

<div class="col-lg-4"><input type="text" class="form-control" name="txtNomUsu"
placeholder="Ingrese el nombre del nuevo usuario"></div>

```

```

</div>

```

```

<div class="form-group">

```

```

<label for="ing_pass_u" class="col-lg-3 control-label">Contraseña:</label>

```

```

<div class="col-lg-2"><input type="text" class="form-control" name="txtPassUsu"
placeholder="Password"></div>

```

```

</div>

```

```

<div class="form-group">

```

```

<label for="ing_dir_u" class="col-lg-3 control-label">Dirección:</label>

```

```

<div class="col-lg-5"><input type="text" class="form-control" name="txtDirUsu"
placeholder="Ingrese la dirección"></div>

```

```

</div>

```

```

<div class="form-group">

```

```

<label for="ing_fono_u" class="col-lg-3 control-label">Teléfono:</label>

```

```

<div class="col-lg-2"><input type="text" class="form-control" name="txtFonoUsu"
placeholder="Ingrese el teléfono"></div>

```

```

</div>

```

```

<div class="form-group">

```

```

<label for="ing_nom" class="col-lg-3 control-label">Tipo de usuario:</label>

```

```

        <div><input
            type="radio"    class="radioUsu"    value="ADM"    name="radioUsu"
checked>Administrador
        <br>
        <input type="radio" class="radioUsu" value="VEND" name="radioUsu">Vendedor</div>
    </div>

    <div class="form-group">
        <div class="col-lg-offset-1 col-lg-5"><button type="submit" class="btn btn-primary"
onClick="usu_agrega_guardar()" >Agregar</button></div>
        ></div>
    </form>
</div></div></div>
</body>

```

5.1.2. usu_agrega_guardar.php

```

<?php
    include("usu_admin1.php");
    $prim = new sistema;
    $prim->conectar();

    $ru = $_GET['txtRutUsu'];
    $nu = $_GET['txtNomUsu'];
    $pu = $_GET['txtPassUsu'];
    $du = $_GET['txtDirUsu'];
    $fu = $_GET['txtFonoUsu'];
    $tu = $_GET['radioUsu'];

    $insertar= mysql_query("INSERT INTO usuario (cod_usuario, tipo_usuario, password, nombre,
direccion, telefono) VALUES ('$ru', '$tu', '$pu', '$nu', '$du', '$fu')");

    if($insertar)
    {
        print "Usuario agregado correctamente";
    } else
    {
        print "Usuario ya ingresado";
    }
?>

```

5.2. Listar usuarios

5.2.1. usu_admin2.php

```

<?php
    session_start();
    include("usu_admin1.php");
    $tx_tbl_usu = new sistema;
    $tx_tbl_usu->conectar();

```

```

if (isset($_SESSION['ingRut']))
{
?>
<label>Bienvenida <?php print $_SESSION['nombre'];?></label>
<?php
}
if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
{
    include("b_menu_admin.php");
}else
{
    header('Location: http://localhost/TT/inicio.php');
}
?>

<html>
<!doctype html>
<html lang="es">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Administrador de usuarios</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
    <center>
        <script type="text/javascript" src="js/bootstrap.js"></script>
        <div class="container">
            <div class="row-fluid">
                <div class="col-xs-12 col-sm-12 col-md-20 well well-sm">
                    <h4>Administrador de usuarios</h4>
                </div>
            </div>
        </div>
    </center>

    <script src="js/ajax.js" language="javascript"></script>
    <div class = "container">
        <label style="color:#009" style="font:bold" class="col-lg-4 control-label">Buscador:</label>
    </div>
    <div class = "container">
        <label style="color: #936" for="lbl_busca_ped" class="col-lg-4 control-label">1)Por RUT</label>
    </div>
    <div class = "container">
        <label style="color:#936" for="lbl_busca_ped" class="col-lg-4 control-label">2)Por
nombre</label>
    </div>
    <div class = "container">
        <label style="color: #009" for="lbl_busca_ped" class="col-lg-4 control-label">Presione espacio para
mostrar la lista completa</label>

```

```

<div class = "container">
  <div class="input"><input type="col-lg-4" class="text_buscador_usu" name="text_buscador_usu"
id="valorUsu" onKeyUp="busca_usu()" placeholder="Ingrese el dato"></div>
</div>
<div class = "container">
  <div class="tbl_res_usu" id= "tbl_res_usu"></div>
</div>
</body>

```

5.3. Modificar usuarios

5.3.1. usu_mod.php

```

<?php
  session_start();
  include("usu_admin1.php");
  $tx_tbl_usu = new sistema;
  $tx_tbl_usu->conectar();
  $tx_busca_usu = $_GET['tx_busca_usu'];
  $tx_tbl_usu = mysql_fetch_assoc(mysql_query ("SELECT *          FROM usuario WHERE cod_usuario
= '$tx_busca_usu'"));

  if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']="ADM"))
  {

  }else
  {
    header('Location: http://localhost/TT/inicio.php');
  }
?>

<!doctype html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Modifica usuarios</title>
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
  <center>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>

    <div class="container"><div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-
sm">
      <h3>Modifica usuarios</h3>
    </div></div></div>
  </center>

```

```

<div class="container"><div class="row"><div class="span3">

    <form class="form-horizontal" role="form" method = "GET">
        <div class="form-group">
            <label for="ing_rut_usu" class="col-lg-3 control-label">Rut del usuario:</label>
            <div class="col-lg-2"><input type="text" class="form-control" id="cod_usuario" value="<?php
print $tx_tbl_usu['cod_usuario'];?>" disabled></div>
        </div>

        <div class="form-group">
            <label for="lbl_cb" class="col-lg-3 control-label">Nombre:</label>
            <div class="col-lg-2"><input type="text" class="form-control" id="nombre" value="<?php print
$tx_tbl_usu['nombre'];?>"></div>
        </div>

        <div class="form-group">
            <label for="ing_cat" class="col-lg-3 control-label">Contraseña:</label>
            <div class="col-lg-3"><input type="text" class="form-control" id="password" value="<?php
print $tx_tbl_usu['password'];?>"></div>
        </div>

            <div class="form-group">
                <label for="ing_cat" class="col-lg-3 control-label">Dirección:</label>
                <div class="col-lg-3"><input type="text" id="direccion" class="form-control" value="<?php
print $tx_tbl_usu['direccion'];?>"></div>
            </div>

        <div class="form-group">
            <label for="ing_nom" class="col-lg-3 control-label">Teléfono:</label>
            <div class="col-lg-3"><input type="text" class="form-control" id="telefono" value="<?php
print $tx_tbl_usu['telefono'];?>" ></div>
        </div>

        <div class="form-group">
            <label for="ing_nom" class="col-lg-3 control-label">Tipo de usuario:</label>
            <div><input type="radio" class="radioUsu" value="ADM" name="radioUsu" value="ADM"<?php
if ($tx_tbl_usu['tipo_usuario'] == 'ADM') echo "checked";?>>Administrador
            <br>
            <input type="radio" class="radioUsu" value="VEND" name="radioUsu" value="VEN"<?php if
($tx_tbl_usu['tipo_usuario'] == 'VEN') echo "checked";?>>Vendedor</div>
        </div>

        <div class="form-group">
            <div class="col-lg-offset-1 col-lg-5"><button type="submit" class="btn btn-success"
onClick="usu_mod_guardar(<?php print $tx_tbl_usu['cod_usuario'];?>)">Modificar</button></div>
        </div>
    </form>
</div>
</div>
</div>
</body>

```

5.3.2. usu_mod_guardar.php

```

<?php
    include("usu_admin1.php");
    $tx_tbl_usu = new sistema;
    $tx_tbl_usu->conectar();

    $cod_usuario = $_GET['cod_usuario'];
    $nombre = $_GET['nombre'];
    $password = $_GET['password'];
    $direccion = $_GET['direccion'];
    $telefono = $_GET['telefono'];
    $tipo_usuario = $_GET['radioUsu'];

    $actualizar = mysql_query ("UPDATE usuario SET cod_usuario = '$cod_usuario', nombre =
'$nombre', password = '$password', direccion = '$direccion', telefono = '$telefono', tipo_usuario =
'$tipo_usuario' WHERE cod_usuario = '$cod_usuario'");

    if($actualizar)
        {
            print "Usuario modificado correctamente";
        } else
        {
            print "Error al modificar usuario";
        }
?>

```

5.4. Eliminar usuarios

5.4.1. usu_elim.php

```

<?php
    session_start();
    include("usu_admin1.php");
    $tx_tbl_usu = new sistema;
    $tx_tbl_usu->conectar();

    if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']!="ADM"))
    {

    }else
    {
        header('Location: http://localhost/TT/inicio.php');
    }

    $cod_usuario = $_GET['tx_busca_usu'];
    $tx_tbl_usu->EliminarUsu($cod_usuario);
?>

```

6. Ajustes

6.1. Ingresar ajustes de existencias

6.1.1. p_aj_agrega.php

```

<?php
    session_start();
    if (isset($_SESSION['ingRut']))
    {
        ?>
        <label>Bienvenida <?php print $_SESSION['nombre'];?></label>
        <?php
        }
        if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']="ADM"))
        {
            include("b_menu_admin.php");
        }else
        {
            header('Location: http://localhost/TT/inicio.php');
        }
        include("aj_admin1.php");
        $tx_tbl_aj = new sistema;
        $tx_tbl_aj->conectar();
    ?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Agrega un ajuste</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
    <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>
    <script type="text/javascript" src="js/jquery.validate.js"></script>
    <script type="text/javascript" src="js/jquery.numeric.js"></script>

    <center>
    <div class="container">
        <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h3>Agregar
ajustes de control de existencias</h3></div></div>
    </div>
    </center>

    <div class="container"><div class="row"><div class="span3">

        <script type="text/javascript">

```

```

$(function()
{
  $('#form_agrega_aj').validate(
  {
    rules :
    {
      txtCantAj:
      {
        required : true,
        maxlength : 3,
        digits: true,
        min: 1
      },
      txtComenAj:
      {
        required : true,
        maxlength : 40,
      },
    },
    messages :
    {
      txtCantAj :
      {
        required : "Debe ingresar una cantidad.",
        maxlength : "La cantidad debe tener un máximo de 3 caracteres.",
        digits: "Debe ingresar sólo dígitos",
        min : "La cantidad no debe ser igual o menor a 0."
      },
      txtComenAj:
      {
        required : "Debe ingresar un comentario.",
        maxlength : "El largo de la cantidad debe tener un
máximo de 40 caracteres.",
      }
    }
  });
});
</script>

<form class="form-horizontal" id= "form_agrega_aj" name= "form_agrega_aj" role="form"
method = "GET">

  <div class="form-group">
    <label for="ingSubcCat" class="col-lg-3 control-label">Producto:</label>
    <div class="col-lg-3">
      <select class="form-control" id="txtProdAj" name="txtProdAj">
        <?php
                                $resultado = "SELECT * FROM
producto ORDER BY nom_producto asc";

```

```

                                $query = mysql_query
($resultado);
                                while ($fila = mysql_fetch_array($query))
                                {
                                    echo ' <option value=
". $fila["cod_producto"].">' . $fila["nom_producto"].' </option>';
                                    }
                                while ($fila = mysql_fetch_array($resultado));
                                    echo '</select>';
                                ?>

                                </select>
                                </div>
                                </div>

                                <div class="form-group">
                                    <label for="ing_cant" class="col-lg-3 control-label">Cantidad: </label>
                                    <div class="col-lg-1"><input type="text" class="form-control" name="txtCantAj" id =
"txtCantAj" placeholder="xxxx"></div>
                                </div>

                                <div class="form-group">
                                    <label for="ing_nom" class="col-lg-3 control-label">Tipo de usuario: </label>
                                    <div><input type="radio" class="radioAj" value="1" name="radioAj"
checked>Positivo
                                    <br><input type="radio" class="radioAj" value="0" name="radioAj">Negativo</div>
                                </div>

                                <div class="form-group">
                                    <label for="inComenAj" class="col-lg-3 control-label">Comentario: </label>
                                    <div class="col-lg-4"><textarea class="form-control" rows="3" name="txtComenAj"
id="txtComenAj" placeholder="Comente aquí el motivo del ajuste de existencia"></textarea>
                                    </div>
                                </div>

                                <div class="form-group">
                                    <div class="col-lg-offset-1 col-lg-5"><button type="submit" class="btn btn-success"
onClick="aj_agrega_guardar()">Ingresar</button></div>
                                </div>
                                </form>
                                </div>
                                </div>
                                </div>
                                </body>

```

6.1.2. aj_agrega_guardar.php

```

<?php
    session_start();
    if (isset($_SESSION['ingRut']))
    {

```

```

        $ajrutusu = $_SESSION['ingRut'];
    }
?>

<?php
    include("aj_admin1.php");
    $prim = new sistema;
    $prim->conectar();

    $ajcodprod = $_GET['txtProdAj'];
    $ajcant = $_GET['txtCantAj'];
    $ajtipo = $_GET['radioAj'];
    $comen = $_GET['txtComenAj'];

    $insertar = mysql_query("INSERT INTO ajustes (cod_producto, cod_usuario, fecha, cantidad,
    tipo_ajuste, comentario) VALUES ('$ajcodprod', '$ajrutusu', now(), '$ajcant', '$ajtipo', '$comen')");

    if($insertar)
        {
            print "Ajuste agregado correctamente";
            $consulta = mysql_query("SELECT cantidad_producto FROM producto WHERE
cod_producto = '$ajcodprod'");
            $columna = mysql_fetch_array($consulta);
            $cantOrig = $columna['cantidad_producto'];
            //print 'Cantidad original: ';print $cantOrig;

            if ($ajtipo == 0)
                {
                    $cantOrig = ($cantOrig - $ajcant);
                    //print 'Cantidad original despues neg: ';print $cantOrig;

                } else
                {
                    $cantOrig = ($cantOrig + $ajcant);
                    //print 'Cantidad original despues pos: ';print $cantOrig;

                }

            $actualizarProducto = mysql_query ("UPDATE producto SET cantidad_producto
= '$cantOrig' WHERE cod_producto = '$ajcodprod'");
        } else
        {
            print "Error al agregar ajuste";
        }
?>

```

6.2. Listar ajustes de existencias

6.2.1. aj_admin2.php

```

<?php
    session_start();
    include("aj_admin1.php");

```

```

$tx_tbl_usu = new sistema;
$tx_tbl_usu->conectar();

    if (isset($_SESSION['ingRut']))
    {
    ?>
    <label>Bienvenida <?php print $_SESSION['nombre'];?></label>
    <?php
    }
    if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
    {
        include("b_menu_admin.php");
    }else
    {
        header('Location: http://localhost/TT/inicio.php');
    }

?>

<html>
<!doctype html>
<html lang="es">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Administrador de ajustes</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>
</html>

<body>
    <center>
        <script type="text/javascript" src="js/bootstrap.js"></script>
        <div class="container">
            <div class="row-fluid">
                <div class="col-xs-12 col-sm-12 col-md-20 well well-sm">
                    <h4>Administrador de ajustes</h4>
                </div>
            </div>
        </div>
    </center>

    <script src="js/ajax.js" language="javascript"></script>

    <div class = "container">
        <label style="color:#009" style="font:bold" class="col-lg-4 control-label">Buscador por fecha
(aaaa-mm-dd:</label>
    </div>

        <label style="color: #009" for="lbl_busca_ped" class="col-lg-4 control-label">Presione espacio para
mostrar la lista completa</label>
        <div class="input"><input type="col-lg-4" class="text_buscador_aj" name="text_buscador_aj"
id="valorAj" onKeyUp="busca_aj()" placeholder="Ingrese la fecha"></div>

```

```
<label for="lbl_busca_aj" class="col-lg-4 control-label">*Tecla espacio para mostrar todos</label>
  <div class="tbl_res_aj" id="tbl_res_aj"></div>
```

```
</div>
</body>
```

6.3. Eliminar ajustes de existencias

6.3.1. aj_elim.php

```
<?php
include("aj_admin1.php");
$tx_tbl_aj = new sistema;
$tx_tbl_aj->conectar();
$cod_ajuste = $_GET['tx_busca_aj'];
$tx_tbl_aj->EliminarAj($cod_ajuste);
?>
```

7. Facturas

7.1. Ingresar facturas

7.1.1. p_fact_agrega.php

```
<?php
    session_start();
    include("fact_admin1.php");
    $tx_tbl_fact = new sistema;
    $tx_tbl_fact->conectar();

    if (isset($_SESSION['ingRut']))
    {
        ?>
        <label>Bienvenida <?php print $_SESSION['nombre'];?></label>
        <?php
        }
        if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
        {
            include("b_menu_admin.php");
        }else
        {
            header('Location: http://localhost/TT/inicio.php');
        }
    }
```

?>

```

<!doctype html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Agrega facturas</title>
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
  <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" src="js/bootstrap.js"></script>
  <script type="text/javascript" src="js/ajax.js"></script>
  <script type="text/javascript" src="js/jquery.validate.js"></script>
  <script type="text/javascript" src="js/additional-methods.js"></script>

  <script type="text/javascript" src="js/jquery.numeric.js"></script>

  <center>
    <div class="container">
      <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h3>Agrega
una factura</h3></div></div>
    </div>
  </center>

  <div class="container"><div class="row"><div class="span4">
    <script type="text/javascript">
      $(function()
      {
        $('#form_agrega_fact').validate(
        {
          rules :
          {
            txtIvaFact:
            {
              required : true,
              maxlength : 5,
              digits: true,
              min: 1
            },
            txtTotalFact:
            {
              required : true,
              maxlength : 7,
              digits:true,
              min:1
            }
          }
        },
        messages :

```

```

    {
        txtIvaFact:
        {
            required : "Debe ingresar una cantidad",
            maxlength : "La cantidad no debe superar los
5 dígitos",
            digits : "Debe ingresar sólo números",
            min : "El número no puede ser menor o igual
a 0"
        },
        txtTotalFact:
        {
            required : "Debe ingresar una cantidad",
            maxlength : "La cantidad no debe superar los
7 dígitos",
            digits : "Debe ingresar sólo números",
            min : "El número no puede ser menor o igual
a 0"
        },
    },
});
});
</script>

```

```

<form class="form-horizontal" id= "form_agrega_fact" name= "form_agrega_fact"
role="form" method = "GET">

```

```

<div class="form-group">
<label for="ing_fono_u" class="col-lg-3 control-label">Número de factura:</label>
<div class="col-lg-2"><input type="text" class="form-control" id="txtIvaFact"
name="txtIvaFact" placeholder="Ingrese el n°"></div>
</div>

```

```

<div class="form-group">
<label for="ing_codprov" class="col-lg-3 control-label">Código de proveedor:</label>
<div class="col-lg-3">
<select class="form-control" id="txtProvFact" name="txtProvFact">
<?php
                    $resultado = "SELECT * FROM
proveedor ORDER BY nombre_proveedor asc";
                    $query = mysql_query
($resultado);
                    while ($fila = mysql_fetch_array($query))
                    {
                        echo '<option value=
'.'. $fila["cod_proveedor"].'.'>'. $fila["nombre_proveedor"].'</option>';
                    }
                    while ($fila = mysql_fetch_array($resultado));
                    echo '</select>';
                ?>
</select>
</div>
</div>

```

```

    <div class="form-group">
        <label for="ing_fono_u" class="col-lg-3 control-label">IVA:</label>
        <div class="col-lg-2"><input type="text" class="form-control" id="txtIvaFact"
name="txtIvaFact" placeholder="$"></div>
    </div>

    <div class="form-group">
        <label for="ing_fono_u" class="col-lg-3 control-label">Total Factura:</label>
        <div class="col-lg-2"><input type="text" class="form-control" id="txtTotalFact"
name="txtTotalFact" placeholder="$"></div>
    </div>

    <div class="form-group">
        <label for="ing_nom" class="col-lg-3 control-label">Estado de pago:</label>
        <div><input type="radio" class="radioEstadoFact" value="0" id="radioEstadoFact"
name="radioEstadoFact" checked>Pagada
            <br>
            <input type="radio" class="radioEstadoFact" value="1" id="radioEstadoFact"
name="radioEstadoFact">Debe</div>
    </div>

    <div class="form-group">
        <div class="col-lg-offset-1 col-lg-8"><button type="submit" class="btn btn-success"
onClick="fact_agrega_guardar ()">Siguiente</button></div>
    </div>
</form>
</div></div></div>
</body>

```

7.1.2. fact_agrega_guardar.php

```

<?php
    session_start();
    if (isset($_SESSION['ingRut']))
    {
        $factrutusu = $_SESSION['ingRut'];
    }
?>

<?php
    include("fact_admin1.php");
    $prim = new sistema;
    $prim->conectar();

    $factcodprov = $_GET['txtProvFact'];
    $factiva = $_GET['txtIvaFact'];
    $facttotal = $_GET['txtTotalFact'];
    $tipoEstadoFact = $_GET['radioEstadoFact'];

```

```
$insertar= mysql_query("INSERT INTO factura (cod_proveedor, fecha_compra, iva_factura,
total_factura, estado_de_pago) VALUES ('$factcodprov', now(), '$factiva', '$facttotal', '$tipoEstadoFact')");
```

```
if($insertar)
    {
        print "wasappp";
    } else
    {
        print "Error al agregar encabezado de factura";
    }
?>
```

7.2. Listar factura

7.2.1. **fact_admin2.php**

```
<?php
    session_start();
    include("fact_admin1.php");
    $tx_tbl_fact = new sistema;
    $tx_tbl_fact->conectar();

    if (isset($_SESSION['ingRut']))
    {
        ?>
        <label>Bienvenida <?php print $_SESSION['nombre'];?></label>
        <?php
        }
        if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
        {
            include("b_menu_admin.php");
        }else
        {
            header('Location: http://localhost/TT/inicio.php');
        }
    }

?>
<html>
<!doctype html>
<html lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Administrador de facturas</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>
</html>

<body>
<center>
```

```

<script type="text/javascript" src="js/bootstrap.js"> </script>
<div class="container">
  <div class="row-fluid">
    <div class="col-xs-12 col-sm-12 col-md-20 well well-sm">
      <h4>Administrador de facturas</h4>
    </div>
  </div>
</div>
</center>

  <script src="js/ajax.js" language="javascript"></script>

<div class = "container">
  <label style="color: #009" style="font:bold" class="col-lg-4 control-label">Buscador: </label>
</div>
<div class = "container">
  <label style="color: #936" for="lbl_busca_ped" class="col-lg-4 control-label">1)Por N° de
Factura</label>
</div>
<div class = "container">
  <label style="color:#936" for="lbl_busca_ped" class="col-lg-4 control-label">2)Por fecha de
compra (yyyy-mm-dd)</label>
</div>}
<div class = "container">
  <label style="color: #009" for="lbl_busca_ped" class="col-lg-4 control-label">Presione espacio para
mostrar la lista completa</label>

  <div class="input"><input type="col-lg-4" class="txt_buscador_fact" name="txt_buscador_fact"
id="valorFact" onKeyUp="busca_fact()" placeholder="Ingrese un dato"></div>
  <div class="tbl_res_fact" id= "tbl_res_fact"></div>
</div>
</body>

```

7.2.2. Ver detalle de factura

7.2.2.1. fact_det.php

```

<?php
  session_start();
  include("fact_admin1.php");
  $tx_tbl_fact = new sistema;
  $tx_tbl_fact->conectar();

  $tx_busca_fact = $_GET['tx_busca_fact'];

  print $tx_busca_fact;

```

```

?>
<label>Bienvenida <?php print $_SESSION['nombre'];?></label>
<?php

if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
{
}else
{
    header('Location: http://localhost/TT/inicio.php');
}
?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Detalle facturas</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
    <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>
    <script type="text/javascript" src="js/jquery.validate.js"></script>
    <script type="text/javascript" src="js/additional-methods.js"></script>

    <script type="text/javascript" src="js/jquery.numeric.js"></script>

    <center>
    <div class="container">
        <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h3>Detalle
de facturas</h3></div></div>
        </div>
    </center> .

<?php
    $tx_busca_fact = trim($tx_busca_fact);
    $query = (mysql_query ("SELECT * FROM detalle_factura WHERE cod_factura =
'$tx_busca_fact'"));

    if (mysql_num_rows($query) <=0)
    {
        print 'Sin resultados';
    } else
    {
        print '<table class="table table-hover">
        <td>Código factura</td>
        <td>Código producto</td>
        <td>Cantidad</td>
        <td>Precio por unidad</td>

```

```

        <td>Subtotal</td>
    <tr>';

    while ($fila = mysql_fetch_assoc($query))
    {

        print '<td>'.$fila['cod_factura'].'</td>
            <td>'.$fila['cod_producto'].'</td>
            <td>'.$fila['cantidad'].'</td>
            <td>'.$fila['precio_comprado'].'</td>
            <td>'.$fila['total_detalle_factura'].'</td>
            <td>
                <button type="button" class="btn
btn-info btn-xs" onclick="fact_det_mod('.$fila['cod_factura'].');">Modificar</button>
                <button type="button" class="btn btn-
danger btn-xs" onclick="ConfirmarDetFact('.$fila['cod_producto'].');" >Eliminar</button></td>
            </tr>';

    }
    print '</table>';
}

?>

```

7.2.3. Modificar detalle de factura

7.2.3.1. fact_det_mod.php

```

<?php
    session_start();
    include("fact_admin1.php");
    $tx_tbl_fact = new sistema;
    $tx_tbl_fact->conectar();
    $tx_busca_fact = $_GET['tx_busca_fact'];

    $tx_tbl_fact = mysql_fetch_assoc(mysql_query ("SELECT * FROM detalle_factura WHERE cod_factura
= '$tx_busca_fact'"));

    if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']="ADM"))
    {

    }else
    {
        header('Location: http://localhost/TT/inicio.php');
    }

?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Modifica detalle de facturas</title>

```

```

<link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
  <center>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>

    <div class="container"><div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm">
      <h3>Modifica un detalle de factura</h3>
    </div></div></div>
  </center>

  <div class="container"><div class="row"><div class="span3">

    <form class="form-horizontal" role="form" method = "GET">
      <div class="form-group">
        <label for="ing_rut_usu" class="col-lg-3 control-label">Código de factura:</label>
        <div class="col-lg-2"><input type="text" class="form-control" id="cod_factura" value="<?php
print $tx_tbl_fact['cod_factura'];?>" disabled></div>
      </div>

      <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">Código de producto:</label>
        <div class="col-lg-3"><input type="text" class="form-control" id="cod_producto" disabled
value="<?php print $tx_tbl_fact['cod_factura'];?>" disabled></div>
      </div>

      <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">(*)Cantidad:</label>
        <div class="col-lg-3"><input type="text" id="cantidad" class="form-control" value="<?php
print $tx_tbl_fact['cantidad'];?>"></div>
      </div>

      <div class="form-group">
        <label for="lbl_cb" class="col-lg-3 control-label">(*)Precio producto
(unidad):</label>
        <div class="col-lg-2"><input type="text" class="form-control" id="precio_comprado"
value="<?php print $tx_tbl_fact['precio_comprado'];?>"></div>
      </div>

      <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">Subtotal factura:</label>
        <div class="col-lg-3"><input type="text" class="form-control" id="total_detalle_factura"
value="<?php print $tx_tbl_fact['total_detalle_factura'];?>" disabled></div>
      </div>

      <div class="form-group">
        <div class="col-lg-offset-1 col-lg-5"><button type="submit" class="btn btn-success"
onClick="fact_det_mod_guardar(<?php print $tx_tbl_fact['cod_factura'];?>)">Modificar</button></div>
      </div>
    </form>
  </div>
</div>

```

```

    </div>
  </div>
</div>
</body>

```

7.2.3.2. fact_det_mod_guardar.php

```

<?php
    include("fact_admin1.php");
    $tx_tbl_fact = new sistema;
    $tx_tbl_fact->conectar();

    $cod_factura = $_GET['cod_factura'];
    $cod_producto = $_GET['cod_producto'];
    $cantidad = $_GET['cantidad'];
    $precio_comprado = $_GET['precio_comprado'];

    $subtotal = ($precio_comprado * $cantidad);

    print($cod_factura);
    print $cod_producto;
    print $cantidad;
    print $precio_comprado; print $subtotal;

    $actualizar = mysql_query ("UPDATE detalle_factura
                                SET cantidad = '$cantidad',
                                precio_comprado           =
'$precio_comprado',
                                total_detalle_factura = '$subtotal'
                                WHERE cod_factura = '$cod_factura'
                                AND      cod_producto   =
'$cod_producto'");

    if($actualizar)
    {

    print "Detalle de factura modificado correctamente";
    } else
    {
        print "Error al modificar detalle de factura";
    }
?>

```

7.2.4. Eliminar detalle de factura

7.2.4.1. fact_det_elim.php

```

<?php
    include("fact_admin1.php");
    $tx_tbl_fact = new sistema;
    $tx_tbl_fact->conectar();

```

```

$cod_producto = $_GET['tx_busca_fact'];
$tx_tbl_fact->EliminarDetFact($cod_producto);
?>

```

7.3. **Modificar facturas**

7.3.1. **fact_mod.php**

```

<?php
    session_start();
    include("fact_admin1.php");
    $tx_tbl_fact = new sistema;
    $tx_tbl_fact->conectar();
    $tx_busca_fact = $_GET['tx_busca_fact'];
    $tx_tbl_fact = mysql_fetch_assoc(mysql_query ("SELECT *          FROM factura WHERE cod_factura =
'$tx_busca_fact'"));

    if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
    {

    }else
    {
        header('Location: http://localhost/TT/inicio.php');
    }
?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Modifica facturas</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
    <center>
        <script type="text/javascript" src="js/bootstrap.js"></script>
        <script type="text/javascript" src="js/ajax.js"></script>

        <div class="container"><div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-
sm">
            <h3>Modifica factura</h3>
        </div></div></div>
    </center>

    <div class="container"><div class="row"><div class="span3">

        <form class="form-horizontal" role="form" method = "GET">
            <div class="form-group">
                <label for="ing_rut_usu" class="col-lg-3 control-label">Código de factura:</label>

```

```

        <div class="col-lg-2"><input type="text" class="form-control" id="cod_factura" value="<?php
print $tx_tbl_fact['cod_factura'];?>" disabled></div>
    </div>

    <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">Proveedor ingresado:</label>
        <div class="col-lg-3"><input type="text" id="prov_Ant" class="form-control" value="<?php
print $tx_tbl_fact['cod_proveedor'];?>" disabled></div>
    </div>

    <div class="form-group">
        <label for="ing_subcat" class="col-lg-3 control-label">(*)Proveedor nuevo:</label>
        <div class="col-lg-3">
            <select class="form-control" id="cod_proveedor">
                <?php
                    $resultado = "SELECT * FROM proveedor
ORDER BY nombre_proveedor asc";
                    $query = mysql_query ($resultado);
                    while ($fila = mysql_fetch_array($query))
                    {
                        echo '<option value= "'. $fila["cod_proveedor"]."'>'. $fila["nombre_proveedor"].'</option>';
                    }
                    while ($fila = mysql_fetch_array($resultado));
                </select>;
                ?>
            </select>
        </div>
    </div>

    <div class="form-group">
        <label for="lbl_cb" class="col-lg-3 control-label">Fecha de compra:</label>
        <div class="col-lg-2"><input type="text" class="form-control" id="fecha" value="<?php print
$tx_tbl_fact['fecha_compra'];?>" disabled></div>
    </div>

    <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">IVA:</label>
        <div class="col-lg-3"><input type="text" class="form-control" id="iva_factura" value="<?php
print $tx_tbl_fact['iva_factura'];?>"></div>
    </div>

    <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">Total factura:</label>
        <div class="col-lg-3"><input type="text" class="form-control" id="total_factura"
value="<?php print $tx_tbl_fact['total_factura'];?>"></div>
    </div>

    <div class="form-group">
        <label for="ing_nom" class="col-lg-3 control-label">Estado de pago:</label>
        <div><input type="radio" class="radioEstadoFact" value="0" name="radioEstadoFact"
value="0"<?php if ($tx_tbl_fact['estado_de_pago'] == '0') echo "checked";?>>Pagada
        <br>

```

```

        <input type="radio" class="radioEstadoFact" value="1" name="radioEstadoFact"
value="1"><?php if ($tx_tbl_fact['estado_de_pago'] == '1') echo "checked";?>>Se debe
        </div>

        <div class="form-group">
            <div class="col-lg-offset-1 col-lg-5"><button type="submit" class="btn btn-success"
onClick="fact_mod_guardar(<?php print $tx_tbl_fact['cod_factura'];?>)">Modificar</button></div>
            </div>
        </form>
    </div>
</div>
</div>
</body>

```

7.3.2. fact_mod_guardar.php

```

<?php
include("fact_admin1.php");
$tx_tbl_fact = new sistema;
$tx_tbl_fact->conectar();

$cod_factura = $_GET['cod_factura'];
$cod_proveedor = $_GET['cod_proveedor'];
$fecha_compra = $_GET['fecha_compra'];
$iva_factura = $_GET['iva_factura'];
$total_factura = $_GET['total_factura'];
$estadoFact = $_GET['radioEstadoFact'];

$actualizar = mysql_query ("UPDATE factura SET
    cod_factura = '$cod_factura', cod_proveedor = '$cod_proveedor', fecha_compra =
'$fecha_compra', iva_factura = '$iva_factura', total_factura = '$total_factura', estado_de_pago =
'$estadoFact' WHERE cod_factura = '$cod_factura'");

if($actualizar)
    {
        print "Factura modificada correctamente";
    } else
    {
        print "Error al modificar factura";
    }
?>

```

7.4. Eliminar facturas

7.4.1. fact_elim.php

```

<?php
include("fact_admin1.php");

```

```

$tx_tbl_fact = new sistema;
$tx_tbl_fact->conectar();
$cod_factura = $_GET['tx_busca_fact'];
$tx_tbl_fact->EliminarFact($cod_factura);
?>

```

8. Ventas

8.1. Ingresar ventas

8.1.1. venta.php

```

<?php
include("fact_admin1.php");
$tx_tbl_fact = new sistema;
$tx_tbl_fact->conectar();
$cod_factura = $_GET['tx_busca_fact'];
$tx_tbl_fact->EliminarFact($cod_factura);
?>

```

8.1.2. ven_3.php

```

<?php
session_start();
include("ven_admin1.php");
$tx_tbl_ven = new sistema;
$tx_tbl_ven->conectar();

if (isset($_SESSION['ingRut']))
{
    $venrutusu = $_SESSION['ingRut'];
}

$rs_ven = mysql_query("SELECT MAX(cod_venta) AS ultimaven FROM venta");
if ($row = mysql_fetch_row($rs_ven))
{
    $ultimaven = trim($row[0]);
}

$fila= mysql_fetch_assoc(mysql_query("SELECT SUM(total_detalle_venta) AS suma_total FROM
detalle_venta WHERE cod_venta = '$ultimaven'"));
$sumatotal = $fila['suma_total'];

?>

<!doctype html>
<html lang="es">
<head>

```

```

<meta charset="UTF-8">
<title>lista detalle de venta</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css">
<style type="text/css">
body {
  background-image: url(img/fondo.png);
}
</style>
</head>

<body>
  <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" src="js/bootstrap.js"></script>
  <script type="text/javascript" src="js/ajax.js"></script>
  <script type="text/javascript" src="js/jquery.validate.js"></script>
  <script type="text/javascript" src="js/additional-methods.js"></script>

  <script type="text/javascript" src="js/jquery.numeric.js"></script>

  <center>
  <div class="container">
    <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h3>Lista
detalle de venta</h3></div></div>
    </div>
  </center>

  <div class="container"><div class="row"><div class="span4">
    <form class="form-horizontal" id="form_agrega_ven3" name="form_agrega_ven3"
role="form" method="GET">

    <div class="form-group">

      <label for="ing_fono_u" class="col-lg-3 control-label">Total de compra:</label>
      <div class="col-lg-2"><input type="text" class="form-control" id="txtTotalVen"
name="txtTotalVen" value="<?php print $sumatotal ?>" readonly></div>
    </div>

    <div class="form-group">
      <label for="ing_nom" class="col-lg-3 control-label">Modo de pago:</label>
      <div><input type="radio" class="radioPago" value="EF" id="radioPago" name="radioPago"
checked>Efectivo
      <br>
      <input type="radio" class="radioPago" value="DB" id="radioPago"
name="radioPago">Débito</div>
    </div>

    <div class="form-group">
      <div class="col-lg-offset-1 col-lg-8"><button type="submit" class="btn btn-success"
onClick="ven_3_guardar()">Confirmar venta</button></div>
    </div>
    <div class="form-group">
      <div class="col-lg-offset-1 col-lg-8"><button type="submit" class="btn btn-
warning"><a href="javascript:history.back(1)">Volver Atrás</a></button></div>

```

```

                <div class="tbl_res_det_ven" id= "tbl_res_ven_det"></div>
            </div>
        </div>
    </form>
</div></div></div>
</body>

```

8.1.3. ven_3_agrega_guardar.php

```

<?php
include("ven_admin1.php");
$tx_tbl_ven = new sistema;
$tx_tbl_ven->conectar();

$total = $_GET['txtTotalVen'];
$tipo_pago = $_GET['radioPago'];

$rs_ven = mysql_query("SELECT MAX(cod_venta) AS ultimaven FROM venta");
if ($row = mysql_fetch_row($rs_ven))
{
    $ultimaven = trim($row[0]);
}

$rs_det_bol = mysql_query("SELECT MAX(num_boleta) AS ultimabol FROM venta");
if ($row = mysql_fetch_row($rs_det_bol))
{
    $ultimabol = trim($row[0]);
}

$rs_caja = mysql_query("SELECT MAX(cod_dia_caja) AS ultimacaja FROM caja");
if ($row = mysql_fetch_row($rs_caja))
{
    $ultimacaja = trim($row[0]);
}

if ($total >= 180)
{
    $bol = $ultimabol + 1;

    date_default_timezone_set("America/Santiago");
    $h = time();
    $hora = date("H:i", $h);

    $actualizar = mysql_query ("UPDATE venta SET num_boleta = '$bol', hora = '$hora',
tipo_pago = '$tipo_pago', total_venta = '$total' WHERE cod_venta = '$ultimaven'");

    if ($tipopago = 'EF')
    {
        $actualizacaja = mysql_query ("UPDATE caja SET ventas_con_boleta = 'total',
ventas_efectivo = '$total' WHERE cod_dia_caja = '$ultimacaja'");
    }else

```

```

        {
            $actualizacaja = mysql_query ("UPDATE caja SET ventas_sin_boleta = 'total',
ventas_debito = '$total' WHERE cod_dia_caja = '$ultimacaja'");
        }
    }else
    {

        date_default_timezone_set("America/Santiago");
        $h = time();
        $hora = date("H:i", $h);

        $actualizar = mysql_query ("UPDATE venta SET hora = '$hora', tipo_pago = '$tipo_pago',
total_venta = '$total' WHERE cod_venta = '$ultimaven'");

        if ($tipo_pago = 'EF')
        {
            $actualizacaja = mysql_query ("UPDATE caja SET ventas_con_boleta = 'total',
ventas_efectivo = '$total' WHERE cod_dia_caja = '$ultimacaja'");
        }else
        {
            $actualizacaja = mysql_query ("UPDATE caja SET ventas_sin_boleta = 'total',
ventas_debito = '$total' WHERE cod_dia_caja = '$ultimacaja'");
        }
    }
    if($actualizar)
    {
        print "Venta realizada exitosamente";
    } else
    {
        print "Venta no realizada";
    }
}

?>

```

8.1.4. ven_vender.php

```

<?php
    session_start();
    include("ven_admin1.php");
    $tx_tbl_prod_ven = new sistema;
    $tx_tbl_prod_ven->conectar();

    $tx_busca_prod_ven= $_GET['tx_busca_prod_ven'];

    $rs = mysql_query("SELECT MAX(cod_venta) AS ultimaventa FROM venta");
    if ($row = mysql_fetch_row($rs))
    {
        $ultimaventa = trim($row[0]);
    }
}

```

```

    }

?>

<!doctype html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>VENTAS</title>
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
  <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" src="js/bootstrap.js"></script>
  <script type="text/javascript" src="js/ajax.js"></script>
  <script type="text/javascript" src="js/jquery.validate.js"></script>
  <script type="text/javascript" src="js/additional-methods.js"></script>

  <script type="text/javascript" src="js/jquery.numeric.js"></script>

  <center>
  <div class="container">
    <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h6>Ingresa
productos</h6></div></div>
    </div>
  </center> .

  <div class="container"><div class="row"><div class="span4">
  <script type="text/javascript">
    $(function()
    {
      $('#form_agrega_prod_ven').validate(
      {
        rules :
        {
          cantDetVen:
          {
            required : true,
            maxlength : 3,
            digits: true,
            min: 1
          },
        },
        messages :
        {
          cantDetVen:
          {
            required : "Debe ingresar una cantidad",

```

```

3 dígitos",
a 0"

maxlength : "La cantidad no debe superar los
digits: "Debe ingresar sólo números",
min: "El número no puede ser menor o igual

},

},
});
});
</script>
<form class="form-horizontal" id="form_agrega_prod_ven" name="
"form_agrega_prod_ven" role="form" method = "GET">
<?php
        $fila = mysql_fetch_assoc(mysql_query ("SELECT * FROM
venta WHERE cod_venta = '$ultima_venta'"));
        ?>
        <div class="form-group">
            <label for="ing_fono_u" class="col-lg-3 control-label">Número de venta:</label>
            <div class="col-lg-2"><input type="text" class="form-control" id="vencodven"
name="vencodven" value="<?php print $fila['cod_venta'];?>" disabled></div>
        </div>

        <div class="form-group">
            <label for="ing_codprov" class="col-lg-3 control-label">Producto:</label>
            <div class="col-lg-2"><input type="text" class="form-control" id="cod_producto"
name="cod_producto" value="<?php print $tx_busca_prod_ven;?>" disabled></div>
        </div>

        <div class="form-group">
            <label for="ing_fono_u" class="col-lg-3 control-label">Cantidad:</label>
            <div class="col-lg-3"><input type="text" class="form-control" id="cantDetVen"
name="cantDetVen" placeholder="Cantidad a vender"><button type="button" class="btn btn-warning"
onClick="venDetProd_agrega_guardar()">Agregar al detalle</span></button>
        </div>

    </span>
</div></div></div>
</body>

```

8.1.5. venDetProd_agrega_guardar.php

```

<?php
    session_start();

    include("ven_admin1.php");
    $prim = new sistema;
    $prim->conectar();
    $vencodven = $_GET['vencodven'];
    $vencodprod = $_GET['cod_producto'];

```

```

$vendetcant = $_GET['cantDetVen'];

$query = mysql_query("SELECT precio_unitario FROM producto where cod_producto =
$vencodprod");
if($query)
{
    $fila = mysql_fetch_assoc($query);
    $precio = $fila['precio_unitario'];
    $total_detalle = $precio * $vendetcant;
    $insertar1= mysql_query("INSERT INTO detalle_venta
(cod_venta, cod_producto, precio_prod, cantidad, total_detalle_venta) VALUES
('$vencodven', '$vencodprod', $precio,
'$vendetcant', $total_detalle)");

    if ($insertar1) //si no esta rep
    {
        $actualizar = mysql_query ("UPDATE detalle_venta
SET precio_prod = '$precio', total_detalle_venta =
'$total_detalle'
WHERE cod_venta = '$vencodven' AND cod_producto
= '$vencodprod'");

        if ($actualizar)
        {
            print "Ingreso de producto al detalle realizado con éxito";
            $query = mysql_query("SELECT cantidad_producto FROM
producto where cod_producto = $vencodprod");
            $fila = mysql_fetch_assoc($query);
            $cantidad_producto = $fila['cantidad_producto'];
            $cantidad_nueva = $cantidad_producto - $vendetcant;
            $actualizarProd = mysql_query ("UPDATE producto SET
cantidad_producto = '$cantidad_nueva' WHERE cod_producto = '$vencodprod'");
        }else {print "No actualiza BD";} //sacar mensaje
    }else {print "Producto ya ingresado";} //repetido
} //query}}

?>

```

8.1.6. ven_det_det_mod.php

```

<?php
    session_start();
    include("ven_admin1.php");
    $tx_tbl_detven = new sistema;
    $tx_tbl_detven->conectar();
    $tx_busca_detven = $_GET['tx_busca_detven'];

    $tx_tbl_detven = mysql_fetch_assoc(mysql_query ("SELECT * FROM detalle_venta WHERE cod_venta
= '$tx_busca_detven'"));

?>

```

```

<!doctype html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Modifica detalle de ventas</title>
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
  <center>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>

    <div class="container"><div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm">
      <h3>Modifica un detalle de ventas</h3>
    </div></div></div>
  </center>

  <div class="container"><div class="row"><div class="span3">

    <form class="form-horizontal" role="form" method = "GET">
      <div class="form-group">
        <label for="ing_rut_usu" class="col-lg-3 control-label">Código de venta:</label>
        <div class="col-lg-2"><input type="text" class="form-control" id="cod_venta" value="<?php
print $tx_tbl_detven['cod_venta'];?>" disabled></div>
      </div>

      <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">Código de producto:</label>
        <div class="col-lg-3"><input type="text" class="form-control" id="cod_producto" disabled
value="<?php print $tx_tbl_detven['cod_producto'];?>" disabled></div>
      </div>

      <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">(*) Cantidad:</label>
        <div class="col-lg-3"><input type="text" id="cantidad" class="form-control" value="<?php
print $tx_tbl_detven['cantidad'];?>"></div>
      </div>

      <div class="form-group">
        <label for="lbl_cb" class="col-lg-3 control-label">(*) Precio de producto
(unidad):</label>
        <div class="col-lg-2"><input type="text" class="form-control" id="precio_prod" value="<?php
print $tx_tbl_detven['precio_prod'];?>"></div>
      </div>

      <div class="form-group">
        <label for="ing_cat" class="col-lg-3 control-label">Subtotal pedido:</label>
        <div class="col-lg-3"><input type="text" class="form-control" id="total_detalle_venta"
value="<?php print $tx_tbl_detven['total_detalle_venta'];?>" disabled></div>
      </div>

```

```

    <div class="form-group">
        <div class="col-lg-offset-1 col-lg-5"><button type="submit" class="btn btn-success"
onClick="ven_det_det_mod_guardar(<?php
$tx_tbl_detven['cod_venta'];?>)">Modificar</button></div>
        </div>
    </form>
</div>
</div>
</div>
</body>

```

8.1.7. ven_det_det_elim.php

```

<?php
include("ven_admin1.php");
$tx_tbl_ven = new sistema;
$tx_tbl_ven->conectar();
$cod_producto = $_GET['tx_busca_detven'];
$tx_tbl_ven->EliminarDetDetVen($cod_producto);
?>

```

8.1.8. ven_det_det_mod_guardar.php

```

<?php
include("ven_admin1.php");
$tx_tbl_detven = new sistema;
$tx_tbl_detven->conectar();

$cod_venta = $_GET['cod_venta'];
$cod_producto = $_GET['cod_producto'];
$cantidad = $_GET['cantidad'];
$precio_prod = $_GET['precio_prod'];

$total_detalle_venta = ($precio_prod * $cantidad);

$actualizar = mysql_query("UPDATE detalle_venta
                                SET cantidad = $cantidad,
                                precio_prod =
$precio_prod,
                                total_detalle_venta =
$total_detalle_venta
                                WHERE cod_venta = $cod_venta AND
                                cod_producto = $cod_producto");

if($actualizar)
{
    print "Detalle de venta modificado correctamente";

    //ÑAAAAAA
    $query = mysql_query("SELECT cantidad_producto FROM producto where
cod_producto = $cod_producto");

```

```

        $fila = mysql_fetch_assoc($query);
        $cantidad_producto = $fila['cantidad_producto'];

        if ($cantidad <= $cantidad_producto)
            {
                $cantidad_nueva = $cantidad_producto -
$cantidad;
                //$cantidad_calculada = $cantidad_producto-
$cantidad_nueva;
            }
            else
            {
                print ("No hay suficiente stock");
            }
            $actualizarProd = mysql_query ("UPDATE producto SET cantidad_producto =
'$cantidad_nueva' WHERE cod_producto = '$cod_producto'");
        }
    ?>

```

8.2. Listar ventas

8.2.1. vent_admin2.php

```

<?php
    session_start();
    include("vent_admin1.php");
    $tx_tbl_ped = new sistema;
    $tx_tbl_ped->conectar();

    if (isset($_SESSION['ingRut']))
    {
        ?>
        <label>Bienvenida <?php print $_SESSION['nombre'];?></label>
        <?php
        }
        if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
        {
            include("b_menu_admin.php");
        }else
        {
            header('Location: http://localhost/TT/inicio.php');
        }
    }

?>
<html>
<!doctype html>
<html lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

```

```

<title>Administrador de pedidos</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>
</html>

<body>
<center>
<script type="text/javascript" src="js/bootstrap.js"> </script>
<div class="container">
<div class="row-fluid">
<div class="col-xs-12 col-sm-12 col-md-20 well well-sm">
<h4>Consulta de ventas</h4>
</div>
</div>
</div>
</center>

<script src="js/ajax.js" language="javascript"></script>

<div class = "container">
<label for="lbl_busca_vent" class="col-lg-4 control-label">Buscador por fecha:</label>
<div class="input"><input type="col-lg-4" class="txt_buscador_vent" name="txt_buscador_vent"
id="valorVent" onKeyUp="busca_vent()" placeholder="Ingrese la fecha"></div>
<div class="tbl_res_vent" id= "tbl_res_vent"></div>
</div>
</body>

```

8.2.2. Ver detalle de venta

8.2.2.1. vent_det.php

```

<?php
    session_start();
    include("vent_admin1.php");
    $tx_tbl_vent = new sistema;
    $tx_tbl_vent->conectar();

    $tx_busca_vent = $_GET['tx_busca_vent'];

    if ((isset($_SESSION['ingRut']))&&($_SESSION['tipo_usuario']=="ADM"))
    {
    }else
    {
        header('Location: http://localhost/TT/inicio.php');
    }
?>

<!doctype html>
<html lang="es">

```

```

<head>
  <meta charset="UTF-8">
  <title>Detalle pedidos</title>
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
</head>

<body>
  <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" src="js/bootstrap.js"></script>
  <script type="text/javascript" src="js/ajax.js"></script>
  <script type="text/javascript" src="js/jquery.validate.js"></script>
  <script type="text/javascript" src="js/additional-methods.js"></script>

  <script type="text/javascript" src="js/jquery.numeric.js"></script>

  <center>
  <div class="container">
    <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h3>Detalle
de ventas</h3></div></div>
    </div>
  </center> .

<?php
  $tx_busca_vent = trim($tx_busca_vent);
  $query = (mysql_query ("SELECT d.cod_venta, d.cod_producto, d.cantidad,d.precio_prod,
d.total_detalle_venta, p.cod_producto, p.nom_producto FROM detalle_venta d, producto p WHERE
p.cod_producto = d.cod_producto AND cod_venta = '$tx_busca_vent'"));

  if (mysql_num_rows($query) <=0)
  {
    print 'Sin resultados';
  } else
  {
    print '<table class="table table-hover">
<td>Código venta</td>
      <td>Nombre del producto</td>
      <td>Cantidad</td>
      <td>Precio por unidad</td>
<td>Subtotal</td>
<tr>';

    while ($fila = mysql_fetch_assoc($query))
    {
      print '<td>'. $fila['cod_venta']. '</td>
      <td>'. $fila['nom_producto']. '</td>
      <td>'. $fila['cantidad']. '</td>

<td>'. $fila['precio_pedido_unidad']. '</td>

      <td>'. $fila['subtotal_pedido']. '</td>
      <td>

</tr>';

```

```

    }
    print '</table>';
}

```

?>

9. Caja

9.1. Cierre de caja

9.1.1. cierre.php

```

<?php
    session_start();
    include("ven_admin1.php");
    $tx_tbl_caja = new sistema;
    $tx_tbl_caja->conectar();

    $rs_det_caja = mysql_query("SELECT MAX(cod_dia_caja) AS ultimacaja FROM caja");
    if ($row = mysql_fetch_row($rs_det_caja))
    {
        $ultimacaja = trim($row[0]);
    }
?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Z</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
    <style type="text/css">
    body {
        background-image: url(img/fondo.png);
    }
    </style>
</head>

<body>
    <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.js"></script>
    <script type="text/javascript" src="js/ajax.js"></script>
    <script type="text/javascript" src="js/jquery.validate.js"></script>
    <script type="text/javascript" src="js/additional-methods.js"></script>

    <script type="text/javascript" src="js/jquery.numeric.js"></script>

    <center>
    <div class="container">
        <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-
sm"><h3>Z</h3></div></div>

```

```

</div>
</center>

<div class="tbl_res_det_ven" id= "tbl_res_ven_det"></div>

<?php
$query = (mysql_query ("SELECT * FROM caja WHERE cod_dia_caja = '$ultimacaja'"));

if (mysql_num_rows($query) <=0)
{
    print 'Sin resultados';
} else
    {

        print '<table id="tbl_res_ven_det" class="table table-hover">
        <td>Día de caja</td>
            <td>Efectivo inicial</td>
            <td>Ventas sin boleta</td>
            <td>Ventas con boleta</td>
        <td>Total giros</td>
            <td>Ventas efectivos</td>
            <td>Ventas débitos</td>
            <td>Total efectivo</td>

        <tr>;

        while ($fila = mysql_fetch_assoc($query))
        {

            print '<td>'.$fila['cod_dia_caja'].'</td>
                <td>'.$fila['efectivo_inicial'].'</td>

            <td>'.$fila['ventas_sin_boleta'].'</td>
                <td>'.$fila['ventas_con_boleta'].'</td>
                <td>'.$fila['total_giro'].'</td>

            <td>'.$fila['ventas_efectivo'].'</td>
                <td>'.$fila['ventas_debito'].'</td>
                <td>'.$fila['total_caja'].'</td>
                <td>

            </tr>;

        }
        print '</table>';
    }

    $actualizar = mysql_query("UPDATE caja SET estado_caja = 'D' WHERE cod_dia_caja =
'$ultimacaja'");

?>

```

```

<div class="form-group">
  <div class="col-lg-offset-1 col-lg-8"><button type="submit" class="btn btn-danger" onClick="<?php
header('Location: http://localhost/TT/inicio.php')?>">Salir</button>
  </div>
</div>
</div>

```

9.2. Historial de movimiento de caja

9.2.1. historial.php

```

<?php
  session_start();
  include("ven_admin1.php");
  $tx_tbl_caja = new sistema;
  $tx_tbl_caja->conectar();

  $rs_det_caja = mysql_query("SELECT MAX(cod_dia_caja) AS ultimacaja FROM caja");
  if ($row = mysql_fetch_row($rs_det_caja))
  {
    $ultimacaja = trim($row[0]);
  }

?>

<!doctype html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Z</title>
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
  <style type="text/css">
  body {
    background-image: url(img/fondo.png);
  }
</style>
</head>

<body>
  <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" src="js/bootstrap.js"></script>
  <script type="text/javascript" src="js/ajax.js"></script>
  <script type="text/javascript" src="js/jquery.validate.js"></script>
  <script type="text/javascript" src="js/additional-methods.js"></script>

  <script type="text/javascript" src="js/jquery.numeric.js"></script>

  <center>
  <div class="container">

```

```

    <div class="row-fluid"><div class="col-xs-12 col-sm-12 col-md-20 well well-sm"><h3>Historial
de movimiento de caja</h3></div></div>
    </div>
</center>

```

```

<div class="tbl_res_det_ven" id= "tbl_res_ven_det"></div>

```

```

<div class = "container">
    <label for="lbl_busca_cod" class="col-lg-4 control-label">Buscador por fecha(dd-mm-
aaaa):</label>
    <div class="input"><input type="col-lg-4" class="text_buscador" name="text_buscador"
id="valor" onKeyUp="busqueda_vent()" placeholder="Ingrese el dato"></div>
    <div class="tabla_result_vent" id= "tabla_result_vent"></div>
</div>

```

```

<?php
$query = (mysql_query ("SELECT * FROM caja"));

if (mysql_num_rows($query) <=0)
{
    print 'Sin resultados';
} else
    {

        print '<table id="tbl_res_ven_det" class="table table-hover">
        <td>Día de caja</td>
            <td>Efectivo inicial</td>
            <td>Ventas sin boleta</td>
            <td>Ventas con boleta</td>
        <td>Total giros</td>
            <td>Ventas efectivos</td>
            <td>Ventas débitos</td>
            <td>Total efectivo</td>

        <tr>';

        while ($fila = mysql_fetch_assoc($query))
        {

            print '<td>'.$fila['cod_dia_caja'].'</td>
                <td>'.$fila['efectivo_inicial'].'</td>

            <td>'.$fila['ventas_sin_boleta'].'</td>
                <td>'.$fila['ventas_con_boleta'].'</td>
                <td>'.$fila['total_giro'].'</td>

            <td>'.$fila['ventas_efectivo'].'</td>
                <td>'.$fila['ventas_debito'].'</td>
                <td>'.$fila['total_caja'].'</td>
                <td>

            </tr>';

```

```
    }  
    print '</table>';  
}
```

```
?>
```

```
<div class="form-group">  
  <div class="col-lg-offset-1 col-lg-8"><button type="submit" class="btn btn-success"  
onClick="">Volver</button>  
  </div>  
  </div>  
</div>
```