



Software Inventory

MIGRACIÓN BASE DE DATOS

Proyecto: Software Inventory

Versión: 1

SENA – Servicio Nacional de Aprendizaje
Análisis y Desarrollo de Sistemas de Información

Ficha: 2395871 – G2

Bogotá – 2023

Historia de revisiones

Versión	Autores	Descripción	Fecha
1	Sergio Galindo	Creación del documento	Marzo 2023

Contenido

INTRODUCCIÓN	4
OBJETIVO	4
REFERENCIA.....	4
ALCANCE.....	4
CONTEXTO DE LA MIGRACIÓN	5
PROCESO MIGRACIÓN.....	6

INTRODUCCIÓN

OBJETIVO

Este plan de migración sirve como guía proporcionando información para planificar y desarrollar los pasos necesarios para una correcta migración de la base de datos a un gestor diferente al utilizado en el inicio del proyecto.

REFERENCIA

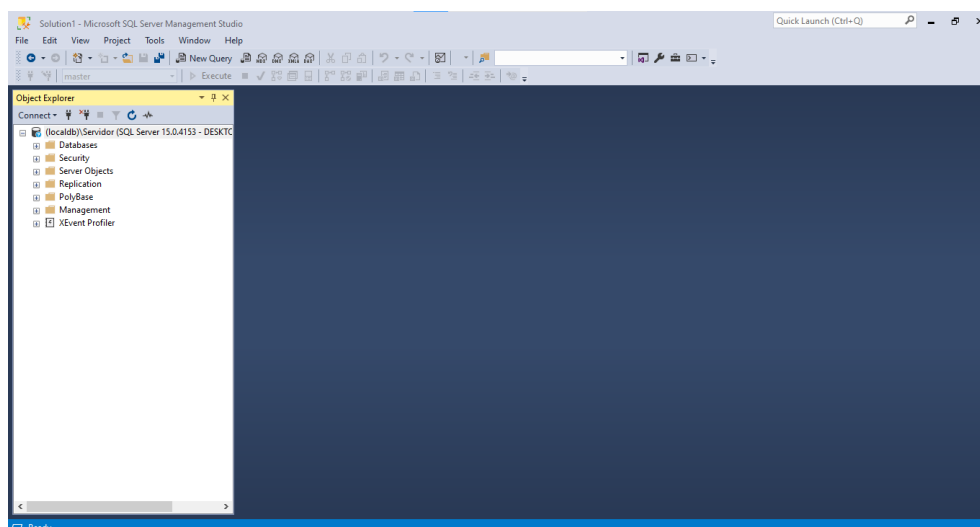
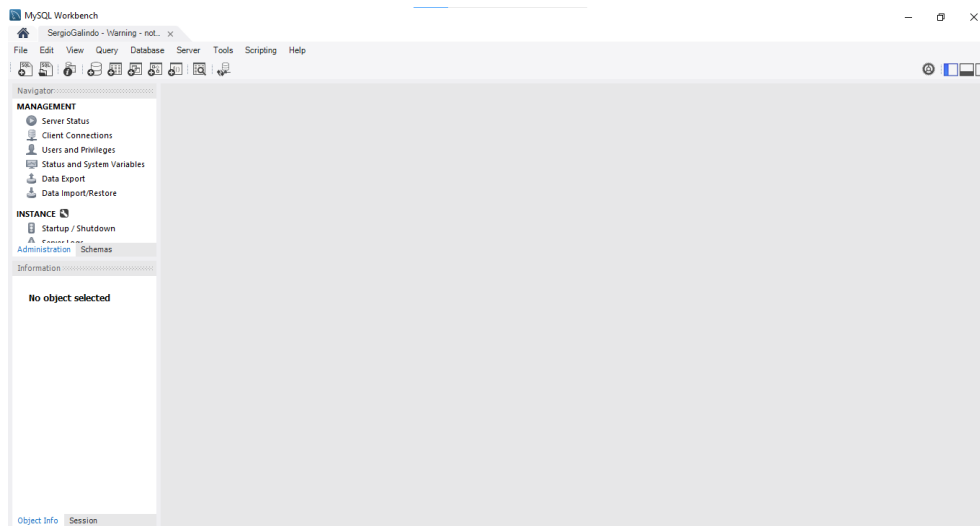
- ISO 29119
- ISO 27001
- ISO 9126
- ISO/IEC 25010 (Describe los tipos y niveles de pruebas de software).
- Especificación de Requerimientos del Proyecto

ALCANCE

El presente documento contiene las directrices para la correcta migración del gestor de la base de datos del proyecto. La migración ira del gestor MySQL al gestor SQL Server.

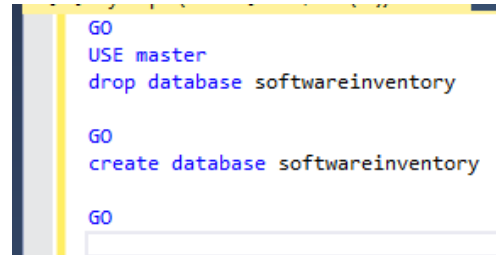
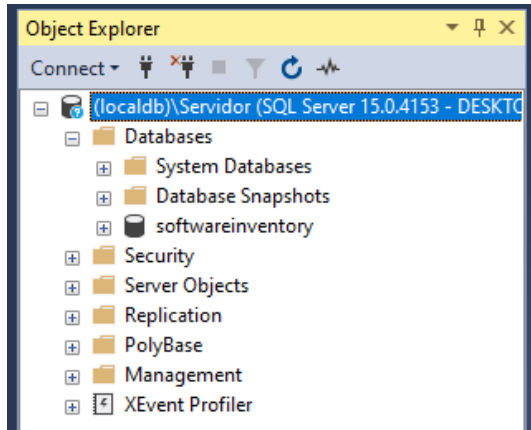
CONTEXTO DE LA MIGRACIÓN

Inicialmente la base de datos del proyecto se implemento sobre el gestor de base de datos MySQL Bajo los parámetros indicados es necesario realizar una migración a otro gestor por lo cual se opto por migrar al gestor de base de datos SQL Server.

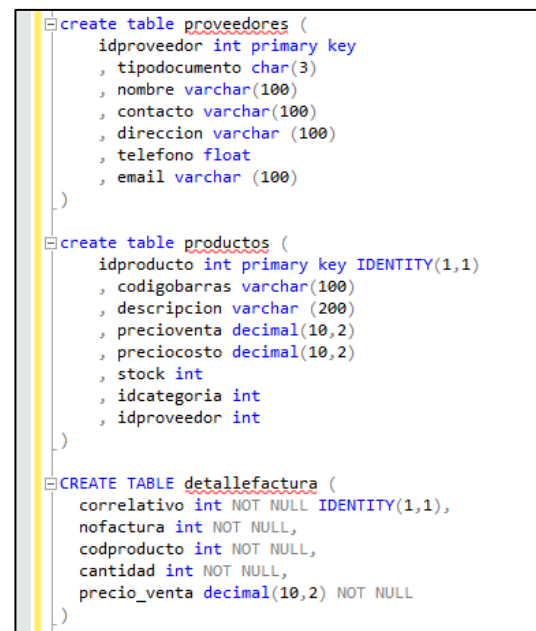
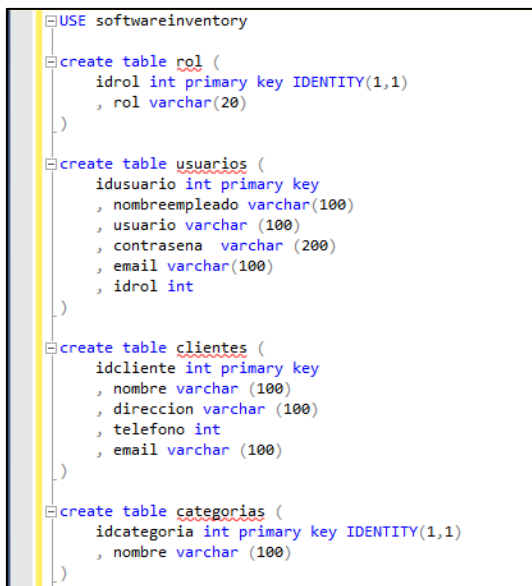


PROCESO MIGRACIÓN

En este proceso de migración primeramente se realiza creación de la base de datos. Para ello se realiza bajo una instancia local.



Procedemos a crear el query donde se realizará la creación de las tablas con sus respectivos atributos.



```

CREATE TABLE detalle_temp (
    correlativo int NOT NULL IDENTITY(17,1),
    token_user varchar(50) NOT NULL,
    codproducto int NOT NULL,
    cantidad int NOT NULL,
    precio_venta decimal(10,2) NOT NULL
)

```

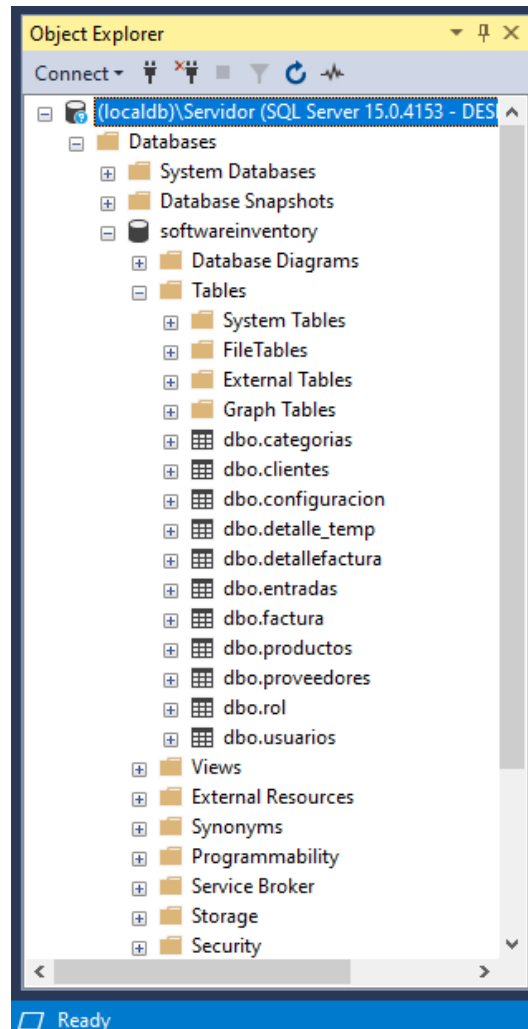
```

create table proveedores (
    idproveedor int primary key
    , tipodocumento char(3)
    , nombre varchar(100)
    , contacto varchar(100)
    , direccion varchar (100)
    , telefono float
    , email varchar (100)
)

create table productos (
    idproducto int primary key IDENTITY(1,1)
    , codigobarras varchar(100)
    , descripcion varchar (200)
    , precioventa decimal(10,2)
    , preciocosto decimal(10,2)
    , stock int
    , idcategoria int
    , idproveedor int
)

CREATE TABLE detallefactura (
    correlativo int NOT NULL IDENTITY(1,1),
    nofactura int NOT NULL,
    codproducto int NOT NULL,
    cantidad int NOT NULL,
    precio_venta decimal(10,2) NOT NULL
)

```



Una vez tengamos creadas las entidades procedemos a realizar procesos de CRUD para verificar la correcta migración de las tablas.

```
USE softwareinventory

insert into rol (rol) values
('Administrador'),
('Compra'),
('Vendedor')

insert into usuarios values
(1, 'pruebaadmin', 'padmin', '21232f297a57a5a743894a0e4a801fc3', 'sergio.galindo.cruz@gmail.com', 1),
(2, 'prueba compra', 'pcompra', '21232f297a57a5a743894a0e4a801fc3', 'sergio.galindo.cruz@gmail.com', 2),
(3, 'pruebaventa', 'pvnta', '21232f297a57a5a743894a0e4a801fc3', 'sergio.galindo.cruz@gmail.com', 3)

insert into clientes values
(9, 'Publico General', 'Calle 1 Carrera 1', 2999999, 'PublicoGeneral@notiene.com')

insert into categorias (nombre) values
('Categoria General')

insert into proveedores values
(9, 'NIT', 'Proveedor General', 'Contacto General', 'Calle 1 Carrera 1', 2999999, 'ProveedorGeneral@notiene.com')

insert into productos (codigobarras, descripcion, precioventa, preciocosto, stock, idcategoria, idproveedor) values
('ABC123', 'Productos General', 1, 1, 100, 1, 9)

INSERT INTO configuracion (dni, nombre, razon_social, telefono, email, direccion, iva) VALUES
(123456789, 'Software inventory', 'Software inventory', 2999999, 'Softwareinventory@Softwareinventory.com', 'Colombia - Bogotá', '1.19')
```

Luego de tener las entidades y sus atributos procedemos con la creación de la relación de las tablas.

```
USE softwareinventory

ALTER TABLE usuarios ADD CONSTRAINT FK_rol FOREIGN KEY (idrol) REFERENCES rol(idrol)
ALTER TABLE productos ADD CONSTRAINT FK_categorias FOREIGN KEY (idcategoria) REFERENCES categorias(idcategoria)
ALTER TABLE productos ADD CONSTRAINT FK_proveedores FOREIGN KEY (idproveedor) REFERENCES proveedores(idproveedor)
ALTER TABLE detallefactura ADD CONSTRAINT FK_productos FOREIGN KEY (codproducto) REFERENCES productos(idproducto)
ALTER TABLE detallefactura ADD CONSTRAINT FK_detallefactura_factura FOREIGN KEY (nofactura) REFERENCES factura(nofactura)
ALTER TABLE detalle_temp ADD CONSTRAINT FK_detalle_temp_productos FOREIGN KEY (codproducto) REFERENCES productos(idproducto)
ALTER TABLE factura ADD CONSTRAINT FK_clientes FOREIGN KEY (codcliente) REFERENCES clientes(idcliente)
ALTER TABLE entradas ADD CONSTRAINT FK_entradas_productos FOREIGN KEY (codproducto) REFERENCES productos(idproducto)
ALTER TABLE entradas ADD CONSTRAINT FK_entradas_usuarios FOREIGN KEY (usuario_id) REFERENCES usuarios(idusuario)
```

Por último, procedemos con la migración de los procedimientos que luego serán utilizados por nuestro código fuente de la aplicación.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Description: <Procedimiento almacenado para agregar productos al detallado temporal>
--
-- =====
CREATE PROCEDURE add_detalle_temp
    @codigo INT,
    @cantidad INT,
    @token_user VARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;
    DROP TABLE IF EXISTS #precio_actual
    SELECT precioventa
    INTO #precio_actual
    FROM productos
    WHERE idproducto = @codigo

    INSERT INTO detalle_temp(token_user, codproducto, cantidad, precio_venta)
    VALUES (@token_user, @codigo, @cantidad, (SELECT * From #precio_actual))

    SELECT tmp.correlativo, tmp.codproducto, p.descripcion, tmp.cantidad, tmp.precio_venta
    FROM detalle_temp tmp
    INNER JOIN productos p
    ON tmp.codproducto = p.idproducto
    WHERE tmp.token_user = token_user
END
GO
```



```
-- =====
-- Description: <Procedimiento almacenado para eliminar productos del detallado temporal>
-- =====

CREATE PROCEDURE del_detalle_temp
    @id_detalle INT,
    @token VARCHAR(50)
AS
BEGIN
    DELETE FROM detalle_temp WHERE correlativo = @id_detalle
    SELECT tmp.correlativo, tmp.codproducto, p.descripcion, tmp.cantidad, tmp.precio_venta
    FROM detalle_temp tmp
    INNER JOIN productos p
    ON tmp.codproducto = p.idproducto
    WHERE tmp.token_user = @token
END
GO
```

```
-- =====
-- Description: <Procedimiento almacenado para generar venta y guardar esta informacion>
-- =====

CREATE PROCEDURE procesar_venta
    @cod_usuario INT,
    @cod_cliente INT,
    @token VARCHAR(50),
    @factura INT,
    @registros INT,
    @total DECIMAL(10,2),
    @nueva_existencia int,
    @existencia_actual int,
    @tmp_cod_producto int,
    @tmp_cant_producto int,
    @a int
AS
BEGIN
    SET @a = 1

    DROP TABLE IF EXISTS #tbl_tmp_tokenuser
    CREATE TABLE #tbl_tmp_tokenuser(
        id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
        cod_prod BIGINT,
        cant_prod int)

    IF (SELECT COUNT(*) FROM detalle_temp WHERE token_user = @token) > 0
    BEGIN
        INSERT INTO #tbl_tmp_tokenuser(cod_prod, cant_prod)
        SELECT codproducto, cantidad FROM detalle_temp WHERE token_user = @token
    END
```

```
INSERT INTO factura (usuario,codcliente)
VALUES (@cod_usuario, @cod_cliente);

SET @factura = SCOPE_IDENTITY();

INSERT INTO detallefactura(nofactura,codproducto,cantidad,precio_venta)
SELECT (@factura) AS nofactura, codproducto, cantidad,precio_venta
FROM detalle_temp
WHERE token_user = @token;

WHILE @a <= (SELECT COUNT(*) FROM detalle_temp WHERE token_user = @token)
BEGIN
    DROP TABLE IF EXISTS #tempprod1
    CREATE TABLE #tempprod1 (tmp_cod_producto INT, tmp_cant_producto INT)
    INSERT INTO #tempprod1 (tmp_cod_producto, tmp_cant_producto)
    SELECT cod_prod, cant_prod
    FROM #tbl_tmp_tokenuser
    WHERE id = @a;

    DROP TABLE IF EXISTS #tempprod2
    CREATE TABLE #tempprod2 (existencia_actual INT)
    INSERT INTO #tempprod2 (existencia_actual)
    SELECT stock
    FROM productos
    WHERE idproducto = @tmp_cod_producto;

    SET @nueva_existencia = ((Select existencia_actual from #tempprod2) - (Select tmp_cant_producto From #tempprod1));
    UPDATE productos SET stock = @nueva_existencia WHERE idproducto = @tmp_cod_producto;
    SET @a = @a+1;
END
```

```
-- Description: <Procedimiento almacenado para actualizar precio y stock de la tabla de productos>
-- =====
CREATE PROCEDURE actualizar_precio_producto
    @n_cantidad INT,
    @n_precio DECIMAL(10,2),
    @codigo INT,
    @nueva_existencia int,
    @nuevo_total decimal(10,2),
    @nuevo_precio decimal(10,2),
    @cant_actual int,
    @pre_actual decimal(10,2),
    @actual_existencia int,
    @actual_precio decimal(10,2)
AS
BEGIN
    DROP TABLE IF EXISTS #temporalproductos
    CREATE TABLE #temporalproductos (actual_precio INT, actual_existencia INT)
    INSERT INTO #temporalproductos (actual_precio, actual_existencia)
        SELECT precioventa, stock
        FROM productos
        WHERE idproducto = @codigo;

    SET @nueva_existencia = (Select actual_existencia From #temporalproductos) + @n_cantidad
    SET @nuevo_total = @n_precio
    SET @nuevo_precio = @nuevo_total





























    UPDATE productos
        SET stock = @nueva_existencia, precioventa = @nuevo_precio
        WHERE idproducto = @codigo

    SELECT @nueva_existencia, @nuevo_precio
END
GO
```

Migración PHP conexión SQL Server

Primeramente se procede a realizar configuración del Xampp para que este tome la conexión del servidor local de SQL Server.

En la carpeta C:\xampp\php\ext se debe proceder a dejar los drivers descargados de la página oficial de Microsoft para PHP.

	Microsoft DRIVERS FOR PHP FOR MICRO...	28/02/2023 6:02 p. m.	Formato de texto ...	208 KB
	Microsoft DRIVERS FOR PHP FOR MICRO...	28/02/2023 6:02 p. m.	Documento de te...	11 KB
	php_pdo_sqlsrv_80_nts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	318 KB
	php_pdo_sqlsrv_80_nts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	280 KB
	php_pdo_sqlsrv_80_ts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	320 KB
	php_pdo_sqlsrv_80_ts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	283 KB
	php_pdo_sqlsrv_81_nts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	318 KB
	php_pdo_sqlsrv_81_nts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	281 KB
	php_pdo_sqlsrv_81_ts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	321 KB
	php_pdo_sqlsrv_81_ts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	283 KB
	php_pdo_sqlsrv_82_nts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	318 KB
	php_pdo_sqlsrv_82_nts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	281 KB
	php_pdo_sqlsrv_82_ts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	321 KB
	php_pdo_sqlsrv_82_ts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	283 KB
	php_sqlsrv_80_nts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	335 KB
	php_sqlsrv_80_nts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	289 KB
	php_sqlsrv_80_ts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	341 KB
	php_sqlsrv_80_ts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	294 KB
	php_sqlsrv_81_nts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	336 KB
	php_sqlsrv_81_nts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	289 KB
	php_sqlsrv_81_ts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	343 KB
	php_sqlsrv_81_ts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	294 KB
	php_sqlsrv_82_nts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	336 KB
	php_sqlsrv_82_nts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	289 KB
	php_sqlsrv_82_ts_x64.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	343 KB
	php_sqlsrv_82_ts_x86.dll	28/02/2023 6:03 p. m.	Extensión de la ap...	295 KB
	release	28/02/2023 6:02 p. m.	Documento de te...	4 KB
	SQLSRV_Readme	28/02/2023 6:02 p. m.	Chrome HTML Do...	16 KB

De acuerdo a la versión de PHP instalada se debe agregar a la configuración del apache en el módulo php.ini en la sección extensiones los nombres de los archivos alojados en el paso anterior.

```
php: Bloc de notas
Archivo Edición Formato Ver Ayuda
extension=bz2
extension=curl
;extension=ffi
;extension=ftp
extension=fileinfo
;extension=gd
extension=gettext
;extension=gmp
;extension=intl
;extension=imap
;extension=ldap
extension=mbstring
extension=exif ; Must be after mbstring as it depends on it
extension=mysqli
;extension=oci8_12c ; Use with Oracle Database 12c Instant Client
;extension=oci8_19 ; Use with Oracle Database 19 Instant Client
;extension=odbc
;extension=openssl
;extension=pdo_firebird
extension=pdo_mysql
;extension=pdo_oci
;extension=pdo_odbc
;extension=pdo_pgsql
extension=pdo_sqlite
;extension=pgsql
;extension=shmop
extension=php_sqlsrv_82_nts_x64.dll
extension=php_sqlsrv_82_nts_x86.dll
extension=php_sqlsrv_82_ts_x64.dll
extension=php_sqlsrv_82_ts_x86.dll

; The MIBS data available in the PHP distribution must be installed.
; See https://www.php.net/manual/en/snmp.installation.php
;extension=snmp

;extension=soap

Línea 941, columna 1 100% Windows (CRLF) UTF-8
```

Procedemos con la configuración del PHP para cambiar las conexiones a SQL Server.

```
conexion.php - softwareinventory - Visual Studio Code
conexion.php
1 <?php
2 $servername = "localhost";
3 $conexioninfo =
4     array("Database"=>"softwareinventory"
5         , "UID"=>"softwareinventory"
6         , "PWD"=>"softwareinventory");
7 $conectar = sqlsrv_connect($servername, $conexioninfo);
8
9 if($conectar){
10     //echo "conexion correcta";
11 }else {
12     echo "Revise los datos de conexión.<br> Ha habido un error.<br />";
13     die( print_r( sqlsrv_errors(), true));
14 }
15 ?>
```

```
1 <?php
2 include_once "includes/header.php";
3 // Validar producto
4
5 if (empty($_REQUEST['id'])) {
6     header("Location: listar_productos.php");
7 } else {
8     $id_producto = $_REQUEST['id'];
9     if (!is_numeric($id_producto)) {
10         header("Location: listar_productos.php");
11     }
12     $query_producto = sqlsrv_query($conectar, "SELECT idproducto, descripcion, idproveedor, precioventa, stock FROM productos WHERE idproducto = $id_producto");
13     $result_producto = sqlsrv_num_rows($query_producto);
14
15     if ($result_producto > 0) {
16         $data_producto = sqlsrv_fetch_assoc($query_producto);
17     } else {
18         header("Location: listar_productos.php");
19     }
20 }
21 // Agregar Productos a entrada
22 if (!empty($_POST)) {
23     $alert = "";
24     if (!empty($_POST['cantidad']) || !empty($_POST['precio']) || !empty($_POST['producto_id'])) {
25         $precio = $_POST['precio'];
26         $cantidad = $_POST['cantidad'];
27         $producto_id = $_GET['id'];
28         $usuario_id = $_SESSION['idUser'];
29         $query_insert = sqlsrv_query($conectar, "INSERT INTO entradas(codproducto,cantidad,precio,usuario_id) VALUES ($producto_id,$cantidad,$precio,$usuario_id)");
30         if ($query_insert) {
31             // ejecutar procedimiento almacenado
32             $query_upd = sqlsrv_query($conectar, "CALL actualizar_precio_producto($cantidad,$precio,$producto_id)");
33             $result_upd = sqlsrv_num_rows($query_upd);
```

```
1 <?php
2 include_once "includes/header.php";
3 if (!empty($_POST)) {
4     $alert = "";
5     if (empty($_POST['nombre'])) {
6         $alert = "<div class='alert alert-primary' role='alert'>
7             Todo los campos son requeridos
8         </div>";
9     } else {
10         $idcategoria = $_GET['id'];
11         $nombre = $_POST['nombre'];
12         $query_update = sqlsrv_query($conectar, "UPDATE categorias SET nombre = '$nombre' WHERE idcategoria = $idcategoria");
13         if ($query_update) {
14             $alert = "<div class='alert alert-primary' role='alert'>
15                 Modificado
16             </div>";
17         } else {
18             $alert = "<div class='alert alert-primary' role='alert'>
19                 Error al Modificar
20             </div>";
21         }
22     }
23 }
24 // Validar producto
25
26 if (empty($_REQUEST['id'])) {
27     header("Location: listar_categorias.php");
28 } else {
29     $idcategoria = $_REQUEST['id'];
30     if (!is_numeric($idcategoria)) {
31         header("Location: listar_categorias.php");
32     }
33 }
```

The screenshot shows a web browser window with the address bar displaying 'listar_categorias.php - softwareinventory - Visual Studio Code'. The browser window is displaying a PHP script in a code editor. The script is titled 'listar_categorias.php' and is located at 'index.php C:\softwareinventory(MySql)'. The script is a PHP file that displays a table of categories. The table has columns for ID, Category Name, and Actions. The first row shows a category with ID 1 and name 'CATEGORIAS'. The second row shows a category with ID 1 and name 'ACCIONES'. The script uses a database connection and SQL queries to fetch data.

```

1 <?php include_once "includes/header.php"; ?>
2 <!-- Contenido -->
3 <div id="contenido">
4 <div class="container-fluid">
5 <div class="d-sm-flex align-items-center justify-content-between mb-4">
6 <h1 class="h3 mb-0 text-gray-800">Categorias</h1>
7 <a href="registro_categorias.php" class="btn btn-primary">Nuevo</a>
8 </div>
9 <div class="row">
10 <div class="col-lg-12">
11 <div class="table-responsive">
12 <table class="table table-striped table-bordered" id="table">
13 <thead class="thead-dark">
14 <tr>
15 <th>ID</th>
16 <th>CATEGORIAS</th>
17 <th>ACCIONES</th>
18 </tr>
19 </thead>
20 <tbody>
21 <tr>
22 <td><?php echo $data['idcategoria'];></td>
23 <td><?php echo $data['nombre'];></td>
24 <td><?php echo $data['acciones'];></td>
25 </tr>
26 </tbody>
27 </table>
28 </div>
29 </div>
30 </div>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>

```

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda registro_categoria.php - SoftwareInventory - Visual Studio Code

EXPLORADOR ... index.php \ registro_categoria.php x index.php C:\SoftwareInventory(MySql)

SOFTWAREINVENTORY

- style
- vendor
- agregar_producto.php
- editar_categoria.php
- editar_cliente.php
- editar_productos.php
- editar_proveedores.php
- editar_usuarios.php
- eliminar_categoria.php
- eliminar_cliente.php
- eliminar_productos.php
- eliminar_proveedores.php
- eliminar_usuarios.php
- index.php
- listar_categoria.php
- listar_cliente.php
- listar_productos.php
- listar_proveedores.php
- listar_usuarios.php
- modal.php
- registro_categoria.php
- registro_cliente.php
- registro_productos.php
- registro_proveedores.php
- registro_usuarios.php
- registro_venta.php
- ESQUEMA
- LÍNEA DE TIEMPO

sistema > registro_categoria.php

```

1 <?php include_once "includes/header.php"; ?>
2 <?php
3     if (empty($_POST)) {
4         $alert = "";
5         if (empty($_POST['nombre'])) {
6             $alert = '<div class="alert alert-danger" role="alert">
7                 Todo los campos son obligatorio
8             </div>';
9         } else {
10            $nombre = $_POST['nombre'];
11            $idcategoria = 1;
12
13            $result = 0;
14            if (is_numeric($idcategoria) and $idcategoria != 0) {
15                $query = sqlsrv_query($conectar, "SELECT * FROM categorias where nombre = '$nombre'");
16                $result = mysqli_fetch_array($query);
17            }
18            if ($result > 0) {
19                $alert = '<div class="alert alert-danger" role="alert">
20                    La categoria ya existe
21                </div>';
22            } else {
23                $query_insert = sqlsrv_query($conectar, "INSERT INTO categorias(nombre) values ('$nombre')");
24                if ($query_insert) {
25                    $alert = '<div class="alert alert-primary" role="alert">
26                        Categoría Registrada
27                    </div>';
28                } else {
29                    $alert = '<div class="alert alert-danger" role="alert">
30                        Error al Guardar
31                    </div>';
32                }
33            }
34        }
35    }
36}

```

Lin. 1 col. 1 Tamaño de tabulación: 4 UTF-8 CRLF PHP