

# INSTITUTO TECNOLÓGICO DE MINATITLÁN

INGENIERÍA EN SISTEMAS COMPUTACIONALES

“Proyecto SIABA”

PRESENTA:

- 18230721 - Dominguez Tovar Israel David
- 18230724 - Gamas Michael
- 18230748 - Pozos Miranda Luis Eduardo
- 17230772 - Rodriguez Gonzalez Aleydis Gabriel



Minatitlán, Ver., a 20 de noviembre del 2021

## INDICE

<b>ANÁLISIS.....</b>	<b>1</b>
<b>PRESENTACIÓN DEL PROYECTO.....</b>	<b>1</b>
INFORMACIÓN DEL PROYECTO.....	1
<b>ANÁLISIS DE RECOLGIDA DE LA INFORMACIÓN.....</b>	<b>2</b>
ENTREVISTA.....	2
REVISIÓN DE REGISTROS.....	2
<b>REQUISITOS FUNCIONALES Y NO FUNCIONALES.....</b>	<b>6</b>
LISTA DE REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.....	6
MATRIZ DE TRAZABILIDAD.....	7
DOCUMENTACIÓN DE REQUISITOS.....	9
<b>DIAGRAMA DEL DOMINIO (DIAGRAMA DE CLASES).....</b>	<b>15</b>
DC REPUE.....	15
<b>MODELO DEL NEGOCIO (DIAGRAMAS DE CASO DE USO).....</b>	<b>25</b>
DCU CONSULTAS.....	25
DCU GESTIONAR INVENTARIO.....	28
DCU INICIO DE SESIÓN.....	32
DCU REGISTRO.....	35
<b>ESTUDIO DE FACTIBILIDAD.....</b>	<b>38</b>
FACTIBILIDAD ECONÓMICA.....	38
FACTIBILIDAD TÉCNICA.....	39
FACTIBILIDAD OPERATIVA.....	40
<b>COSTO-BENEFICIO DEL SISTEMA.....</b>	<b>41</b>
<b>DISEÑO.....</b>	<b>42</b>
<b>DICCIONARIO DE DATOS.....</b>	<b>42</b>
<b>MODELO E-R-E (NORMALIZADO).....</b>	<b>45</b>
<b>MODELO RELACIONAL.....</b>	<b>46</b>
<b>JUSTIFICACIÓN.....</b>	<b>47</b>
<b>SCRIPT DE BASE DE DATOS.....</b>	<b>47</b>
TABLAS PADRE.....	47

TABLAS HIJO.....	49
DISEÑO ARQUITECTÓNICO.....	50
DISEÑO DE INTERFAZ DE USUARIO.....	51
<b>DESARROLLO.....</b>	<b>62</b>
<b>CÓDIGOS DE SECCIONES PRINCIPALES.....</b>	<b>62</b>
LOGIN.....	62
PAGINA DE INICIO.....	64
MODULO DE USUARIOS.....	65
<b>CAPTURAS DE FUNCIONALIDAD.....</b>	<b>71</b>
INDEX.....	71
PAGINA DE INICIO (HOME).....	71
PAGINA DE USUARIOS.....	72
AGREGAR UN USUARIO.....	72
EDICIÓN DE UN REGISTRO.....	73
RESULTADO DE EDICIÓN.....	74
ELIMINAR REGISTRO.....	74
<b>ANEXOS.....</b>	<b>77</b>
<b>ANEXO A.....</b>	<b>77</b>

## ANÁLISIS

### Presentación del proyecto

#### Información del proyecto

**Titulo formal del proyecto:** Sistema de abarrotes

**Titulo informal del proyecto:** SIABA

**Objetivo general:** Diseñar e implementar un sistema que permita gestionar de forma eficiente las entradas y salidas de productos dentro de la microempresa "Pily" para la automatización del inventario.

#### Objetivos específicos:

- Analizar los requerimientos de sistematización dentro de la microempresa "Pily" para la gestión de inventarios del almacén.
- Diseñar una base de datos y la interfaz del sistema de inventarios dentro de la microempresa "Pily" para el gestionamiento de los productos.
- Desarrollar una aplicación para crear, actualizar y eliminar la información de la base de datos, para mejorar la eficacia en el proceso de ventas o pedidos de los productos.
- Implementar el sistema SIABA para el gestionamiento del inventario y las entradas y salidas de los productos en la microempresa "Pily".

#### Descripción del proyecto:

La microempresa "Pily" lleva un control manual, donde es anotado todo en hojas recicladas, las hojas tienen de encabezado los productos en existencia dentro del almacén y el total de los productos que hay (stock).

Va comparando los productos que hay con los que se vendieron, los empleados tienen el deber de anotar todo lo que vendieron en cantidad y con su respectivo precio, para que el administrador sume el total de lo vendido y verifique que este

coincida con el dinero que está en caja al momento de hacer el corte, dicho corte es realizado diariamente.

En el control de proveedores, cada proveedor le proporciona un ticket al administrador, donde se muestra los productos y la cantidad que el proveedor está entregando, también se lleva el control de los tickets clasificándolos en una libreta (Barcel, Sabritas, Ricolino, Coca-Cola, etc).

Todos los artículos que ingresen se van al almacén por lo que tienen que ser agregados en la hoja donde maneja el stock.

## **Análisis de recopilación de la información.**

### **Entrevista**

Se aplicó una entrevista al administrador de la microempresa “Pily” **Ver anexo A**. Analizamos que el cliente tiene problemas para administrar su negocio de manera eficaz y eficiente, actualmente cuenta con un formato que elaboró en papel y lápiz, dónde apunta cantidad, producto y costo, este mismo es el formato para el conteo de inventario.

Ademas, tiene un formato para las ventas del negocio, ahí lleva el registro diario pagina por pagina. Al hacer la entrevista nos dimos cuenta de las necesidades de nuestro cliente, recopilamos información para el desarrollo del sistema.

Ahora sabemos como se hacen todos los procesos del negocio, quien es el responsable de esos procesos, que información debe almacenar y información acerca de la interfaz del sistema.

### **Revisión de registros**

En la figura 1 notamos que el cliente en hojas recicladas, lleva un formato de inventario, en dónde en el encabezado, él maneja el nombre del artículo y debajo, va poniendo el número de artículos que se encuentran al finalizar el día. El cliente cuenta lo de su almacén y lo que tiene en exhibición.

ESQUEMA DE MAYOR													NOMBRE DEL ALUMNO		
Inventario													FECH		
chips	takis	1/20 9/0	tostadas	Tosta chos	wapa	runes	Chipo ties	Big mix	pop	Caca huaste	duice	Mand pandi	Galle tas	Florentino Poronllas	
12	12	2	2	2	3	3	2	8	2	15	47	5	20	1	2
12	12	2	2	2	3	3	2	8	2	14	44	5	20	4	2
11	12	2	-	2	3	3	2	8	2	13	43	5	19	4	2
12	12	2	4	3	3	3	3	8	1	15	51	3	10	4	26
11	11	1	3	3	3	3	3	8	1	14	49	3	9	4	25
11	10	1	3	3	3	2	3	8	2	13	46	3	20	3	4
11	10	1	1	3	3	2	3	6	2	13	44	3	18	3	4

Figura 1. Registro de inventario.

De ahí va comparando con su lista de ventas diarias para hacer un arqueo de las ventas diarias, y que cuadre con la existencia del producto, en la Figura 2, podemos notar como lleva a cabo sus registros de ventas diarias.

Sabado 7 Mayo 2021	
Sabritas y Galletas	Refrescos y Aguas ✓
1 Sabritas = 13 ✓	2 Vasos 10 ✓
1 emperador = 13 ✓	1 Agua 1 L. = 12 ✓
1 tostadas = 10 ✓	1 mandel Saber = 20 ✓
1 rollis = 13 ✓	1 coca 400 = 10 ✓
1 papas mordidas = 13 ✓	1 Agua 1L = 12 ✓
1 big mix = 12 ✓	1 coca 1/4 = 19 ✓
1 Paletazo = 13 ✓	1 coca 2L R = 28 ✓
1 tostada = 10 ✓	1 coca 2.5L R = 28 ✓
1 dulce = 1 ✓	1 Seneca = 18 ✓
1 galletas = 13 ✓	1 coca 8L R = 33 ✓
<b>\$ 832</b>	4000 500m Vida = 48 ✓
	1 ron 600 = 15 ✓
	1 coca 600 = 15 ✓
	1 coca 1/4 = 19 ✓
	1 mandet = 20 ✓
	1 coca 3L = 33 ✓
	1 coca 2.5 = 28 ✓
	Cigares 12 = 10 ✓
	1 coctilla M.R = 70 ✓
	6 Vasos = 6 ✓
	2 Vasos = 2 ✓
	1 coca 1/4 = 19 ✓
	1 mandet = 20 ✓
	1 taza 2.5 = 33 ✓
	1 Coctilla = 70 ✓
	1 borsal = 5 ✓
	1 coca 2.5 = 28 ✓
	1 coca 3L = 33 ✓
	1 taza en Cm = 22 ✓
	Tostadas \$ 13 ✓
	coca 3L R \$ 33 ✓
	Perfume 600 \$ 13 ✓
	<b>(855)</b>

Figura 2. Registro de ventas.

Respecto al control de los proveedores, cada uno de ellos le proporcionan un ticket dónde le muestra la cantidad y precio que se adquirieron. La manera de

controlar y archivar esos tickets es guardándolos en una libreta, en dónde los clasifica por la marca del producto.

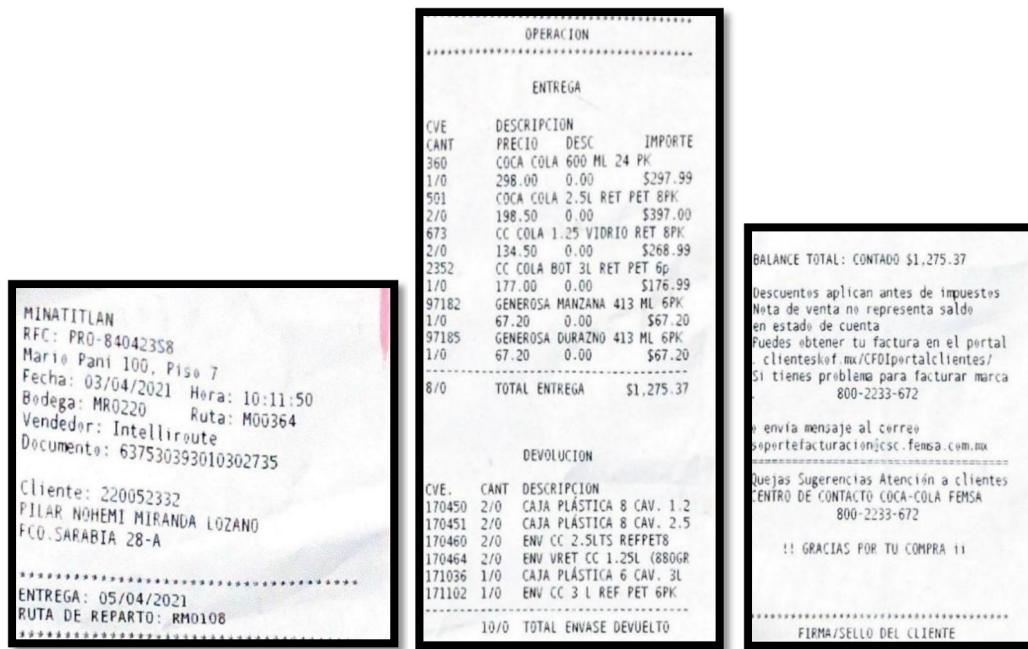


Figura 3. Tickets de pedido.

Al momento que se adquiere mercancía nueva, el administrador los ingresa a su hoja de inventario, sin embargo, como el cliente no cuenta con una buena administración, divide el almacén en dos partes, los productos exhibidos y los que cuenta en almacén.

ESQUEMA DE MAYOR									
<b>Bolzaza</b>									
Paketaxo									
Sabritas y Ruffles									
Duntos y ralentitos									
Fritos y crujitas									
Chotos y chumomais									
Yahult									
2 Jugo vidrio									
6 delaware ch.									
8 delaavare lata									
1 coca lata									
pintada									
power									
agua 1L									
87 agua 600									
6 Sabor 600 Mineral									
8 Sabor 2L									
21 Sidral 2L									
13 coca 1L									
coca 600									
13 coca 400									
23 coca 2.5									
17 coca 3L									
coca 3L desech									
6 coca vidrio 500									
coca 1.4									
ESQUEMA DE MAYOR									
6 4 4 15 13 18 11 9 12 11									
25 24 24 28 26 24 23 21 27 23									
12 11 11 24 24 23 20 19 21 21									
1 1 1 8 8 8 7 6 9 6									
14 12 11 32 30 28 26 26 34 32									
8 8 6 14 14 13 5 5 13 9									
8 8 7 6 6 6 8 8 8 8									
(3) 6 6 6 5 5 5 7 7 5									
(8) 4 3 3 5 5 5 5 5 6 6									
3 3 3 8 5 4 4 5 6 6									
— — 4 2 - 10 ✓ 10 13 10									
5 6 6 3 2 2 7 3 7 8 8									
21 11 5 5 11 4 11 4 3 5 4									
(10) 4 4 4 2 4 - 4 4 9 6 ✓ 6 5 5									
6 4 4 17 17 17 16 12 ✓ 11 15 13									
2 2 2 3 3 3 3 3 3 3 3									
21 3 3 2 - - - 2 ✓ 10 3 2									
5 5 5 5 5 5 5 5 5 2									
19 14 14 10 8 8 21 ✓ 20 18 14									
6 6 6 8 8 8 7 9 2 8 8 8 5 8									
18 8 20 18 8 8 9 15 8 2 7 8 3 7									
7 17 10 8 15 12 6 8 5 19 6 9 4									
2 2 2 3 2 1 1 3 3 3 3 3									
6 6 6 3 2 - 16 5 4 13 5 9 5									
(8) 5 11 11 3 8 8 15 5 2 2 5 3									

Figura 4. Registro de inventario.

En el negocio son 3 personas los que laboran ahí, el administrador también realiza actividades de empleado, sin embargo, es el único que puede agregar productos nuevos, modificar precios o eliminar productos, además de añadir productos nuevos para el negocio. También es el único que puede dar de baja del inventario mercancía dañada.

## **Requisitos funcionales y no funcionales**

### **Lista de requerimientos funcionales y no funcionales**

#### **REQUERIMIENTOS FUNCIONALES**

- Autenticación de usuario.
- Registrar usuarios.
- Consultar información.
- Menú de inicio.
- Gestionar inventario.
- Generar reportes.
- Registro de datos

#### **REQUERIMIENTOS NO FUNCIONALES**

- Interfaz del sistema.
- Soporte del sistema.
- Nivel de usuario.
- Confiabilidad del sistema.
- Seguridad de información.

## Matriz de trazabilidad

	MATRIZ DE TRAZABILIDAD DE REQUISITOS	PAG. 1 DE 1
--	--------------------------------------	-------------

TÍTULO DEL PROYECTO	Sistema de ventas para abarrotes (SIABA)	Fecha edición	14/09/2021
		Código Proyecto	#####

### ESTADO DE LOS REQUISITOS DEL PROYECTO

ID	REQUISITO	TIPO	PRIORIDAD	ESTADO	OBJETIVO	ENTREGABLE(s)	ESTADO (Entreg.)	VALIDACIÓN (persona/fecha)
RF01	Autenticación de usuario	RN	Alta	Activo	Restringir las acciones de unos usuarios a otros	Módulo con inicio de sesión		Pilar Nohemi Miranda Lozano
RF02	Registrar usuarios	RP	Alta	Activo	Almacenar información de los usuarios y crear su cuenta para acceder al sistema	Módulo donde se registrarán los usuarios		Pilar Nohemi Miranda Lozano
RF03	Consultar información	RN	Media	Activo	El usuario podrá ver información acerca de su negocio	Módulo donde se generarán las consultas		Pilar Nohemi Miranda Lozano
RF04	Menú de inicio	RP	Media	Activo	Tener un control en los productos de bajo stock	Un menú al iniciar sesión		Pilar Nohemi Miranda Lozano
RF05	Gestionar inventario	RN	Alta	Activo	Tener un control sobre las entradas y	Módulo para automatización		Pilar Nohemi Miranda Lozano

					salidas	de inventario		
RF06	Generar reportes	RN	Alta	Activo	Generar reportes de tablas específicas	Módulo donde se generarán reportes		Pilar Nohemi Miranda Lozano
RF07	Registro de datos	RN	Alta	Activo	Registrar información en tablas específicas	Módulo para el registro general de datos		Pilar Nohemi Miranda Lozano
RFN01	Interfaz del sistema	RP	Baja	Activo	Ser sencilla de usar y conectar con todos los usuarios	Interfaz de usuario		Pilar Nohemi Miranda Lozano
RFN02	Soporte del sistema	RC	Baja	Activo	Ofrecer ayuda a los futuros usuarios	Manual de usuario		Pilar Nohemi Miranda Lozano
RFN03	Nivel de usuario	RN	Alta	Activo	Acceder a la información de acuerdo al rol del usuario	Acceso a la información de acuerdo al usuario		Pilar Nohemi Miranda Lozano
RFN04	Confiabilidad del sistema	RC	Alta	Activo	Un sistema activo	Backup constante		Pilar Nohemi Miranda Lozano
RFN05	Seguridad de información	RC	Alta	Activo	Un sistema donde se almacenen de manera segura documentación, archivos, contraseñas, etc.	Sistema con inicio de sesión		Pilar Nohemi Miranda Lozano

RN	Requerimientos del negocio	RP	Requerimientos del proyecto	RI	Requerimientos de las partes interesadas
RI	Requerimientos de las partes interesadas	RC	Requerimientos de calidad		

# Documentación de requisitos

## Requisitos Funcionales

<b>Número del requisito:</b>	RF01
<b>Nombre del requisito:</b>	Autenticación de usuario
<b>Características:</b>	Los usuarios deberán identificarse con un usuario y contraseña para acceder al sistema.
<b>Descripción del requisito:</b>	<p>Usuario:</p> <p>Descripción paso a paso de como se autentificará en el sistema:</p> <ol style="list-style-type: none"><li>1. Deberá estar registrado en el sistema, en caso contrario deberá cubrir el RF02.</li><li>2. Si ya tiene una cuenta, deberá ingresar su usuario: RFC y contraseña.</li><li>3. Si los datos son correctos, entra al sistema mostrando el menú principal</li><li>4. Si los datos son incorrectos, tendrá un máximo de 3 intentos para acceder.</li><li>5. Si supera el numero máximo de intentos, tendrá que llamar al administrador para un cambio de contraseña.</li></ol>
<b>Prioridad del requisito:</b>	<b>Alta/Esencial</b> Media/Deseado      Baja/Opcional

<b>Número del requisito:</b>	RF02
<b>Nombre del requisito:</b>	Registrar usuarios
<b>Características:</b>	Los usuarios deberán registrarse en el sistema para tener una cuenta y con esto acceder al sistema.
<b>Descripción del requisito:</b>	El sistema permitirá al usuario (Administrador, empleado) registrarse.

	<ul style="list-style-type: none"> <li>• El usuario administrador debe ingresar datos como: RFC, Nombre, Apellido paterno, Apellido materno, E-mail, Contraseña.</li> <li>• El usuario empleado debe ingresar datos como: RFC, Nombre, Apellido paterno, Apellido materno, Numero telefónico, Calle, No. Calle, Colonia, Ciudad, Copia de INE (PDF), Solicitud de empleo(PDF), Comprobante de domicilio(PDF) y contraseña.</li> </ul>
<b>Prioridad del requisito:</b>	Alta/Esencial      Media/Deseado      Baja/Opcional

<b>Número del requisito:</b>	RF03
<b>Nombre del requisito:</b>	Consultar información
<b>Características:</b>	El sistema ofrecerá información general acerca de los productos, ventas, pedidos y proveedores.
<b>Descripción del requisito:</b>	<p>Consultar productos: Muestra información de todos los productos como su nombre, precio de compra, precio de venta, el stock y unidad de medida.</p> <p>Consultar ventas: permite al usuario ver las ventas realizadas y mostrar datos como fecha, monto total y el usuario que realizó la venta.</p> <p>Consultar pedidos: Muestra a los usuarios información de los pedidos realizados como fecha, monto total, quien realizó el pedido y quien es el proveedor.</p> <p>Consultar proveedores: Permite a los usuarios ver información acerca de los proveedores como el nombre del proveedor.</p> <p>Consulta prestamos: Permite al usuario visualizar los prestamos hechos por los usuarios, mostrando datos como el id del prestamo, concepto del préstamo, monto del préstamo, fecha y hora.</p>
<b>Prioridad del requisito:</b>	Alta/Esencial      Media/Deseado      Baja/Opcional

<b>Número del requisito:</b>	RF04		
<b>Nombre del requisito:</b>	Menú de inicio		
<b>Características:</b>	El sistema ofrecerá una pantalla de inicio donde se observarán los productos que tienen un bajo stock.		
<b>Descripción del requisito:</b>	Al ingresar al sistema cubriendo el RF01, el usuario visualizará una ventana donde se encontrarán los productos que tienen un bajo stock, mostrando el nombre del producto, el precio de compra y el stock.		
<b>Prioridad del requisito:</b>	Alta/Esencial	Media/Deseado	Baja/Opcional

<b>Número del requisito:</b>	RF05		
<b>Nombre del requisito:</b>	Gestionar inventario		
<b>Características:</b>	El sistema ofrecerá una automatización al vender algún producto o realizar un pedido.		
<b>Descripción del requisito:</b>	Al momento de realizar una venta el sistema automáticamente restara la cantidad vendida del producto en su stock.  Al momento de recibir un pedido se dan de alta los productos esto provocará que el stock aumente conforme a la cantidad del producto pedido.		
<b>Prioridad del requisito:</b>	Alta/Esencial	Media/Deseado	Baja/Opcional

<b>Número del requisito:</b>	RF06		
<b>Nombre del requisito:</b>	Generar Reportes		
<b>Características:</b>	El sistema permitirá la generación de reportes de diferentes tablas.		
<b>Descripción del requisito:</b>	<p>El usuario podrá generar reportes de:</p> <ul style="list-style-type: none"> <li>● Ventas</li> <li>● Pedidos</li> <li>● Prestamos</li> <li>● Inventarios</li> </ul>		
<b>Prioridad del requisito:</b>	Alta/Esencial	Media/Deseado	Baja/Opcional

<b>Número del requisito:</b>	RF07		
<b>Nombre del requisito:</b>	Registro de datos		
<b>Características:</b>	El sistema permitirá el registro de datos dentro de las tablas de la base de datos.		
<b>Descripción del requisito:</b>	<p>El usuario podrá registrar datos en las tablas de:</p> <ul style="list-style-type: none"> <li>● Unidades de medida</li> <li>● Proveedores</li> <li>● Roles</li> <li>● Ventas</li> <li>● Pedidos</li> <li>● Prestamos</li> <li>● Productos al inventario</li> </ul>		
<b>Prioridad del requisito:</b>	Alta/Esencial	Media/Deseado	Baja/Opcional

## Requerimientos no funcionales

<b>Número del requisito:</b>	RFN01		
<b>Nombre del requisito:</b>	Interfaz del sistema		
<b>Características:</b>	El sistema presentará una interfaz sencilla para que sea de fácil aprendizaje y manejo a los usuarios del sistema.		
<b>Descripción del requisito:</b>	El sistema tendrá una interfaz de uso intuitiva y sencilla.		
<b>Prioridad del requisito:</b>	Alta/Esencial	Media/Deseado	Baja/Opcional

<b>Número del requisito:</b>	RFN02		
<b>Nombre del requisito:</b>	Soporte del sistema		
<b>Características:</b>	El sistema tendrá un manual de usuario para facilitar el uso del sistema.		
<b>Descripción del requisito:</b>	Le permitirá al usuario disponer de documentación que le permita hacer uso del sistema.		
<b>Prioridad del requisito:</b>	Alta/Esencial	Media/Deseado	Baja/Opcional

<b>Número del requisito:</b>	RFN03		
<b>Nombre del requisito:</b>	Nivel de usuario		
<b>Características:</b>	Garantizar al usuario el acceso de información de acuerdo al nivel que posee.		
<b>Descripción del requisito:</b>	Facilidades y controles para permitir el acceso a la información a los usuarios autorizados con la intención de consultar y modificar la información de acuerdo al usuario.		

<b>Prioridad del requisito:</b>	<b>Alta/Esencial</b>	Media/Deseado	Baja/Opcional
---------------------------------	----------------------	---------------	---------------

<b>Número del requisito:</b>	RFN04
<b>Nombre del requisito:</b>	Confiabilidad del sistema
<b>Características:</b>	El sistema tendrá que estar en funcionamiento 12 horas los 7 días de la semana.
<b>Descripción del requisito:</b>	La disponibilidad del sistema debe ser continua durante las 12 horas garantizando contar con una contingencia por una posible falla. Por lo tanto, el sistema contará con un proceso de backup diario.
<b>Prioridad del requisito:</b>	<b>Alta/Esencial</b>
	Media/Deseado
	Baja/Opcional

<b>Número del requisito:</b>	RFN05
<b>Nombre del requisito:</b>	Seguridad de información
<b>Características:</b>	El sistema garantiza seguridad a los usuarios una seguridad en cuanto a la información que se procede en el sistema.
<b>Descripción del requisito:</b>	Garantizar la seguridad del sistema con respecto a la información almacenada como documentos, contraseñas y datos personales.
<b>Prioridad del requisito:</b>	<b>Alta/Esencial</b>
	Media/Deseado
	Baja/Opcional

## Diagrama del dominio (diagrama de clases)

### DC REPUBE

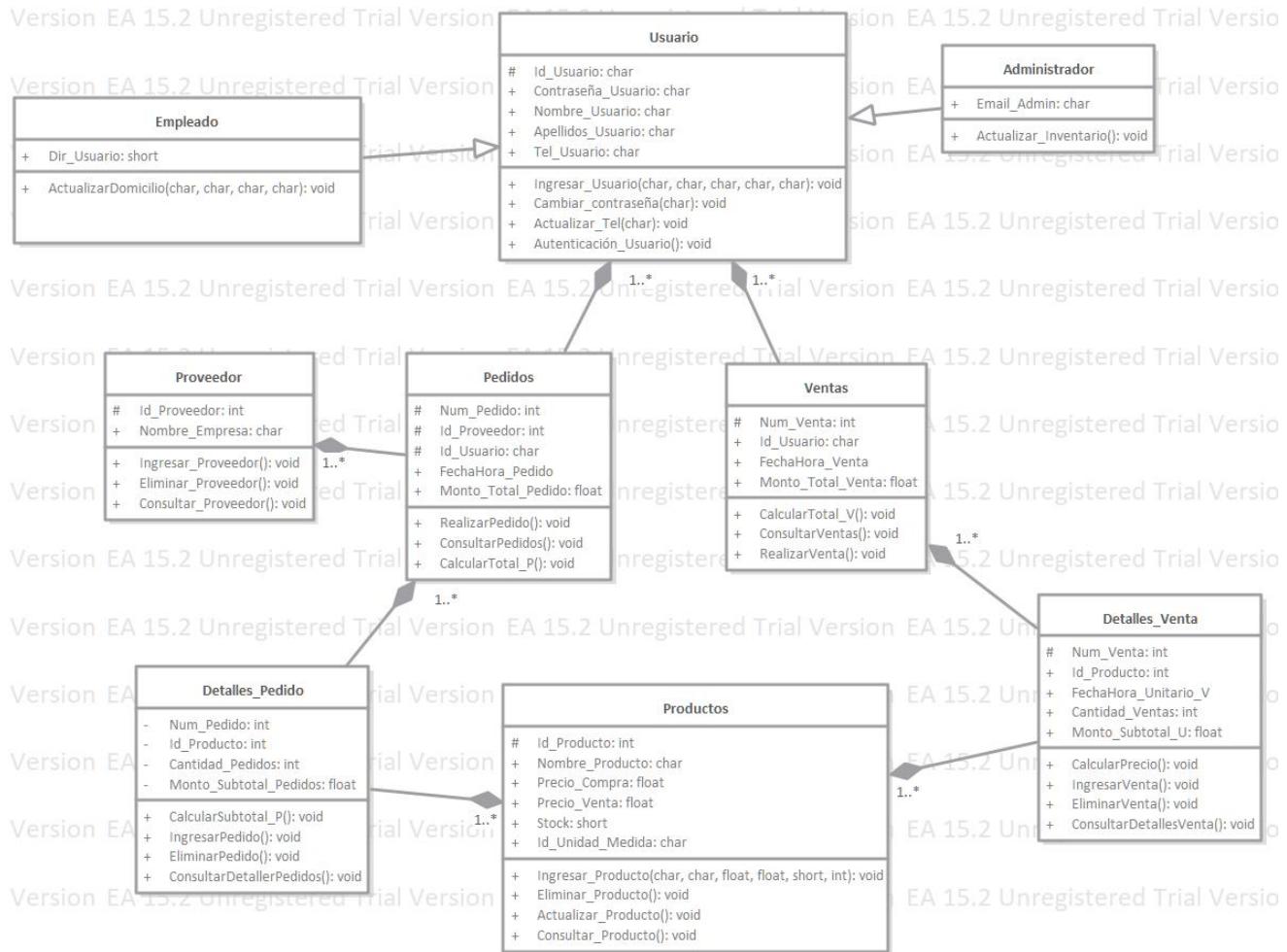


Figura 5. DC REPUBE.

## Administrador

Persona dueña del negocio, al ser el administrador tendrá privilegios sobre el empleado

RELACIÓN
<p>↳ Generalización de Administrador a Usuario</p> <p>[ Direction is 'Source -&gt; Destination'. ]</p>

## ATRIBUTOS

◆ Email\_Admin : char Public

[ Is static True. Containment is Not Specified. ]

## MÉTODOS

◆ Actualizar\_Inventario () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## Detalles\_Pedido

Representa el cuerpo de un pedido, con este vemos los detalles de los productos que se pidieron al proveedor

## RELACIÓN

◀ Agregación de Detalles\_Pedido a Pedidos

[ Direction is 'Source -> Destination'. ]

◀ Agregación de Detalles\_Pedido a Productos

[ Direction is 'Source -> Destination'. ]

## ATRIBUTOS

◆ Num\_Pedido : int Private

[ Is static True. Containment is Not Specified. ]

◆ Id\_Producto : int Private

[ Is static True. Containment is Not Specified. ]

◆ Cantidad\_Pedidos : int Private

[ Is static True. Containment is Not Specified. ]

◆ Monto\_Subtotal\_Pedidos : float Private

[ Is static True. Containment is Not Specified. ]

## MÉTODOS

## MÉTODOS

- ◆ CalcularSubtotal\_P () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ◆ IngresarPedido () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ◆ EliminarPedido () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ◆ ConsultarDetallerPedidos () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## Detalles\_Venta

Representa el cuerpo de una venta, con esta vemos los detalles de los productos que se vendieron en una venta

## RELACIÓN

- ◀ Agregación de Detalles\_Venta a Productos  
[ Direction is 'Source -> Destination'. ]
- ◀ Agregación de Detalles\_Venta a Ventas  
[ Direction is 'Source -> Destination'. ]

## ATRIBUTOS

- ◆ Num\_Venta : int Protected  
[ Is static True. Containment is Not Specified. ]
- ◆ Id\_Producto : int Public  
[ Is static True. Containment is Not Specified. ]
- ◆ FechaHora\_Unitario\_V : Public  
[ Is static True. Containment is Not Specified. ]
- ◆ Cantidad\_Ventas : int Public  
[ Is static True. Containment is Not Specified. ]

## ATRIBUTOS

◆ Monto\_Subtotal\_U : float Public

[ Is static True. Containment is Not Specified. ]

## MÉTODOS

◆ CalcularPrecio () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

◆ IngresarVenta () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

◆ EliminarVenta () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

◆ ConsultarDetallesVenta () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## Empleado

Persona que trabaja para el administrador, tendrá acceso a la información de la base de datos de acuerdo a su nivel de acceso.

## RELACIÓN

◀ Generalización de Empleado a Usuario

[ Direction is 'Source -> Destination'. ]

## ATRIBUTOS

◆ Dir\_Usuario : short Public

[ Is static True. Containment is Not Specified. ]

◆ Doc\_Usuario : Public

[ Is static True. Containment is Not Specified. ]

## ASOCIACIONES

✍ Association (direction: Unspecified)

Source: Public (Class) Empleado

Target: Public (Class) Prestamos

## MÉTODOS

◆ ActualizarDomicilio (CALLE\_USUARIO : char , NO\_CALLE\_USUARIO : char , COLONIA\_USUARIO : char , CIUDAD\_USUARIO : char ) : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## Pedidos

Representa la cabecera de los pedidos con esta identificamos un pedido

### RELACIÓN

◀ Agregación de Pedidos a Proveedor

[ Direction is 'Source -> Destination'. ]

◀ Agregación de Pedidos a Usuario

[ Direction is 'Source -> Destination'. ]

### RELACIONES

⇒ Agregación de Detalles\_Pedido a Pedidos

[ Direction is 'Source -> Destination'. ]

### ATRIBUTOS

◆ Num\_Pedido : int Protected

[ Is static True. Containment is Not Specified. ]

◆ Id\_Proveedor : int Protected

[ Is static True. Containment is Not Specified. ]

◆ Id\_Usuario : char Protected

[ Is static True. Containment is Not Specified. ]

◆ FechaHora\_Pedido : Public

## ATRIBUTOS

[ Is static True. Containment is Not Specified. ]

- ◆ Monto\_Total\_Pedido : float Public

[ Is static True. Containment is Not Specified. ]

## MÉTODOS

- ◊ RealizarPedido () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

- ◊ ConsultarPedidos () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

- ◊ CalcularTotal\_P () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## Productos

Representa los productos que hay en el negocio

## RELACIÓN

- ◀ Agregación de Productos a Unidades\_Medida

[ Direction is 'Source -> Destination'. ]

## RELACIONES

- ⇒ Agregación de Detalles\_Venta a Productos

[ Direction is 'Source -> Destination'. ]

- ⇒ Agregación de Detalles\_Pedido a Productos

[ Direction is 'Source -> Destination'. ]

## ATRIBUTOS

- ◆ Id\_Producto : int Protected

[ Is static True. Containment is Not Specified. ]

## ATRIBUTOS

- ◆ Nombre\_Producto : char Public  
[ Is static True. Containment is Not Specified. ]
- ◆ Precio\_Compra : float Public  
[ Is static True. Containment is Not Specified. ]
- ◆ Precio\_Venta : float Public  
[ Is static True. Containment is Not Specified. ]
- ◆ Stock : short Public  
[ Is static True. Containment is Not Specified. ]
- ◆ Id\_Unidad\_Medida : char Public  
[ Is static True. Containment is Not Specified. ]

## MÉTODOS

- ❖ Ingresar\_Producto (Id\_producto : char , Nombre\_Producto : char , Precio\_Compra : float , Precio\_Venta : float , Stock : short , Unidad\_Medida : int ) : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ❖ Eliminar\_Producto () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ❖ Actualizar\_Producto () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ❖ Consultar\_Producto () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## Proveedor

Empresa que surte los productos por medio de un pedido

## RELACIONES

- ⇒ Agregación de Pedidos a Proveedor  
[ Direction is 'Source -> Destination'. ]

## ATRIBUTOS

- ◆ Id\_Proveedor : int Protected  
[ Is static True. Containment is Not Specified. ]
- ◆ Nombre\_Empresa : char Public  
[ Is static True. Containment is Not Specified. ]

## MÉTODOS

- ◊ Ingresar\_Proveedor () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ◊ Eliminar\_Proveedor () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ◊ Consultar\_Proveedor () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## Usuario

Representa la persona que es usuario del sistema, está es una generalización de los dos posibles usuarios:  
Administrador y empleado

### RELACIONES

- ⇒ Agregación de Ventas a Usuario  
[ Direction is 'Source -> Destination'. ]
- ⇒ Generalización de Administrador a Usuario  
[ Direction is 'Source -> Destination'. ]
- ⇒ Generalización de Empleado a Usuario  
[ Direction is 'Source -> Destination'. ]
- ⇒ Agregación de Pedidos a Usuario  
[ Direction is 'Source -> Destination'. ]

## ATRIBUTOS

- ◆ Id\_Usuario : char Protected  
[ Is static True. Containment is Not Specified. ]
- ◆ Contraseña\_Usuario : char Public  
[ Is static True. Containment is Not Specified. ]
- ◆ Nombre\_Usuario : char Public  
[ Is static True. Containment is Not Specified. ]
- ◆ Apellidos\_Usuario : char Public  
[ Is static True. Containment is Not Specified. ]
- ◆ Tel\_Usuario : char Public  
[ Is static True. Containment is Not Specified. ]

## MÉTODOS

- ◆ Ingresar\_Usuario (NOMBRE\_USUARIO : char , APELLIDO\_P\_USUARIO : char , APELLIDO\_M\_USUARIO : char , ID\_USUARIO : char , TELEFONO\_USUARIO : char ) : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ◆ Cambiar\_contraseña (NuevaContraseña : char ) : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ◆ Actualizar\_Tel (NuevoTelefono : char ) : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]
- ◆ Autenticación\_Usuario () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

# Ventas

Representa la cabecera de las ventas con esta identificamos una venta

## RELACIÓN

⬅ Agregación de Ventas a Usuario

[ Direction is 'Source -> Destination'. ]

## RELACIONES

➡ Agregación de Detalles\_Venta a Ventas

[ Direction is 'Source -> Destination'. ]

## ATRIBUTOS

◆ Num\_Venta : int Protected

[ Is static True. Containment is Not Specified. ]

◆ Id\_Usuario : char Public

[ Is static True. Containment is Not Specified. ]

◆ FechaHora\_Venta : Public

[ Is static True. Containment is Not Specified. ]

◆ Monto\_Total\_Venta : float Public

[ Is static True. Containment is Not Specified. ]

## MÉTODOS

◆ CalcularTotal\_V () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

◆ ConsultarVentas () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

◆ RealizarVenta () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## Modelo del negocio (Diagramas de caso de uso)

### DCU Consultas

Diagrama de casos de uso que representa la visualización de información en forma de consultas y reportes

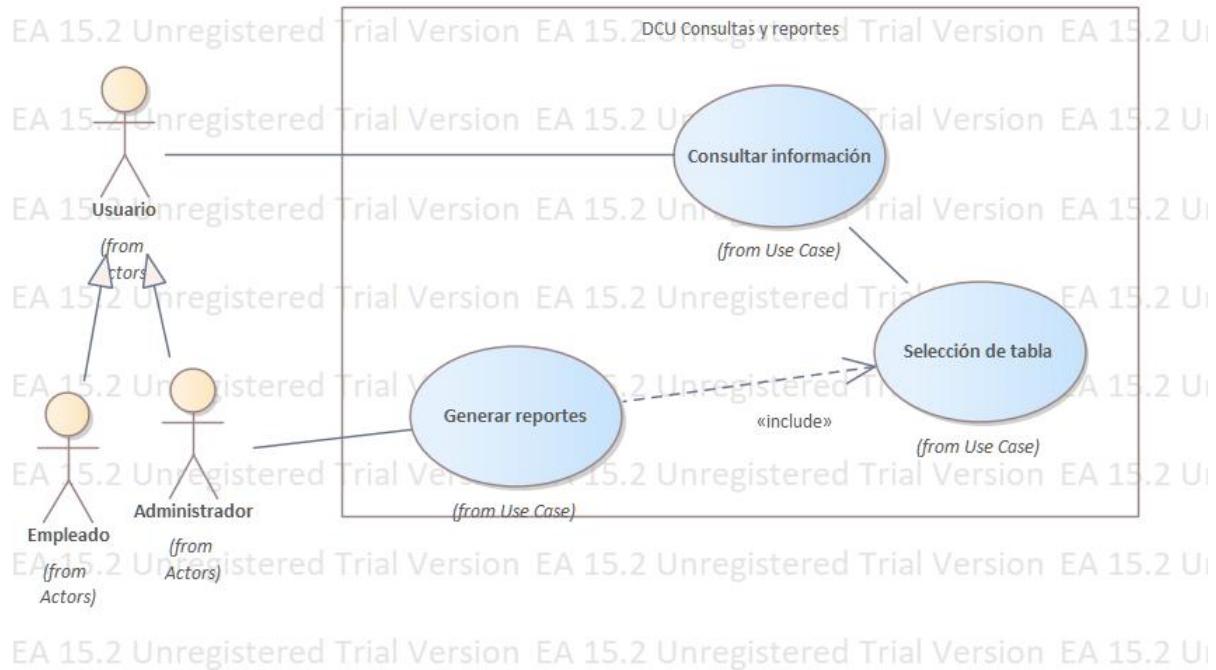


Figura 6. DCU Consultas

## Actores

### Administrador

Persona dueña del negocio

#### RELACIÓN

◀ Generalización de Administrador a Usuario

[ Direction is 'Source -> Destination'. ]

#### ASOCIACIONES

✓ Association (direction: Unspecified)

Source: Public (Actor) Administrador

Target: Public (UseCase) Generar reportes

### Empleado

Persona que trabaja para el administrador

#### RELACIÓN

◀ Generalización de Empleado a Usuario

[ Direction is 'Source -> Destination'. ]

### Usuario

Representa a las personas por igual, estas interactúan con el sistema, hay dos tipos de usuario: El administrador y empleado.

#### RELACIONES

⇒ Generalización de Administrador a Usuario

[ Direction is 'Source -> Destination'. ]

⇒ Generalización de Empleado a Usuario

[ Direction is 'Source -> Destination'. ]

## ASOCIACIONES

 Association (direction: Unspecified)

Source: Public (Actor) Usuario

Target: Public (UseCase) Nivel de usuario

## Casos de uso

### Consultar información

Representan las consultas que se harán por medio de las tablas de la base de datos

## CONECTORES

 Extend «extend» Source -> Destination

De: Nivel de usuario : UseCase, Public

a: Consultar información : UseCase, Public

## ASOCIACIONES

 Association (direction: Unspecified)

Source: Public (UseCase) Consultar información

Target: Public (UseCase) Selección de tabla

## Generar reportes

Representa los reportes que serán generados solo por el administrador.

## CONECTORES

 Include «include» Source -> Destination

De: Generar reportes : UseCase, Public

a: Selección de tabla : UseCase, Public

## ASOCIACIONES

 Association (direction: Unspecified)

Source: Public (Actor) Administrador

Target: Public (UseCase) Generar reportes

## Selección de tabla

Representa la tabla en la cual se hará una acción ya sea ingresar, consultar, borrar o actualizar.

CONECTORES
 <b>Include</b> «include»   Source -> Destination De:      Generar reportes : UseCase, Public a:      Selección de tabla : UseCase, Public
ASOCIACIONES
 Association (direction: Unspecified) Source: Public (UseCase) Consultar información                          Target: Public (UseCase) Selección de tabla

## DCU Gestionar inventario

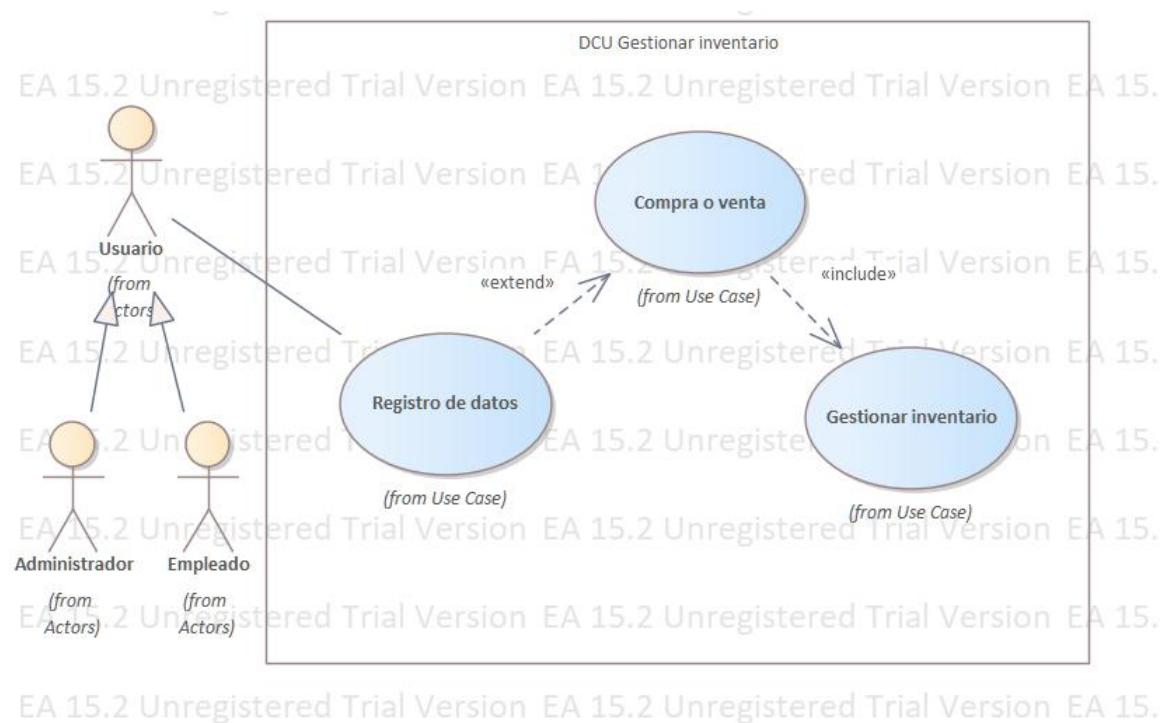


Figura 7. DCU Gestionar inventario

## Actores

### Administrador

Persona dueña del negocio

#### RELACIÓN

↳ Generalización de Administrador a Usuario

[ Direction is 'Source -> Destination'. ]

### Empleado

Persona que trabaja para el administrador

#### RELACIÓN

↳ Generalización de Empleado a Usuario

[ Direction is 'Source -> Destination'. ]

### Usuario

Representa a las personas por igual, estas interactúan con el sistema, hay dos tipos de usuario: El administrador y empleado.

#### RELACIONES

⇒ Generalización de Empleado a Usuario

[ Direction is 'Source -> Destination'. ]

⇒ Generalización de Administrador a Usuario

[ Direction is 'Source -> Destination'. ]

#### ASOCIACIONES

↗ Association (direction: Unspecified)

Source: Public (Actor) Usuario

Target: Public (UseCase) Nivel de usuario

## Casos de uso

## Compra o venta

Representa si el registro de datos es una compra o una venta para que logre el proceso de gestión de inventario

### CONECTORES

#### ↗ **Include** «include»    Source -> Destination

De:      Compra o venta : UseCase, Public  
a:        Gestionar inventario : UseCase, Public

#### ↗ **Extend** «extend»    Source -> Destination

De:      Registro de datos : UseCase, Public  
a:        Compra o venta : UseCase, Public

## Gestionar inventario

Representa la automatización del inventario, la cual es:

Al momento de realizar una venta el sistema automáticamente restara la cantidad vendida del producto en su stock.

Al momento de recibir un pedido se dan de alta los productos esto provocará que el stock aumente conforme a la cantidad del producto pedido

### CONECTORES

#### ↗ **Include** «include»    Source -> Destination

De:      Compra o venta : UseCase, Public  
a:        Gestionar inventario : UseCase, Public

## Registro de datos

Representan los datos que se registrarán en las tablas de la base de datos del sistema

CONECTORES
 <b>Extend</b> «extend»   Source -> Destination De:      Registro de datos : UseCase, Public a:      Compra o venta : UseCase, Public
 <b>Extend</b> «extend»   Source -> Destination De:      Registro de datos : UseCase, Public a:      Selección de tabla : UseCase, Public
 <b>Include</b> «include»   Source -> Destination De:      Nivel de usuario : UseCase, Public a:      Registro de datos : UseCase, Public

## Selección de tabla

Representa la tabla en la cual se hará una acción ya sea ingresar, consultar, borrar o actualizar.

CONECTORES
 <b>Extend</b> «extend»   Source -> Destination De:      Registro de datos : UseCase, Public a:      Selección de tabla : UseCase, Public

## DCU Inicio de sesión

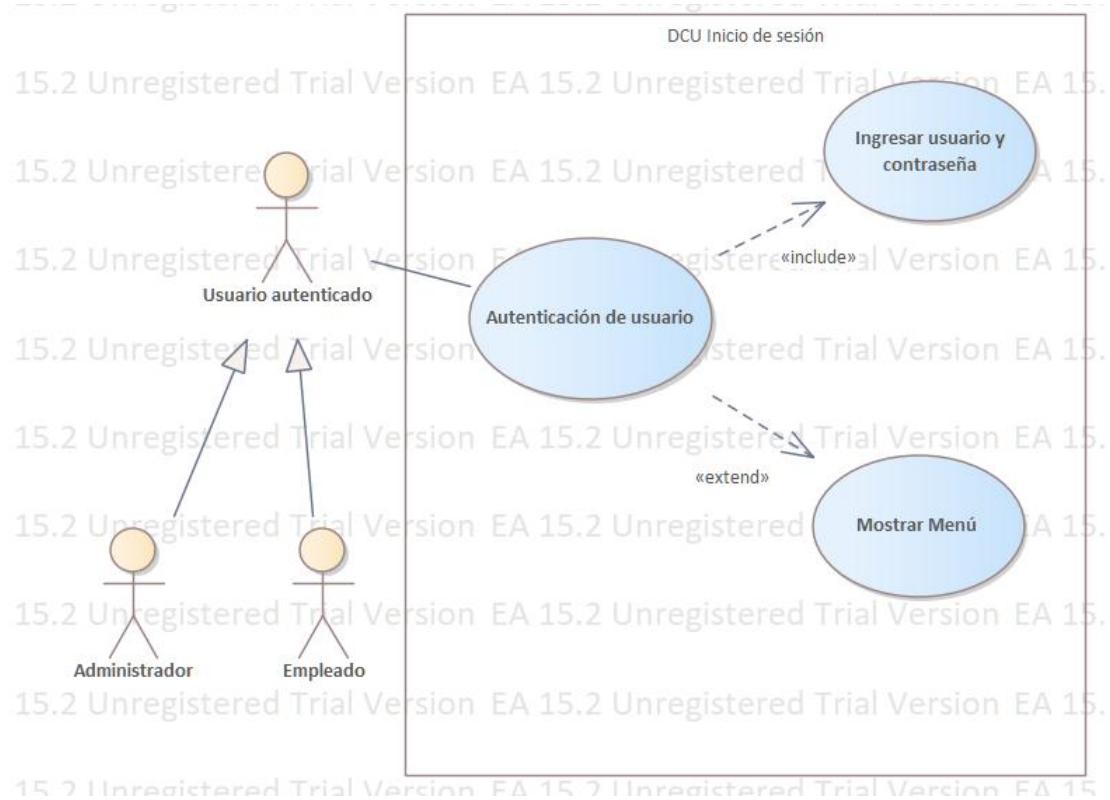


Figura 8. DCU Inicio de sesión

## Actores

### Administrador

Persona dueña del negocio

#### RELACIÓN

← Generalización de Administrador a Usuario autenticado

[ Direction is 'Source -> Destination'. ]

## Empleado

Persona que trabaja para el administrador

### RELACIÓN

↳ Generalización de Empleado a Usuario autenticado

[ Direction is 'Source -> Destination'. ]

## Usuario autenticado

Representa al usuario que registró correctamente sus credenciales al momento del inicio de sesión

### RELACIONES

⇒ Generalización de Empleado a Usuario autenticado

[ Direction is 'Source -> Destination'. ]

⇒ Generalización de Administrador a Usuario autenticado

[ Direction is 'Source -> Destination'. ]

### ASOCIACIONES

↙ Association (direction: Unspecified)

Source: Public (Actor) Usuario autenticado

Target: Public (UseCase) Autenticación de usuario

## Casos de usos

### Autenticación de usuario

Representa que las credenciales de los usuarios sean correctas

CONECTORES
<p>↗ <b>Include</b> «include»    Source -&gt; Destination De:        Autenticación de usuario : UseCase, Public a:        Ingresar usuario y contraseña : UseCase, Public</p>
<p>↗ <b>Extend</b> «extend»    Source -&gt; Destination De:        Autenticación de usuario : UseCase, Public a:        Mostrar Menú : UseCase, Public</p>

ASOCIACIONES
<p>↗ Association (direction: Unspecified) Source: Public (Actor) Usuario autenticado                      Target: Public (UseCase) Autenticación de usuario</p>

### Ingresar usuario y contraseña

El ser un usuario autenticado conlleva el haberse registrado antes, en ese momento especificamos un id de usuario y una contraseña, los cuales representa este caso de uso

CONECTORES
<p>↗ <b>Include</b> «include»    Source -&gt; Destination De:        Autenticación de usuario : UseCase, Public a:        Ingresar usuario y contraseña : UseCase, Public</p>

### Mostrar Menú

el usuario visualizará una ventana donde se encontrarán los productos que tienen un bajo stock, mostrando el nombre del producto, el precio de compra y el stock.

CONECTORES
<p>↗ <b>Extend</b> «extend»    Source -&gt; Destination De:        Autenticación de usuario : UseCase, Public a:        Mostrar Menú : UseCase, Public</p>

## DCU Registro

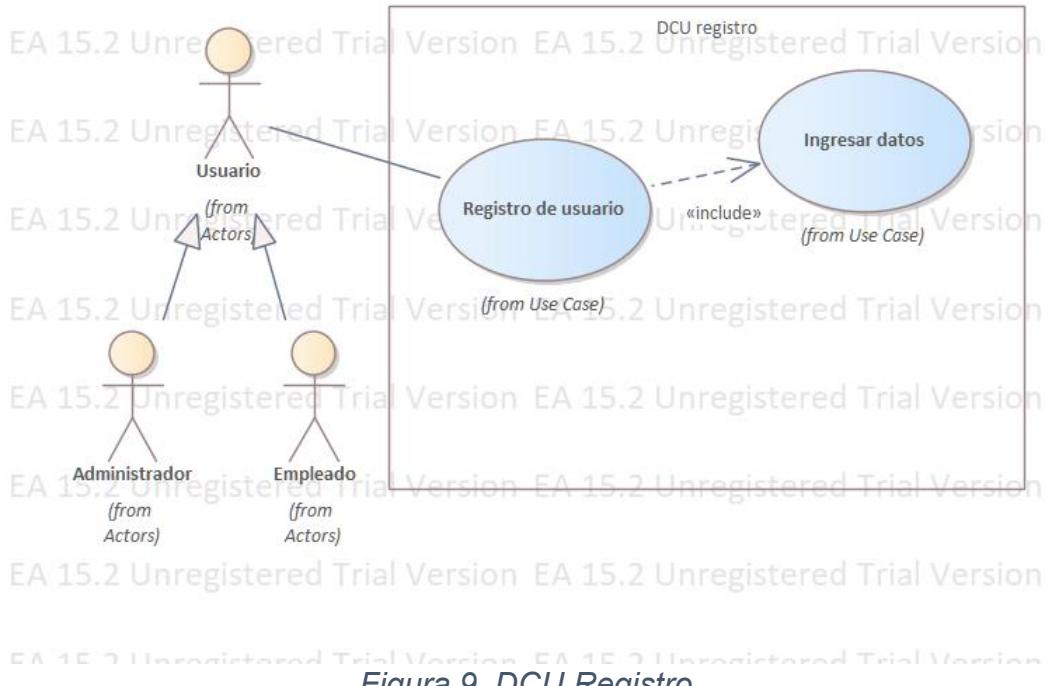


Figura 9. DCU Registro

## Actores

### Administrador

Persona dueña del negocio

#### RELACIÓN

← Generalización de Administrador a Usuario

[ Direction is 'Source -> Destination'. ]

### Empleado

Persona que trabaja para el administrador

#### RELACIÓN

← Generalización de Empleado a Usuario

[ Direction is 'Source -> Destination'. ]

## Usuario

Representa a las personas por igual, estas interactúan con el sistema, hay dos tipos de usuario: El administrador y empleado.

### RELACIONES

⇒ Generalización de Empleado a Usuario

[ Direction is 'Source -> Destination'. ]

⇒ Generalización de Administrador a Usuario

[ Direction is 'Source -> Destination'. ]

### ASOCIACIONES

✍ Association (direction: Unspecified)

Source: Public (Actor) Usuario

Target: Public (UseCase) Registro de usuario

## Casos de uso

### Ingresar datos

Al momento de registrarse el usuario deberá ingresar datos

- El usuario administrador debe ingresar datos como: RFC, Nombre, Apellido paterno, Apellido materno, E-mail, Contraseña.
- El usuario empleado debe ingresar datos como: RFC, Nombre, Apellido paterno, Apellido materno, Numero telefónico, Calle, No. Calle, Colonia, Ciudad, Copia de INE (PDF), Solicitud de empleo(PDF), Comprobante de domicilio(PDF) y contraseña.

### CONECTORES

✍ Include «include» Source -> Destination

De: Registro de usuario : UseCase, Public  
a: Ingresar datos : UseCase, Public

## Registro de usuario

Es el momento en el que el usuario se registra al sistema para poder ser un usuario autenticado.

### CONECTORES

- ↗ **Include** «include»    Source -> Destination  
De:      Registro de usuario : UseCase, Public  
a:        Ingresar datos : UseCase, Public

### ASOCIACIONES

- ↖ Association (direction: Unspecified)

Source: Public (Actor) Usuario

Target: Public (UseCase) Registro de usuario

## **Estudio de factibilidad**

### **Factibilidad económica**

#### **HARDWARE**

Teniendo en cuenta que la administradora posee el equipo necesario, los costos asociados a hardware son nulos.

Componente	Especificaciones mínimas	Costos
Memoria RAM	2 GB	\$399
Tarjeta de video	Estándar	\$1183
Procesador del equipo	Intel celeron	\$840
Disco duro	256 gb	\$650
Teclado	Estándar	\$100
Monitor	800x600	\$900
Impresora	Estándar	\$800
Lector código de barras	Estándar	\$462

#### **SOFTWARE**

Teniendo en cuenta que los equipos poseen el software necesario y apropiado sin la necesidad de recurrir a gastos extras o adquisición de alguna licencia, podemos afirmar que los costos asociados a software son nulos.

Componente	Requisitos del sistema
Sistema Operativo	Windows, MacOS o Linux.
Navegador Web	Google Chrome, Microsoft Edge, Mozilla Firefox.

## Factibilidad técnica

### HARDWARE

Componente	Especificaciones mínimas
Memoria RAM	2 GB
Tarjeta de video	Estándar
Procesador del equipo	Intel celeron
Disco duro	256 gb
Teclado	Estándar
Monitor	800x600
Impresora	Estándar
Lector código de barras	Estándar

## SOFTWARE

Componente	Requisitos del sistema
Sistema Operativo	Windows, MacOS o Linux.
Navegador Web	Google Chrome, Microsoft Edge, Mozilla Firefox.

Actualmente, la empresa cuenta con la tecnología tanto en el hardware como en el software necesaria, basándonos en la factibilidad técnica podemos decir que es factible por realizar el proyecto.

## Factibilidad operativa

Para la microempresa Pily, la necesidad de realizar un cambio en la situación actual es un hecho concreto, la manera en la que se esta gestionando la entrada y salida de sus productos provoca muchas dificultades en los tiempos que genera este proceso. Esto conlleva a que se generara la necesidad de automatizar el proceso de registro y salida de productos.

Teniendo esta información, expresada en reuniones y entrevistas con la dueña de la microempresa, se ha planteado una solución que consistirá en la creación de un sistema automatizado, el cual **SIABA** pretende ser un apoyo en gran parte de la administración que se produce actualmente.

Con el fin de desarrollar un software de calidad, se garantiza un buen funcionamiento del sistema y una excelente percepción por parte del usuario, donde estará presente una interfaz con un diseño minimalista y de uso intuitivo, que facilite la comprensión del sistema y que sea fácil de manejar, provocando que el usuario pueda familiarizarse con el sistema en muy poco tiempo.

El sistema será encargado de automatizar los procesos de entrada y salida del inventario, lo cual va a generar los siguientes aspectos:

- Se reduce la carga de trabajo, lo que permite desempeñar de mejor forma sus labores.
- Aumenta la confianza de los clientes con la empresa al trabajar con nuevas tecnologías de información (TI).
- Se disminuye el tiempo en el proceso de gestionar los productos.
- Permite revisar la base de datos para identificar que productos se encuentran disponibles.

Teniendo en cuenta los impactos positivos que genera el sistema, la microempresa Pily está de acuerdo con la realización de este, pues se encuentra beneficiado tras el desarrollo del proyecto.

Basándonos en la factibilidad operacional podemos decir que es factible realizar el proyecto.

## **Costo-beneficio del sistema**

Nuestro cliente ya cuenta con casi todo a nivel técnico, lo único con lo que no cuenta es con el lector de código de barras, por lo tanto, su inversión se recuperaría en menos de un año, al ser el sistema enfocado en software libre, no usamos licencias que aumenten el precio, con esto el precio final de nuestro sistema incluyendo el lector de código de barras sería de 10,000 pesos.

A tener en cuenta que nuestro cliente gasta aproximadamente 850 por mes en hojas, tinta, plumas, carpetas y corrector. Por lo tanto, el proyecto es factible.

## DISEÑO

### Diccionario de datos

#### Nombre de la tabla: PROVEEDOR

Campo	Tamaño	Tipo de dato	Descripción
ID_PROVEEDOR		INT	PRIMARY KEY
EMPRESA_PROVEEDOR	45	VARCHAR	NOT NULL
TELEFONO_PROVEEDOR	10	VARCHAR	

Relaciones:

Campos clave: ID\_PROVEEDOR

#### Nombre de la tabla: PRODUCTO

Campo	Tamaño	Tipo de dato	Descripción
ID_PRODUCTO	12	VARCHAR	PRIMARY KEY
NOMBRE_PRODUCTO	100	VARCHAR	NOT NULL
PRECIO_VENTA	7,2	FLOAT	NOT NULL
STOCK_MINIMO		INT	NOT NULL
EXISTENCIA		INT	NOT NULL
PESO		VARCHAR	
UNIDAD_MEDIDA		VARCHAR	

Relaciones:

Campos clave: ID\_PRODUCTO

#### Nombre de la tabla: USUARIO

Campo	Tamaño	Tipo de dato	Descripción
ID_USUARIO	13	VARCHAR	
CONTRASEÑA	50	VARCHAR	
NOMBRE_USUARIO	50	VARCHAR	
APELLIDO_P_USUARIO	50	VARCHAR	
APELLIDO_M_USUARIO	50	VARCHAR	

TEL_USUARIO	10	VARCHAR	
DIRECCION_USUARIO	200	VARCHAR	
EMAIL	70	VARCHAR	

**Relaciones:**

**Campos clave:** ID\_USUARIO

**Nombre de la tabla:** VENTAS

Campo	Tamaño	Tipo de dato	Descripción
NUM_VENTA	8	VARCHAR	PRIMARY KEY
CONSECUITIVO_V		INT	
ID_USUARIO	13	VARCHAR	
MONTO_TOTAL_VENTA	7,2	FLOAT	
FECHA_VENTA		DATE	
HORA_VENTA		TIME	

**Relaciones:** ID\_USUARIO(USUARIO)

**Campos clave:** NUM\_VENTA Y CONSECUITIVO\_V

**Nombre de la tabla:** DETALLES\_VENTA

Campo	Tamaño	Tipo de dato	Descripción
NUM_VENTA	8	VARCHAR	PRIMARY KEY
CONSECUITIVO_V		INT	
ID_PRODUCTO	12	VARCHAR	
CANTIDAD_VENTAS		INT	
MONTO_SUBTOTAL_V	7,2	FLOAT	
FECHA_V_U		DATE	
HORA_V_U		TIME	

**Relaciones:** NUM\_VENTA, CONSECUITIVO\_V (VENTAS)

ID\_PRODUCTO(PRODUCTO)

**Campos clave:** NUM\_VENTA, CONSECUITIVO\_V, ID\_PRODUCTO

### Nombre de la tabla: PEDIDOS

Campo	Tamaño	Tipo de dato	Descripción
NUM_PEDIDO	8	VARCHAR	
CONSECUITIVO_P		INT	
ID_PROVEEDOR		INT	
ID_USUARIO	13	VARCHAR	
MONTO_TOTAL_PEDIDOS	7,2	FLOAT	
FECHA_PEDIDOS		DATE	
HORA_PEDIDOS		TIME	

Relaciones: ID\_PROVEEDOR(PROVEEDOR)

ID\_USUARIO(USUARIO)

Campos clave: NUM\_PEDIDO, CONSECUITIVO\_P

### Nombre de la tabla: DETALLES\_PEDIDOS

Campo	Tamaño	Tipo de dato	Descripción
NUM_PEDIDO	8	VARCHAR	
CONSECUITIVO_P		INT	
ID_PRODUCTO	12	VARCHAR	
PRECIO_COMPRA	7,2	FLOAT	
CANTIDAD_PEDIDOS		INT	
MONTO_SUBTOTAL_P	7,2	FLOAT	
FECHA_P_U		DATE	
HORA_P_U		TIME	

Relaciones: NUM\_PEDIDO, CONSECUITIVO\_P(PEDIDOS)

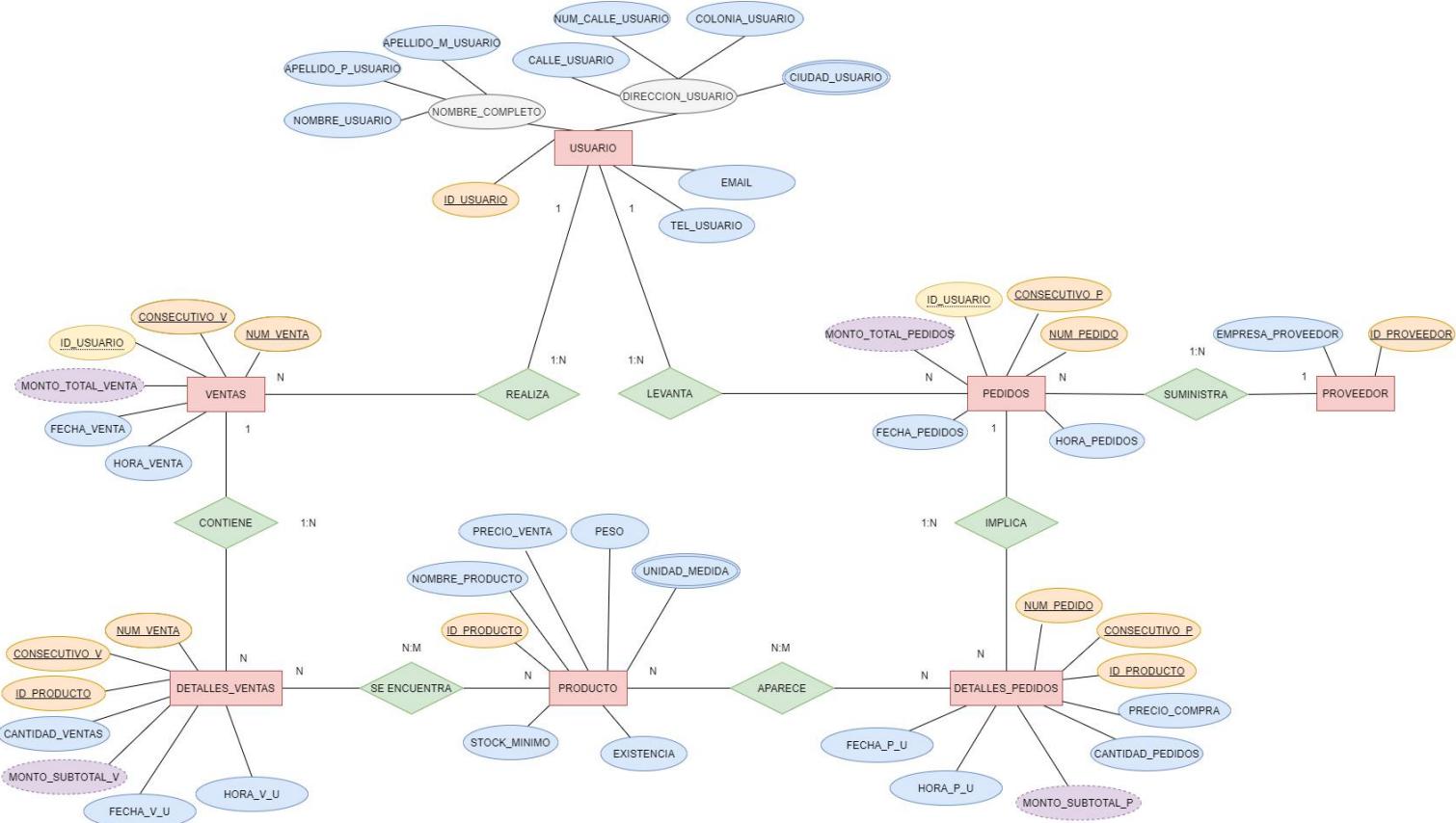
ID\_PRODUCTO(PRODUCTO)

Campos clave: NUM\_PEDIDO, CONSECUITIVO\_P, ID\_PRODUCTO

## Modelo E-R-E (normalizado)

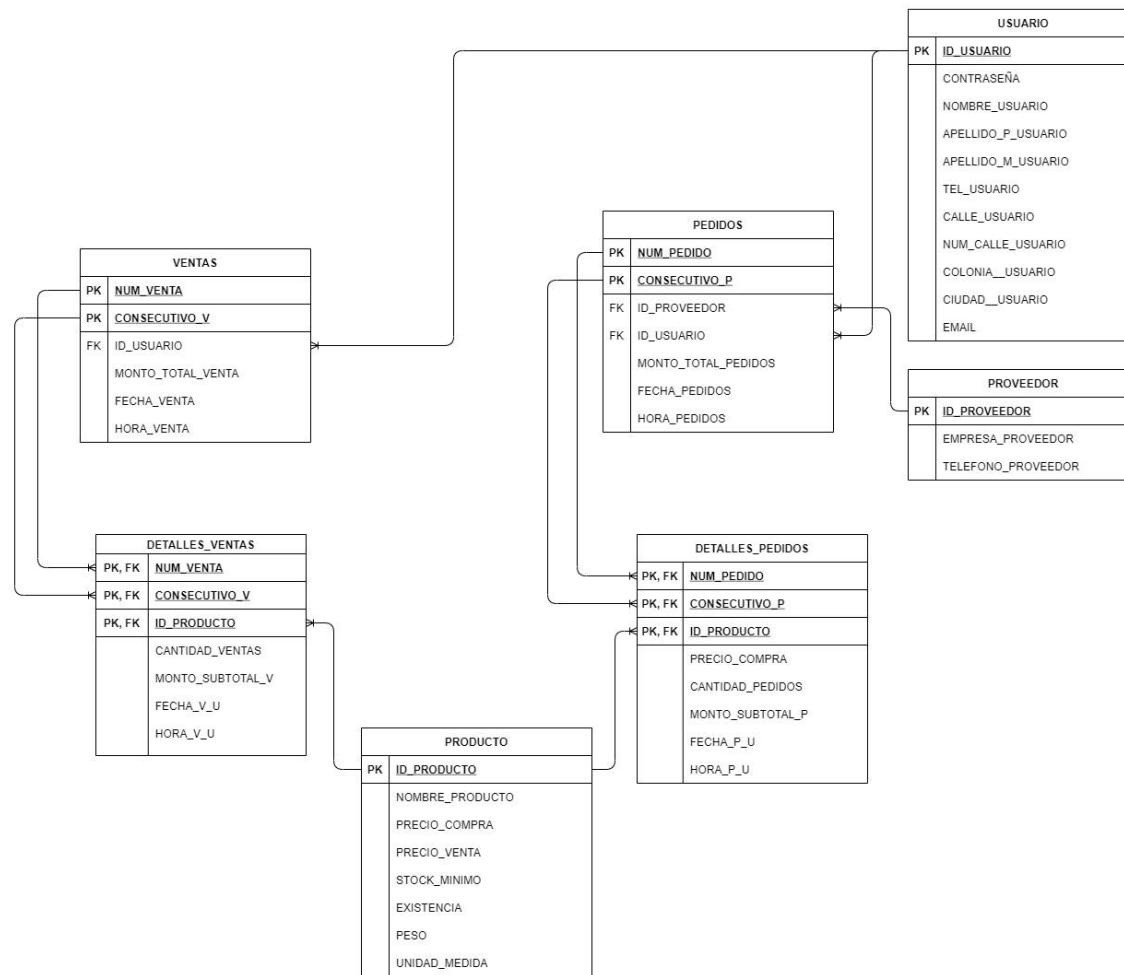
Como se puede observar, el siguiente diagrama es el modelo E-R-E del sistema SIABA, donde se identifican entidades y los diferentes tipos de atributos distinguidos por colores, ademas de las relaciones existentes entre entidades.

Rojo: Entidades, Azul: Atributos simples, Naranja: Atributos clave, Amarillo: Atributos foreign key, Morado: Calculados, Gris: Atributos compuestos, Azul y doble circulo: Atributos multivalorados, Verde: Relaciones.



## Modelo relacional

El siguiente diagrama es el modelo relacional del sistema SIABA, donde se puede observar de una mejor manera las relaciones entre cada tabla, especificando cual es llave primaria y foránea, ademas de tener la notación de pata de gallo la cual nos indica la cardinalidad.



## Justificación

El sistema manejador de base de datos que se ocupará en este proyecto es MariaDB ya que es de software libre y no requiere de alguna compra de licencia, ademas por su alto potencial es perfecto para el modelo del negocio de nuestro cliente.

El lenguaje de programación que se utilizará es HTML, CSS, JavaScript y PHP. Son los establecidos al momento, en algún caso de complicarse se utilizará Visual Basic.

Los lenguajes de programación web es por que se tiene diseñado una interfaz web donde solo tenga acceso el equipo designado, por lo tanto solo sera una aplicación local.

## Script de base de datos

Tenemos las siguientes capturas donde se puede visualizar el script de base de datos del sistema SIABA, como se menciono anteriormente el sistema manejador de bases de datos a utilizar es MariaDB, el script se secciono en tablas padre y tablas hijo.

### Tablas Padre

```
CREATE TABLE PROVEEDOR(
    ID_PROVEEDOR INT NOT NULL AUTO_INCREMENT,
    EMPRESA_PROVEEDOR VARCHAR(45) NOT NULL,
    TELEFONO_PROVEEDOR VARCHAR(10) NOT NULL,
    CONSTRAINT PK_ID_PROVEEDOR PRIMARY KEY (ID_PROVEEDOR)
);
```

Figura 10. Tabla PROVEEDOR.

```
CREATE TABLE PRODUCTO(
    ID_PRODUCTO VARCHAR(12) NOT NULL,
    NOMBRE_PRODUCTO VARCHAR(45) NOT NULL,
    PRECIO_VENTA FLOAT(7,2) NOT NULL,
    STOCK_MINIMO INT NOT NULL,
    EXISTENCIA INT NOT NULL,
    UNIDAD_MEDIDA INT NOT NULL,
    CONSTRAINT PK_ID_PRODUCTO PRIMARY KEY (ID_PRODUCTO)
);
```

Figura 11. Tabla PRODUCTO.

```
CREATE TABLE USUARIO(
    ID_USUARIO VARCHAR(13) NOT NULL,
    CONTRASEÑA VARCHAR(50) NOT NULL,
    NOMBRE_USUARIO VARCHAR(50) NOT NULL,
    APELLIDO_P_USUARIO VARCHAR(50) NOT NULL,
    APELLIDO_M_USUARIO VARCHAR(50) NOT NULL,
    TEL_USUARIO VARCHAR(10) NOT NULL,
    CALLE_USUARIO VARCHAR(50) NOT NULL,
    NUM_CALLE_USUARIO VARCHAR(10) NOT NULL,
    COLONIA_USUARIO VARCHAR(50) NOT NULL,
    CIUDAD_USUARIO VARCHAR(50) NOT NULL,
    EMAIL VARCHAR(50) NOT NULL,
    CONSTRAINT PK_ID_USUARIO PRIMARY KEY (ID_USUARIO)
);
```

Figura 12. Tabla USUARIO.

## Tablas Hijo

```
CREATE TABLE VENTAS(
    NUM_VENTA VARCHAR(8) DEFAULT DATE_FORMAT(CURDATE(), '%X%m%d'),
    CONSECUITIVO_V INT NOT NULL AUTO_INCREMENT,
    ID_USUARIO VARCHAR(13) NOT NULL,
    MONTO_TOTAL_VENTA FLOAT(7,2) NOT NULL,
    FECHA_VENTA DATE DEFAULT CURDATE(),
    HORA_VENTA TIME DEFAULT CURTIME(),
    CONSTRAINT PK_NUMVENTA_CONSECUITIVO PRIMARY KEY (NUM_VENTA, CONSECUITIVO_V),
    CONSTRAINT FK_ID_USUARIO FOREIGN KEY (ID_USUARIO) REFERENCES USUARIO(ID_USUARIO)
);
```

Figura 13. Tabla VENTAS.

```
CREATE TABLE DETALLES_VENTAS(
    NUM_VENTA VARCHAR(8) NOT NULL,
    CONSECUITIVO_V INT NOT NULL,
    ID_PRODUCTO VARCHAR(12) NOT NULL,
    CANTIDAD_VENTAS INT NOT NULL,
    MONTO_SUBTOTAL_V FLOAT NOT NULL,
    FECHA_V_U DATE DEFAULT CURDATE(),
    HORA_V_U TIME DEFAULT CURTIME(),
    CONSTRAINT PK_DETALLES_VENTAS PRIMARY KEY (NUM_VENTA, CONSECUITIVO_V, ID_PRODUCTO),
    CONSTRAINT FK_DETALLES_VENTAS_ID_PRODUCTO FOREIGN KEY (ID_PRODUCTO) REFERENCES PRODUCTO(ID_PRODUCTO)
);
```

Figura 14. Tabla DETALLES\_VENTA.

```
CREATE TABLE PEDIDOS(
    NUM_PEDIDO VARCHAR(8) DEFAULT DATE_FORMAT(CURDATE(), '%X%d%m'),
    CONSECUITIVO_P INT NOT NULL AUTO_INCREMENT,
    ID_PROVEEDOR INT NOT NULL,
    ID_USUARIO VARCHAR(13) NOT NULL,
    MONTO_TOTAL_PEDIDOS FLOAT NOT NULL,
    FECHA_PEDIDOS DATE DEFAULT CURDATE(),
    HORA_PEDIDOS TIME DEFAULT CURTIME(),
    CONSTRAINT PK_NUMPEDIDO_CONSECUITIVO PRIMARY KEY (NUM_PEDIDO, CONSECUITIVO_P),
    CONSTRAINT FK_PEDIDOS_ID_PROVEEDOR FOREIGN KEY (ID_PROVEEDOR) REFERENCES PROVEEDOR(ID_PROVEEDOR),
    CONSTRAINT FK_PEDIDOS_ID_USUARIO FOREIGN KEY (ID_USUARIO) REFERENCES USUARIO(ID_USUARIO)
);
```

Figura 15. Tabla PEDIDOS.

```

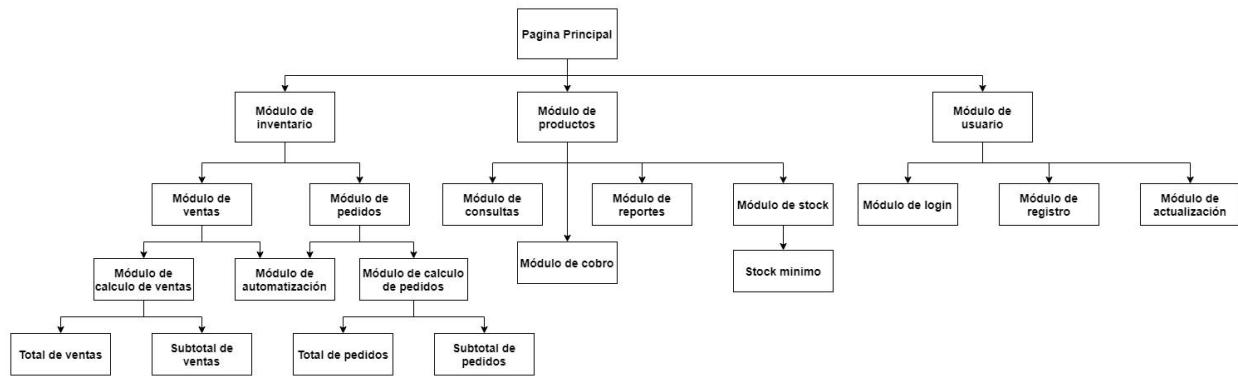
CREATE TABLE DETALLES_PEDIDOS(
    NUM_PEDIDO VARCHAR(8) NOT NULL,
    CONSECUATIVO_P INT NOT NULL,
    ID_PRODUCTO VARCHAR(12) NOT NULL,
    PRECIO_COMPRA FLOAT(7,2) NOT NULL,
    CANTIDAD_PEDIDOS INT NOT NULL,
    MONTO_SUBTOTAL_P FLOAT NOT NULL,
    FECHA_P_U DATE DEFAULT CURDATE(),
    HORA_P_U TIME DEFAULT CURTIME(),
    CONSTRAINT PK_DETALLES_PEDIDOS_NUMPEDIDO PRIMARY KEY(NUM_PEDIDO, CONSECUATIVO_P, ID_PRODUCTO),
    CONSTRAINT FK_DETALLES_IDPRODUCTO FOREIGN KEY (ID_PRODUCTO) REFERENCES PRODUCTO(ID_PRODUCTO)
);

```

Figura 16. Tabla DETALLES\_PEDIDOS.

## Diseño arquitectónico

El siguiente diseño arquitectónico es enfocado en los módulos relacionados a bases de datos y programación, hay que agregar los módulos gráficos como de interfaces.



## Diseño de interfaz de usuario

Las siguientes imágenes representan un prototipo funcional de interfaz de usuario del sistema SIABA.

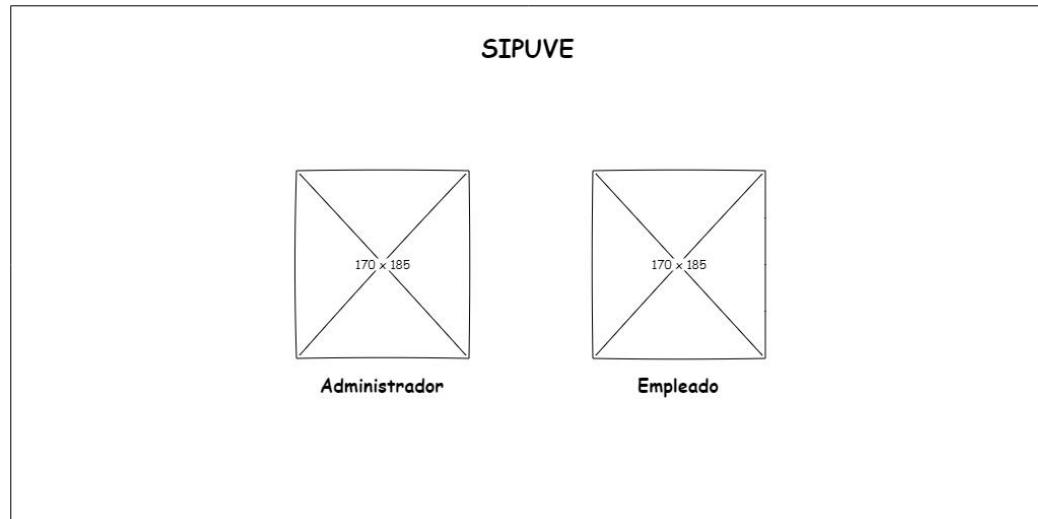


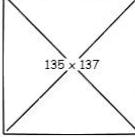
Figura 17. Selección de rol.

Este es un prototipo funcional de la interfaz de usuario para el login del administrador. La pantalla tiene un encabezado "SIPUVE" y un mensaje de bienvenida: "Bienvenido administrador, porfavor ingresa tus credenciales correctamente.". Abajo de este mensaje se muestra un cuadro cuadrado con una 'X' diagonal, con las dimensiones "135 x 137". A continuación, se presentan dos campos de texto: "Usuario:" seguido de un cuadro vacío, y "Contraseña:" seguido de otro cuadro vacío. Debajo de los campos hay un enlace "[¿Ha olvidado la contraseña?](#)" y un botón "Ingresar".

Figura 18. Login administrador.

**SIPUVE**

Bienvenido empleado, porfavor ingresa tus credenciales correctamente.



135 x 137

Usuario:

Contraseña:

[Ha olvidado la contraseña?](#)

*Figura 19. Login empleado.*

**SIPUVE**

*Figura 20. Inicio administrador.*

**Usuarios**

---

No	ID	NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO	TELEFONO	CALLE	# CALLE	COLONIA	CIUDAD	EMAIL	ACCIONES
1	POML000901QDA	LUIS EDUARDO	POZOS	MIRANDA	9212575914	FRANCISCO SARABIA	28	LA BOMBA	MINATITLAN	LALOPOZOS12@GMAIL.COM	<a href="#">Editor</a> <a href="#">Eliminar</a>
2	MILP0000000ABC	LUIS EDUARDO	POZOS	MIRANDA	9212575914	FRANCISCO SARABIA	28	LA BOMBA	MINATITLAN	LALOPOZOS12@GMAIL.COM	<a href="#">Editor</a> <a href="#">Eliminar</a>
3	ZXCV0000000C8A	LUIS EDUARDO	POZOS	MIRANDA	9212575914	FRANCISCO SARABIA	28	LA BOMBA	MINATITLAN	LALOPOZOS12@GMAIL.COM	<a href="#">Editor</a> <a href="#">Eliminar</a>

*Figura 21. Usuarios.*

**Nuevo usuario**

---

ID :	<input type="text"/>
CONTRASEÑA :	<input type="text"/>
NOMBRE :	<input type="text"/>
APELLIDO PATERNO:	<input type="text"/>
APELLIDO MATERNO:	<input type="text"/>
TELEFONO :	<input type="text"/> 9221234567
CALLE :	<input type="text"/>
No CALLE:	<input type="text"/>
COLONIA:	<input type="text"/>
CIUDAD:	<input type="text"/> MINATITLAN 
EMAIL:	<input type="text"/> Ejemplo@hotmail.com

**Agregar usuario**

*Figura 22. Agregar usuario.*

[Atras](#)

### Proveedores

[Nuevo](#)

ID	NOMBRE	TELEFONO	ACCIONES
1	Marinela	9226784598	Editar Eliminar
2	Barcel	9226798273	Editar Eliminar
3	Ricolino	9226096312	Editar Eliminar

Figura 23. Proveedores.

[Atras](#)

### Nuevo proveedor

ID :

NOMBRE :

TELEFONO:  9221234567

[Agregar proveedor](#)

Figura 24. Agregar proveedor.

[Atras](#)

### Productos

[Nuevo](#)

ID	NOMBRE	PRECIO COMPRA	PRECIO VENTA	STOCK MINIMO	EXISTENCIA	PESO	UNIDAD MEDIDA	ACCIONES
876096542341	COCA 3 L	32	33	10	24	3	LITROS	<a href="#">Editar</a> <a href="#">Eliminar</a>
876096542742	COCA 2.5 L	28	29	10	32	2.5	LITROS	<a href="#">Editar</a> <a href="#">Eliminar</a>

Figura 25. Productos.

[Atras](#)

### Nuevo producto

ID :

DESCRIPCION :

PRECIO COMPRA :

PRECIO VENTA :

STOCK MINIMO :

EXISTENCIA :

PESO :

UNIDAD MEDIDA :  PIEZA ▾

[Agregar producto](#)

Figura 26. Agregar producto.

**Pedidos**

NUM_PEDIDO	PROVEEDOR	USUARIO	TOTAL	FECHA	HORA	ACCIONES
202110281	2	POML000901QDA	1,664	28/10/2021	1:19	<a href="#">Editar</a> <a href="#">Eliminar</a>

*Figura 27. Pedidos.*

**Detalles pedidos**

ID:  [Agregar Producto](#)

NUM_PEDIDO	ID_PRODUCTO	PRECIO COMPRA	CANTIDAD	SUBTOTAL	FECHA	HORA	ACCIONES
202110281	876096542341	28	32	896	28/10/2021	1:20	<a href="#">Editar</a> <a href="#">Eliminar</a>
202110281	876096542742	32	24	768	28/10/2021	1:20	<a href="#">Editar</a> <a href="#">Eliminar</a>

*Figura 28. Detalles pedidos.*

Cantidad del producto

"PRODUCTO"

Precio compra:

Cantidad:

Total:

Aceptar Cancelar

Figura 29. Dar de alta producto en pedidos.

Atras

Ventas

NUM_VENTA	USUARIO	TOTAL	FECHA	HORA	ACCIONES
281020211	POML000901QDA	91	28/10/2021	1:25	Editar Eliminar

Figura 30. Ventas.

**Atras**

### Detalles ventas

ID:  **Agregar Producto**

NUM_VENTA	ID_PRODUCTO	CANTIDAD	SUBTOTAL	FECHA	HORA	ACCIONES
281020211	876096542341	1	33	28/10/2021	1:25	<b>Editar Eliminar</b>
281020211	876096542742	2	58	28/10/2021	1:26	<b>Editar Eliminar</b>

**Cobrar** Total:

Figura 31. Detalles ventas.

**Cobrar venta**

Total:

Efectivo:

Cambio:

**Aceptar** **Cancelar**

Figura 32. Cobrar una venta.

**Cantidad del producto**

"PRODUCTO"

Cantidad:

Importe:

**Aceptar** **Cancelar**

Figura 33. Dar de alta un producto en venta.

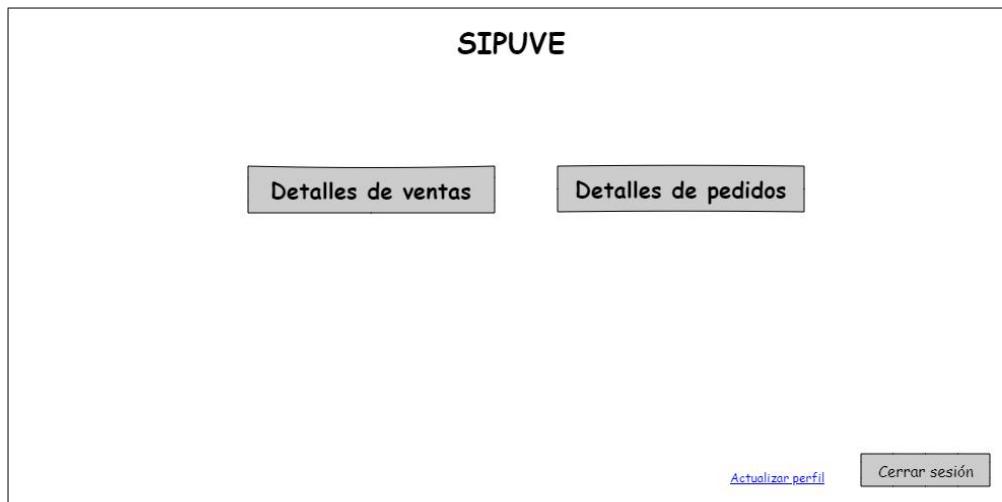


Figura 34. Inicio Empleado.

Detalles ventas											
Atras		Agregar Producto									
ID:											
<hr/>											
NUM_VENTA	ID_PRODUCTO	PRECIO COMPRA	CANTIDAD	SUBTOTAL	FECHA	HORA					
281020211	876096542341	32	1	33	28/10/2021	1:25					
281020211	876096542742	28	2	58	28/10/2021	1:26					
<hr/>											
				<input type="button" value="Cobrar"/>	Total:	<input type="text"/>					

Figura 35. Detalles ventas.

Cobrar venta

Total:

Efectivo:

Cambio:

Figura 36. Cobrar venta.

Cantidad del producto

"PRODUCTO"

Cantidad:

Importe:

Figura 37. Dar de alta un producto en una venta.

Atras

Detalles pedidos

ID:  Agregar Producto

NUM_PEDIDO	ID_PRODUCTO	CANTIDAD	SUBTOTAL	FECHA	HORA
202110281	876096542341	32	896	28/10/2021	1:20
202110281	876096542742	24	768	28/10/2021	1:20

Total:

Figura 38. Detalles pedidos.



*Figura 39. Dar de alta un producto en pedidos.*

## DESARROLLO

### Códigos de secciones principales

#### Login

```
login.php x
PHP projects > SIPUVE > login.php > ...
1  <?php
2  session_start();
3  $usuario=$_POST['usuario'];
4  $contraseña=$_POST['contraseña'];
5  $_SESSION['usuario'] = $usuario;
6
7  include("conexion.php");
8  $consulta="SELECT * FROM USUARIO WHERE ID_USUARIO='$usuario' AND PASSWORD='$contraseña'";
9  $resultado=mysqli_query($conexion, $consulta);
10
11 $filas = mysqli_fetch_array($resultado);
12
13 if ($filas['CARGO'] == 'ADMINISTRADOR') {
14     header("location:home.php");
15 }else
16 if ($filas['CARGO'] == 'EMPLEADO') {
17     header("location:home2.php");
18 }
19
20
21 else{
22 }
23 <?php
24     include("index.php");
25   ?>
26   <h1>ERROR... CONTRASEÑA O USUARIO INCORRECTO</h1>
27   <?php
28 }
29 mysqli_free_result($resultado);
30 mysqli_close($conexion);
```

El login consta de un método llamado `session_start()` el cual nos permite establecer la sesión del usuario actual, con ayuda de `$_POST` obtenemos el valor de la caja de texto con el nombre entre comas simples, por lo tanto la variable `$usuario` es igual al valor que se ingreso en la caja de texto llamada `usuario`.

La variable `$_SESSION` es un tipo de variable el cual se puede pasar entre varias paginas para esto se debe iniciar con el método antes mencionado `session_start()`.

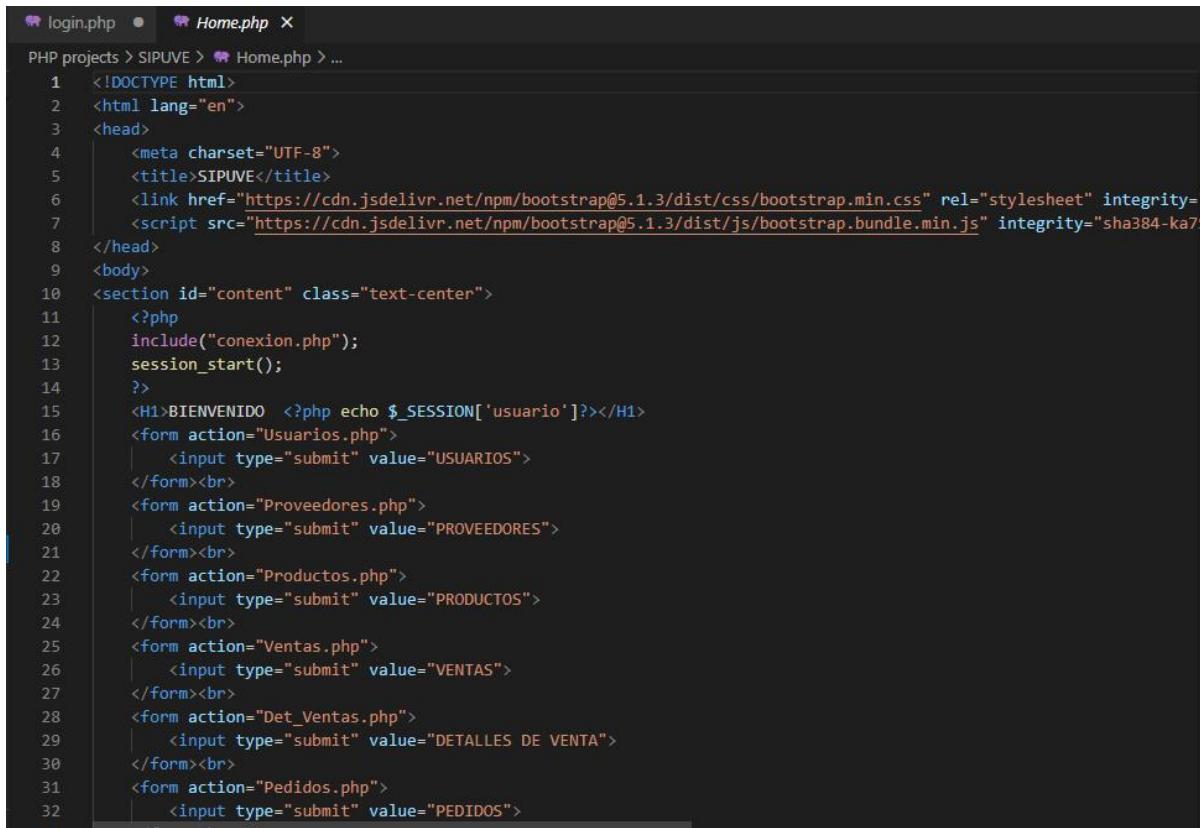
Una parte a tomar en cuenta es el `include(conexion.php)` lo cual hace como si fuera un import en java, el archivo `conexion.php` contiene lo necesario para conectarse a la base de datos.

La consulta se almacena en la variable \$consulta y para que se ejecute esta consulta debemos usar el método mysqli\_query() el cual recibe dos parámetros, la variable de conexión establecida en conexion.php y la variable donde se aloja la consulta.

Con el método mysqli\_fetch\_array() el cual recibe de parámetro la variable que aloja el método anterior, con esto lo que hará es que podemos asignar este parámetro a una variable y podemos obtener los resultados de esa consulta es por eso que ahí ocupamos un if para saber si el cargo del usuario que está ingresando es de administrador o empleado y a cada uno le pertenece su página de inicio.

Los usuarios que está validando son los usuarios de la base de datos, por lo tanto si se intenta ingresar al sistema y la contraseña es incorrecta (observe la línea de código 26) reproducirá el mensaje ERROR... CONTRASEÑA O USUARIO INCORRECTO.

## Página de inicio



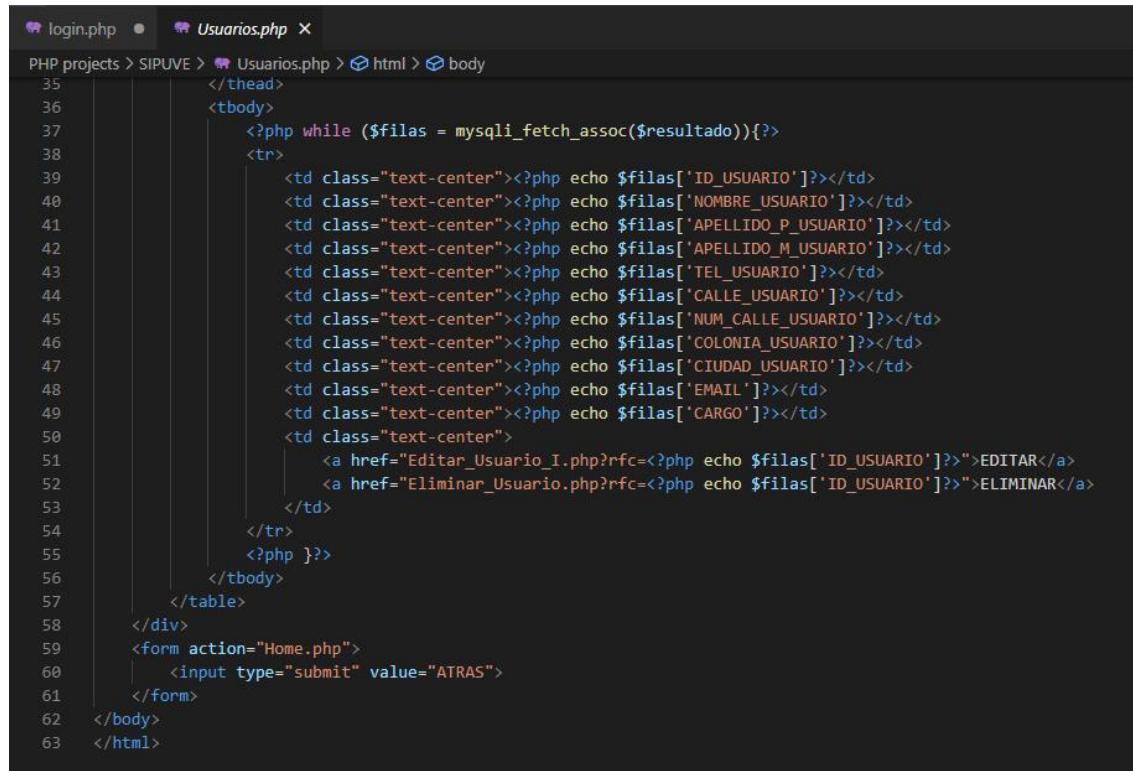
```
login.php • Home.php X
PHP projects > SIPUVE > Home.php > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>SIPUVE</title>
6      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity='
7          <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7c
8  </head>
9  <body>
10 <section id="content" class="text-center">
11     <?php
12         include("conexion.php");
13         session_start();
14     ?>
15     <H1>BIENVENIDO <?php echo $_SESSION['usuario']?></H1>
16     <form action="Usuarios.php">
17         <input type="submit" value="USUARIOS">
18     </form><br>
19     <form action="Proveedores.php">
20         <input type="submit" value="PROVEEDORES">
21     </form><br>
22     <form action="Productos.php">
23         <input type="submit" value="PRODUCTOS">
24     </form><br>
25     <form action="Ventas.php">
26         <input type="submit" value="VENTAS">
27     </form><br>
28     <form action="Det_Ventas.php">
29         <input type="submit" value="DETALLES DE VENTA">
30     </form><br>
31     <form action="Pedidos.php">
32         <input type="submit" value="PEDIDOS">
```

Como se observa en la imagen del código home, es una simple estructura html, mostramos esta ya que en el sistema hay dos páginas de home y esta es la del usuario que tiene cargo de administrador, por lo tanto puede visualizar todo, lo importante que tiene esta sección es el uso de las variables de sesión que ya se mencionó antes, es la variable `$_SESSION['usuario']` que se estableció en Login.php estas variables para que se puedan utilizar se debe usar antes el método `session_start()` es importante entender este funcionamiento ya que son utilizadas mas adelante.

## Modulo de usuarios

Se mostrará a continuación el modulo de usuarios completo que esto conlleva la pagina de inicio de cada modulo, la pagina de agregar un registro en esa tabla, la programación para que haga el registro, la pagina para editar un registro, la programación para editar el registro, la programación para eliminar el registro, esto llevan los módulos de usuario, proveedores, productos. Por lo tanto solo se mostrará uno de estos ya que son muy similares y solo cambia los encabezados de la tabla, y un poco la programación.

### Usuarios.php

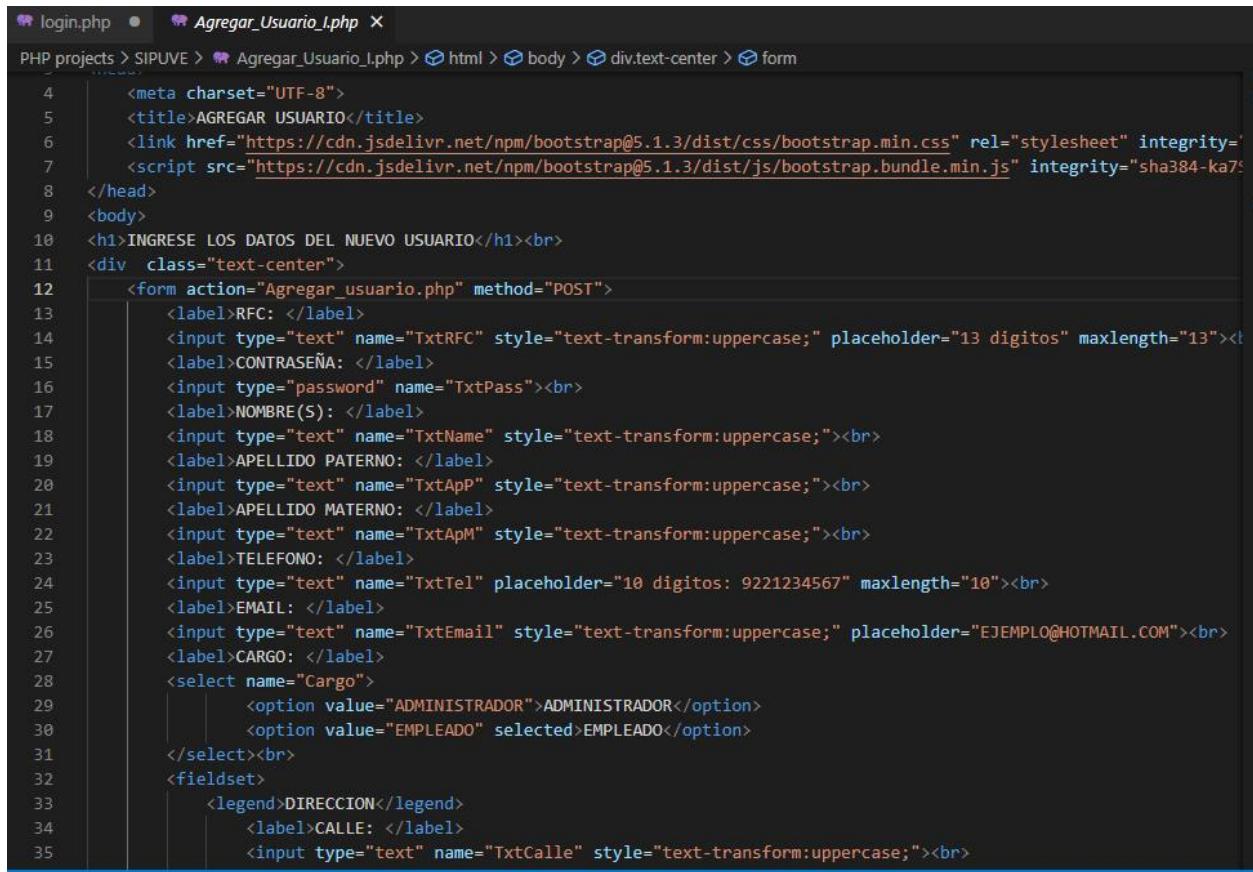


The screenshot shows a code editor with the file 'Usuarios.php' open. The code is a PHP script for displaying user data from a database. It includes a table structure with 10 columns: ID\_USUARIO, NOMBRE\_USUARIO, APELLIDO\_P\_USUARIO, APELLIDO\_M\_USUARIO, TEL\_USUARIO, CALLE\_USUARIO, NUM\_CALLE\_USUARIO, COLONIA\_USUARIO, CIUDAD\_USUARIO, and CARGO. Each row contains these values and two links for EDITAR and ELIMINAR. The code uses mysqli\_fetch\_assoc() to fetch the data. It also includes a form at the bottom to return to 'Home.php'.

```
PHP projects > SIPUVE > Usuarios.php > html > body
35     </thead>
36     <tbody>
37         <?php while ($filas = mysqli_fetch_assoc($resultado)){?>
38             <tr>
39                 <td class="text-center"><?php echo $filas['ID_USUARIO']?></td>
40                 <td class="text-center"><?php echo $filas['NOMBRE_USUARIO']?></td>
41                 <td class="text-center"><?php echo $filas['APELLIDO_P_USUARIO']?></td>
42                 <td class="text-center"><?php echo $filas['APELLIDO_M_USUARIO']?></td>
43                 <td class="text-center"><?php echo $filas['TEL_USUARIO']?></td>
44                 <td class="text-center"><?php echo $filas['CALLE_USUARIO']?></td>
45                 <td class="text-center"><?php echo $filas['NUM_CALLE_USUARIO']?></td>
46                 <td class="text-center"><?php echo $filas['COLONIA_USUARIO']?></td>
47                 <td class="text-center"><?php echo $filas['CIUDAD_USUARIO']?></td>
48                 <td class="text-center"><?php echo $filas['EMAIL']?></td>
49                 <td class="text-center"><?php echo $filas['CARGO']?></td>
50                 <td class="text-center">
51                     <a href="Editar_Usuario_I.php?rfc=<?php echo $filas['ID_USUARIO']?>">EDITAR</a>
52                     <a href="Eliminar_Usuario.php?rfc=<?php echo $filas['ID_USUARIO']?>">ELIMINAR</a>
53                 </td>
54             </tr>
55         <?php }?>
56     </tbody>
57 </table>
58 </div>
59 <form action="Home.php">
60     <input type="submit" value="ATRAS">
61 </form>
62 </body>
63 </html>
```

Esta es la pagina principal del modulo donde se verán todos los registros y cada registro podrá ser editado y eliminado, ademas de que se pueden insertar, se mostró la parte del código donde se imprimen los datos con ayuda del método ya mencionado mysqli\_fetch\_assoc.

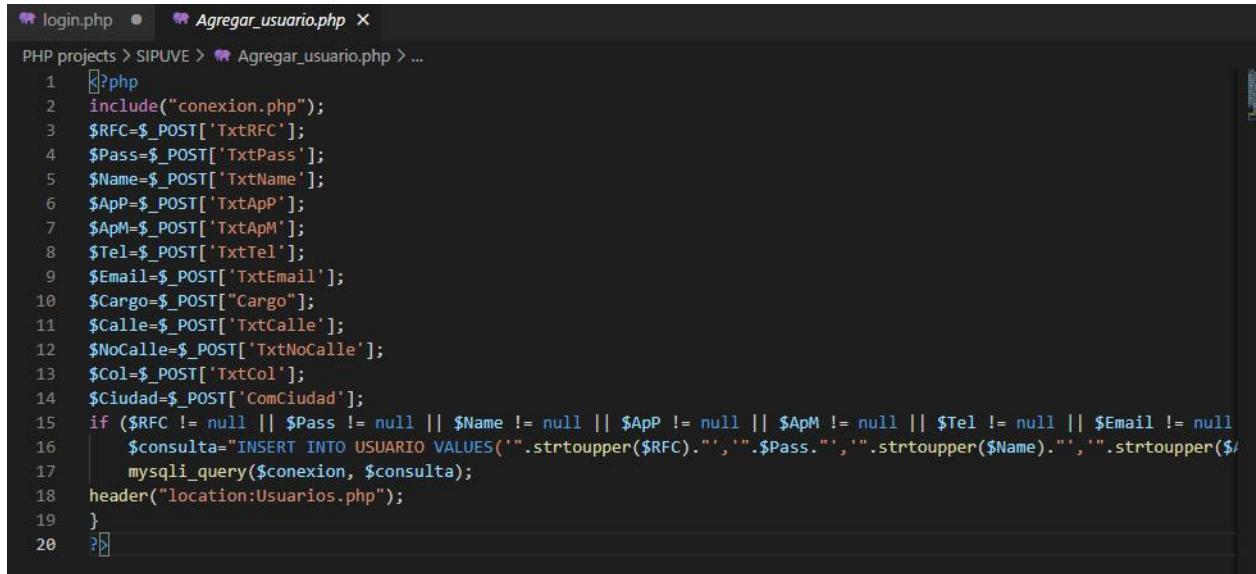
## Agregar\_Usuario\_I.php



```
login.php • Agregar_Usuario_I.php ×
PHP projects > SIPUVE > Agregar_Usuario_I.php > html > body > div.text-center > form
4     <meta charset="UTF-8">
5     <title>AGREGAR USUARIO</title>
6     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity='
7         <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7d0VXfctaJL�
8 </head>
9 <body>
10    <h1>INGRESE LOS DATOS DEL NUEVO USUARIO</h1><br>
11    <div class="text-center">
12        <form action="Agregar_usuario.php" method="POST">
13            <label>RFC: </label>
14            <input type="text" name="TxtRFC" style="text-transform:uppercase;" placeholder="13 digitos" maxlength="13"><br>
15            <label>CONTRASEÑA: </label>
16            <input type="password" name="TxtPass"><br>
17            <label>NOMBRE(S): </label>
18            <input type="text" name="TxtName" style="text-transform:uppercase;"><br>
19            <label>APELIDO PATERNO: </label>
20            <input type="text" name="TxtApP" style="text-transform:uppercase;"><br>
21            <label>APELIDO MATERNO: </label>
22            <input type="text" name="TxtApM" style="text-transform:uppercase;"><br>
23            <label>TELEFONO: </label>
24            <input type="text" name="TxtTel" placeholder="10 digitos: 9221234567" maxlength="10"><br>
25            <label>EMAIL: </label>
26            <input type="text" name="TxtEmail" style="text-transform:uppercase;" placeholder="EJEMPLO@HOTMAIL.COM"><br>
27            <label>CARGO: </label>
28            <select name="Cargo">
29                <option value="ADMINISTRADOR">ADMINISTRADOR</option>
30                <option value="EMPLEADO" selected>EMPLEADO</option>
31            </select><br>
32            <fieldset>
33                <legend>DIRECCION</legend>
34                <label>CALLE: </label>
35                <input type="text" name="TxtCalle" style="text-transform:uppercase;"><br>
```

Esto que se observa en la imagen es la pagina para ingresar un nuevo registros el cual son un conjunto de etiquetas y cajas de texto donde le asignamos el nombre que se ocupará en la programación, estos serán utilizados para mandarlos por medio de un formulario POST hacia la pagina Agregar\_usuario.php donde estará la programación.

## Agregar\_Usuario.php



```
PHP projects > SIPUVE > Agregar_usuario.php > ...
1  <?php
2  include("conexion.php");
3  $RFC=$_POST['TxtRFC'];
4  $Pass=$_POST['TxtPass'];
5  $Name=$_POST['TxtName'];
6  $ApP=$_POST['TxtApP'];
7  $ApM=$_POST['TxtApM'];
8  $Tel=$_POST['TxtTel'];
9  $Email=$_POST['TxtEmail'];
10 $Cargo=$_POST["Cargo"];
11 $Calle=$_POST['TxtCalle'];
12 $NoCalle=$_POST['TxtNoCalle'];
13 $Col=$_POST['TxtCol'];
14 $Ciudad=$_POST['ComCiudad'];
15 if ($RFC != null || $Pass != null || $Name != null || $ApP != null || $ApM != null || $Tel != null || $Email != null
16   $consulta="INSERT INTO USUARIO VALUES('".strtoupper($RFC)."','".$$.$Pass."','".$strtoupper($Name)."','".$strtoupper($
17   mysqli_query($conexion, $consulta);
18 header("location:Usuarios.php");
19 }
20 ?>
```

En esta imagen se encuentra la programación la cual es asignación de variables por POST y entre corchetes el nombre del cuadro de texto donde se ingresarán los valores.

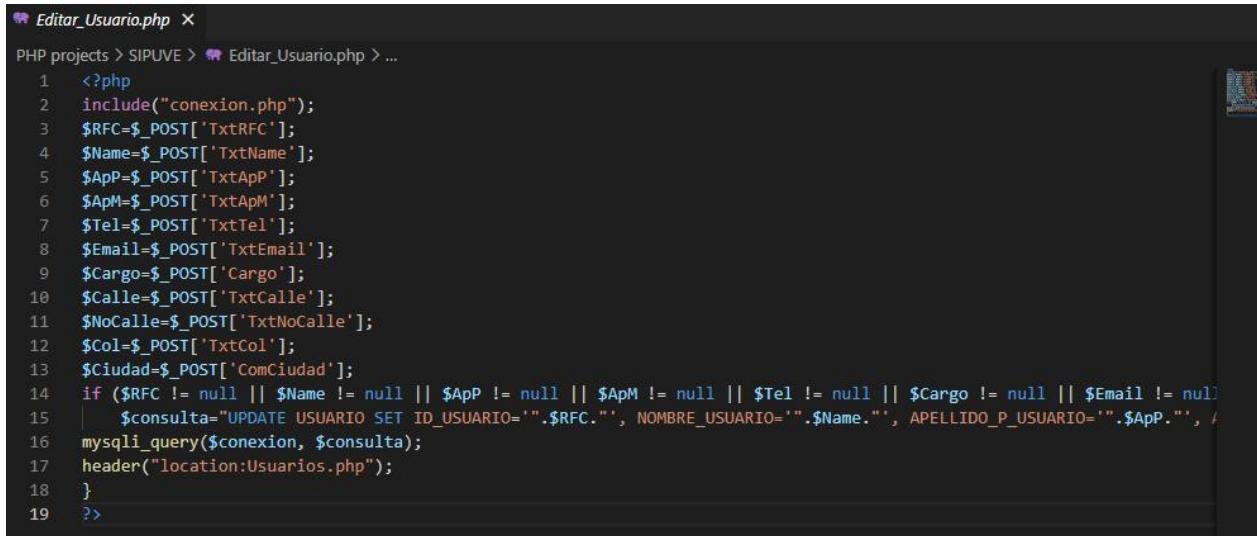
Al final validamos que estos no sean nulos y asignamos la consulta a una variable para que pueda ser ejecutada posteriormente por el método mysqli\_query y redireccionará con ayuda del método header a la pagina Usuarios.php que es la pagina principal del módulo.

## Editar\_Usuario\_I.php

```
PHP projects > SIPUVE > Editar_Usuario_I.php > html > body > div > form.text-center
12  <?php
13  include("conexion.php");
14  $RFC=$_GET['rfc'];
15  $consulta="SELECT * FROM USUARIO WHERE ID_USUARIO='".$RFC."'";
16  $resultado=mysqli_query($conexion, $consulta);
17  while ($filas = mysqli_fetch_assoc($resultado)){
18  ?>
19  <div>
20      <form action="Editar_Usuario.php" method="POST" class="text-center">
21          <label>RFC: </label>
22          <input type="text" name="TxtRFC" value=<?php echo $filas['ID_USUARIO']?>><br>
23          <label>NOMBRE(S): </label>
24          <input type="text" name="TxtName" value=<?php echo $filas['NOMBRE_USUARIO']?>><br>
25          <label>APELLIDO PATERNO: </label>
26          <input type="text" name="TxtApP" value=<?php echo $filas['APELLIDO_P_USUARIO']?>><br>
27          <label>APELLIDO MATERNO: </label>
28          <input type="text" name="TxtApM" value=<?php echo $filas['APELLIDO_M_USUARIO']?>><br>
29          <label>TELEFONO: </label>
30          <input type="number" name="TxtTel" value=<?php echo $filas['TEL_USUARIO']?>><br>
31          <label>EMAIL: </label>
32          <input type="text" name="TxtEmail" value=<?php echo $filas['EMAIL']?>><br>
33          <label>CARGO: </label>
34          <select name="Cargo">
35              <?php if ($filas['CARGO'] == 'ADMINISTRADOR') { ?>
36                  <option value="ADMINISTRADOR" selected>ADMINISTRADOR</option>
37                  <option value="EMPLEADO">EMPLEADO</option>
38              <?php } else{
39                  ?>
40                  <option value="ADMINISTRADOR">ADMINISTRADOR</option>
41                  <option value="EMPLEADO" selected>EMPLEADO</option>
42              <?php ?>
43          </select>
```

Aquí se encuentra la página para editar un registro y es lo mismo que la página para agregar un registro solo se uso el método mysqli\_fetch\_assoc para traer los datos que se ingresaron con eso se reproducen en las cajas de texto para su actualización.

## Editar\_Usuario.php



```
Editor_Usuario.php ×
PHP projects > SIPUVE > Editor_Usuario.php > ...
1  <?php
2  include("conexion.php");
3  $RFC=$_POST['TxtRFC'];
4  $Name=$_POST['TxtName'];
5  $ApP=$_POST['TxtApP'];
6  $ApM=$_POST['TxtApM'];
7  $Tel=$_POST['TxtTel'];
8  $Email=$_POST['TxtEmail'];
9  $Cargo=$_POST['Cargo'];
10 $Calle=$_POST['TxtCalle'];
11 $NoCalle=$_POST['TxtNoCalle'];
12 $Col=$_POST['TxtCol'];
13 $Ciudad=$_POST['ComCiudad'];
14 if ($RFC != null || $Name != null || $ApP != null || $ApM != null || $Tel != null || $Cargo != null || $Email != null)
15 |   $consulta="UPDATE USUARIO SET ID_USUARIO='".$RFC."', NOMBRE_USUARIO='".$Name."', APELLIDO_P_USUARIO='".$ApP."', "
16 mysqli_query($conexion, $consulta);
17 header("location:Usuarios.php");
18 }
19 ?>
```

Como se observa en la imagen es lo mismo que la programación al momento de hacer una inserción solo cambia la palabra reservada INSERT por UPDATE y los campos a actualizar, prácticamente cambia la consulta.

## Eliminar\_Usuario.php

```
 Eliminar_Usuario.php X
PHP projects > SIPUVE > Eliminar_Usuario.php > ...
1  <?php
2  include("conexion.php");
3  $RFC=$_GET['rfc'];
4  $consulta="DELETE FROM USUARIO WHERE ID_USUARIO='".$RFC."'";
5  $resultado=mysqli_query($conexion, $consulta);
6  header("location:Usuarios.php");
7  ?>
```

Este solo lleva pura programación, pasamos al momento de redireccionar a la pagina Eliminar\_Usuario.php un id, este id es rfc con el cual identificamos la fila que se quiere eliminar, que es obtenida por la variable \$\_GET, al final se realiza la instrucción para borrar ese registro con el id.

## Capturas de funcionalidad

### Index

INICIO DE SESION

USUARIO: INGRESE SU RFC

CONTRASEÑA: INGRESE SU CONTRASEÑA

ACCEDER

Error en el index al momento de ingresar un usuario incorrecto

INICIO DE SESION

USUARIO: INGRESE SU RFC

CONTRASEÑA: INGRESE SU CONTRASEÑA

ACCEDER

ERROR... CONTRASEÑA O USUARIO INCORRECTO

Si es correcto muestra la pagina de inicio

### Página de inicio (HOME)

BIENVENIDO POML000901QDA

USUARIOS

PROVEEDORES

PRODUCTOS

VENTAS

DETALLES DE VENTA

PEDIDOS

DETALLES DE PEDIDO

[CERRAR SESION](#)

## Página de Usuarios

ID	NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO	TELÉFONO	CALLE	NO. CALLE	COLONIA	CIUDAD	EMAIL	CARGO	ACCIONES
GAMI000511ABC	MICHAEL	GAMAS	-	9221234567	CINCO DE MAYO	1	OAXAQUEÑA	COSOLEACAQUE	L18230724@MINATITLAN.TECNM.MX	EMPLEADO	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
POML000901QDA	LUIS EDUARDO	POZOS	MIRANDA	9212575914	FRANCISCO SARABIA	28	LA BOMBA	MINATITLAN	L18230724@MINATITLAN.TECNM.MX	ADMINISTRADOR	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>

[ATRAS](#)

## Agregar un usuario

INGRESE LOS DATOS DEL NUEVO USUARIO

RFC:

CONTRASEÑA:

NOMBRE(S):

APELLIDO PATERNO:

APELLIDO MATERNO:

TELÉFONO:

EMAIL:

CARGO:

DIRECCIÓN

CALLE:

NO. CALLE:

COLONIA:

CIUDAD:

[AGREGAR](#) [ATRAS](#)

AGREGAR USUARIO

INGRESE LOS DATOS DEL NUEVO USUARIO

RFC:	POML000901QDZ
CONTRASEÑA:	****
NOMBRE(S):	LUIS
APPELLIDO PATERNO:	MIRANDA
APPELLIDO MATERNO:	POSOZ
TELEFONO:	9212575917
EMAIL:	L18330724@MINATITL
CARGO:	EMPLEADO

DIRECCION

CALLE:	BENITO JUAREZ
NO. CALLE:	45
COLONIA:	HIDALGO
CIUDAD:	COATZACOALCOS

[AGREGAR](#) [ATRAS](#)

SIPUVE USUARIOS

USUARIOS

AGREGAR USUARIO

ID	NOMBRE	APPELLIDO PATERNO	APPELLIDO MATERNO	TELEFONO	CALLE	NO. CALLE	COLONIA	CIUDAD	EMAIL	CARGO	AACCIONES
GAMI000511ABC	MICHAEL	GAMAS	-	9221234567	CINCO DE MAYO	1	OAXAQUEÑA	COSOLEACAQUE	L18230724@MINATITLAN.TECNM.MX	EMPLEADO	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
POML000901QDA	LUIS EDUARDO	POZOS	MIRANDA	9212575914	FRANCISCO SARABIA	28	LA BOMBA	MINATITLAN	L18230724@MINATITLAN.TECNM.MX	ADMINISTRADOR	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
POML000901QDZ	LUIS	MIRANDA	POSOZ	9212575917	BENITO JUAREZ	45	HIDALGO	COATZACOALCOS	L18330724@MINATITLAN.TECNM.MX	EMPLEADO	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>

[ATRAS](#)

## Edición de un registro

EDITAR USUARIO

EDITE LOS DATOS DEL USUARIO

RFC:	POML000901QDZ
NOMBRE(S):	LUIS
APPELLIDO PATERNO:	MIRANDA
APPELLIDO MATERNO:	POSOZ
TELEFONO:	9212575917
EMAIL:	L18330724@MINATITL
CARGO:	EMPLEADO
CALLE:	BENITO JUAREZ
NO. CALLE:	45
COLONIA:	HIDALGO
CIUDAD:	COATZACOALCOS

[ACTUALIZAR](#) [ATRAS](#)

EDITAR USUARIO

localhost/PHP projects/SIPUVE/Editar\_Usuario\_I.php?rfc=POML000901QDZ

90%

EDITE LOS DATOS DEL USUARIO

RFC:	POML000901QDZ
NOMBRE(S):	LUIS
APPELLIDO PATERNO:	MIRANDA
APPELLIDO MATERNO:	POZOS
TELEFONO:	9212575917
EMAIL:	L18330724@MINATITL
CARGO:	EMPLEADO
CALLE:	BENITO JUEAREZ
NO. CALLE:	45
COLONIA:	HIDALGO
CIUDAD:	ACAYUCAN
<a href="#">ACTUALIZAR</a> <a href="#">ATRAS</a>	

## Resultado de edición

SIPUVE USUARIOS

localhost/PHP projects/SIPUVE/Usuarios.php

90%

USUARIOS

AGREGAR USUARIO

ID	NOMBRE	APPELLIDO PATERNO	APPELLIDO MATERNO	TELEFONO	CALLE	NO. CALLE	COLONIA	CIUDAD	EMAIL	CARGO	ACCIONES
GAMI000511ABC	MICHAEL	GAMAS	-	9221234567	CINCO DE MAYO	1	OAXAQUEÑA	COSOLEACAQUE	L18230724@MINATITLAN.TECNM.MX	EMPLEADO	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
POML000901QDA	LUIS EDUARDO	POZOS	MIRANDA	9212575914	FRANCISCO SARABIA	28	LA BOMBA	MINATITLAN	L18230724@MINATITLAN.TECNM.MX	ADMINISTRADOR	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
POML000901QDZ	LUIS	MIRANDA	POZOS	9212575917	BENITO JUEAREZ	45	HIDALGO	ACAYUCAN	L18330724@MINATITLAN.TECNM.MX	EMPLEADO	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>

[ATRAS](#)

## Eliminar registro

SIPUVE USUARIOS

localhost/PHP projects/SIPUVE/Usuarios.php

90%

USUARIOS

AGREGAR USUARIO

ID	NOMBRE	APPELLIDO PATERNO	APPELLIDO MATERNO	TELEFONO	CALLE	NO. CALLE	COLONIA	CIUDAD	EMAIL	CARGO	ACCIONES
GAMI000511ABC	MICHAEL	GAMAS	-	9221234567	CINCO DE MAYO	1	OAXAQUEÑA	COSOLEACAQUE	L18230724@MINATITLAN.TECNM.MX	EMPLEADO	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
POML000901QDA	LUIS EDUARDO	POZOS	MIRANDA	9212575914	FRANCISCO SARABIA	28	LA BOMBA	MINATITLAN	L18230724@MINATITLAN.TECNM.MX	ADMINISTRADOR	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>

[ATRAS](#)

## Muestra de ventas

SIPUVE VENTAS

localhost/PHP projects/SIPUVE/Ventas.php

VENTAS

NUEVA VENTA

NO. VENTA	USUARIO	MONTO TOTAL	FECHA	HORA	ACCIONES
202111201	POML000901QDA	25.00	2021-11-20	16:52:51	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
202111202	POML000901QDA	25.00	2021-11-20	17:26:01	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>

[ATRAS](#)

NUEVA VENTA

localhost/PHP projects/SIPUVE/Nueva\_Venta\_I.php

DATOS DE VENTA

CODIGO:

Warning: Undefined array key "codigo" in C:\xampp\htdocs\PHP projects\SIPUVE\Nueva\_Venta\_I.php on line 21

DESCRIPCION  <br /><b>Warning</b>

CANTIDAD  0

PRECIO VENTA  <br /><b>Warning</b>

[AÑADIR](#) [ATRAS](#)

NUEVA VENTA

localhost/PHP projects/SIPUVE/Nueva\_Venta\_I.php#

DATOS DE VENTA

CODIGO:

DESCRIPCION  DORITOS

CANTIDAD  3

PRECIO VENTA  13.00

[AÑADIR](#) [ATRAS](#)

NUEVA VENTA

localhost/PHP projects/SIPUVE/Nueva\_Venta\_I.php#

DATOS DE VENTA

CODIGO:

Warning: Undefined array key "codigo" in C:\xampp\htdocs\PHP projects\SIPUVE\Nueva\_Venta\_I.php on line 21

DESCRIPCION  <br /><b>Warning</b>

CANTIDAD  0

PRECIO VENTA  <br /><b>Warning</b>

[AÑADIR](#) [ATRAS](#)

CODIGO	DESCRIPCION	CANTIDAD	PRECIO VENTA	SUBTOTAL	ACCIONES
123	DORITOS	3	\$13.00	39.00	<a href="#">ELIMINAR</a>

Total \$39.00

[COMPLETAR VENTA](#)

SIPUVE VENTAS

localhost/PHP projects/SIPUVE/Ventas.php

90%

VENTAS

NUEVA VENTA

NO. VENTA	USUARIO	MONTO TOTAL	FECHA	HORA	ACCIONES
202111201	POML000901QDA	25.00	2021-11-20	16:52:51	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
202111202	POML000901QDA	25.00	2021-11-20	17:26:01	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
202111203	POML000901QDA	39.00	2021-11-20	23:51:58	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>

[ATRAS](#)

SIPUVE DETALLES DE VENTA

localhost/PHP projects/SIPUVE/Det\_Ventas.php?

90%

DETALLES DE VENTA

NO. DE VENTA	ID DEL PRODUCTO	CANTIDAD	SUBTOTAL	FECHA	HORA	ACCIONES
202111202	123	1	13	2021-11-20	17:26:01	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
202111201	125	1	12	2021-11-20	17:26:01	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
202111203	123	3	13	2021-11-20	23:51:58	<a href="#">EDITAR</a> <a href="#">ELIMINAR</a>

[Atras](#)

## Anexos

### Anexo A.

1. ¿Para qué requiere del uso de un software en su microempresa?
2. ¿Quién es el responsable de realizar las ventas de los productos?
3. ¿Existe un formato designado al momento de realizar una venta?
4. ¿Cuál es el procedimiento al momento de realizar una venta?
5. ¿Quién es el que realiza los pedidos a los proveedores?
6. ¿Quién es el responsable de registrar las entradas del inventario?
7. ¿Puede describir el proceso de registrar los productos?
8. ¿Quién realiza el conteo físico de inventario?
9. ¿Cuenta con algún formato donde se lleve el seguimiento de su inventario?
10. ¿Puede describir la manera en la que realiza el seguimiento de su inventario?
11. ¿Qué tiempo le tomaría aproximadamente contar su inventario actual?
12. ¿Qué tan frecuente realiza el proceso de conteo físico de inventario?
13. ¿Cuenta con algún tipo de identificación de sus productos? ¿Quiere algún tipo de identificación en específico?
14. ¿Qué información requiere de sus productos?
15. ¿Maneja productos a granel?
16. ¿Cuenta con algún tipo de identificación para sus empleados (código o número)? ¿le gustaría alguno?
17. ¿Qué información cuenta de sus empleados (dirección, número de teléfono,

etc.)?

18. ¿Requiere llevar un control sobre el salario?
19. ¿Dispone con un equipo de cómputo en la microempresa?
20. ¿Que tipo de equipo? R/ Es un equipo de escritorio marca Lenovo.
21. ¿Sus empleados tienen experiencia en el uso de los equipos de cómputo?
22. ¿Cuántos empleados tiene a su cargo?
23. ¿Existe algún factor que le dificulte el trabajo a su empleado?
24. ¿Cuenta con un logo en su microempresa?, ¿Requiere de uno?
25. ¿Requiere de algunos colores en específico en la interfaz de su sistema?
26. ¿De las siguientes imágenes cual es mas de su agrado?