**Angie**

**sp\_m\_coordenadas\_repartidor**

BEGIN

DECLARE latitudEntregaTR VARCHAR(255);

DECLARE longitudEntregaTR VARCHAR(255);

SELECT odLatitudEntregaTR, odLongitudEntregaTR

INTO latitudEntregaTR, longitudEntregaTR

FROM tblOrden

WHERE idOrden = p\_idOrden;

IF latitudEntregaTR = p\_Latitud AND longitudEntregaTR = p\_Longitud THEN

SELECT 1 AS resultado;

ELSE

SELECT 0 AS resultado;

END IF;

END

**------------------**

**sp\_m\_datos\_tienda\_registrotienda**

BEGIN

UPDATE tbltienda

SET tieNombre = p\_tieNombre,

tieImagen = p\_tieImagen,

tieURLWeb = p\_tieURLWeb,

tieDescripcion = p\_tieDescripcion,

tieCorreo = p\_tieCorreo,

tieTelefono = p\_tieTelefono,

tieDireccion = p\_tieDireccion,

tieCiudad = p\_tieCiudad,

tieEstado = p\_tieEstado,

tieLatitud = p\_tieLatitud,

tieLongitud = p\_tieLongitud

WHERE idTienda = p\_idTienda;

END

**-----------------------**

**\*no sé si es para modificar o agregar nuevos**

**-Modificar**

**sp\_m\_direccion\_persona**

BEGIN

UPDATE tblpersona

SET perUbiLatitud = per\_UbiLatitud,

perUbiLongitud = per\_UbiLongitud

WHERE idDocumentoPersona = id\_DocumentoPersona;

IF ROW\_COUNT() = 0 THEN

INSERT INTO tblpersona (idDocumentoPersona, perUbiLatitud, perUbiLongitud)

VALUES (id\_DocumentoPersona, per\_UbiLatitud, per\_UbiLongitud);

END IF;

END

- Agregar

BEGIN

INSERT INTO tbltienda (

tieNombre,

tieImagen,

tieURLWeb,

tieDescripcion,

tieCorreo,

tieTelefono,

tieDireccion,

tieCiudad,

tieEstado,

tieLatitud,

tieLongitud

) VALUES (

p\_tieNombre,

p\_tieImagen,

p\_tieURLWeb,

p\_tieDescripcion,

p\_tieCorreo,

p\_tieTelefono,

p\_tieDireccion,

p\_tieCiudad,

p\_tieEstado,

p\_tieLatitud,

p\_tieLongitud

);

END

**-----------**

**sp\_m\_estado\_orden\_preparado\_a\_enpreparacion**

BEGIN

UPDATE tblOrden

SET tblOrden.odEstado = 'EN PREPARACIÓN'

WHERE tblOrden.idOrden = idorden;

SELECT tblOrden.odEstado FROM tblOrden

WHERE tblOrden.idOrden = idorden;

END

**-----------------**

**sp\_m\_estado\_orden\_preparado\_a\_rechazado**

BEGIN

UPDATE tblOrden

SET tblOrden.odEstado = 'RECHAZADO'

WHERE tblOrden.idOrden = idorden;

SELECT tblOrden.odEstado FROM tblOrden

WHERE tblOrden.idOrden = idorden;

END

**-----------------**

**sp\_m\_estadorepartidor**

BEGIN

UPDATE tblrepartidor

SET repEstado = p\_repEstado

WHERE idDocumentoPersona = p\_idDocumentoPersona;

END

**-----------------------**

**sp\_m\_estadoantecedente\_repartidor**

BEGIN

UPDATE tblrepartidor

SET repAntecedentesEstado = p\_repAntecedentesEstado

WHERE idDocumentoPersona = p\_idDocumentoPersona;

END

**----------------------**

**sp\_m\_estadolicencia\_repartidor**

BEGIN

UPDATE tblrepartidor

SET repLicenciaEstado = p\_repLicenciaEstado

WHERE idDocumentoPersona = p\_idDocumentoPersona;

END

**---------------------**

**sp\_m\_foto\_usuario**

BEGIN

UPDATE tblusuario

SET usuImagen = p\_usuImagen

WHERE idDocumentoPersona = p\_idDocumentoPersona;

END

**Miguel Sanchez**

**sp\_a\_datos\_tienda\_registrotienda**

BEGIN

-- Verificar si el idDocumentoPersona existe en tblPersona

DECLARE personaExists INT;

SELECT COUNT(\*) INTO personaExists

FROM tblPersona

WHERE idDocumentoPersona = p\_idDocumentoPersona;

-- Si el idDocumentoPersona no existe, no hacemos la inserción

IF personaExists = 0 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'El ID de Documento de Persona no existe';

ELSE

-- Si el idDocumentoPersona existe, proceder con la inserción

INSERT INTO `tblTienda` (

`tieNombre`,

`tieImagen`,

`tieURLWeb`,

`tieDescripcion`,

`tieCorreo`,

`tieTelefono`,

`tieDireccion`,

`tieCiudad`,

`tieEstado`,

`tieVentasMensuales`,

`tieInventarioEstimado`,

`idDocumentoPersona`,

`tieLatitud`,

`tieLongitud`

)

VALUES (

p\_tieNombre,

p\_tieImagen,

p\_tieURLWeb,

p\_tieDescripcion,

p\_tieCorreo,

p\_tieTelefono,

p\_tieDireccion,

p\_tieCiudad,

p\_tieEstado,

p\_tieVentasMensuales,

p\_tieInventarioEstimado,

p\_idDocumentoPersona,

p\_tieLatitud,

p\_tieLongitud

);

END IF;

END

**-------------------------------------------------------------------**

**sp\_a\_reg\_proveedor\_almacen**

BEGIN

INSERT INTO tblproveedor (

provNombre,

provDireccion,

provCiudad,

provDistrito,

provCorreo,

provTelefono,

provRFC

)

VALUES (

p\_provNombre,

p\_provDireccion,

p\_provCiudad,

p\_provDistrito,

p\_provCorreo,

p\_provTelefono,

p\_provRFC

);

END

**------------------------------------------------------------------------**

**sp\_a\_registro\_usuario\_inicio\_sesion**

BEGIN

-- Inserción en la tabla tblpersona

INSERT INTO tblpersona (

idDocumentoPersona,

perNombres,

perApellidos,

perTipo,

perNumeroCelular,

perUbiLatitud,

perUbiLongitud

) VALUES (

p\_idDocumentoPersona,

p\_perNombres,

p\_perApellidos,

p\_perTipo,

p\_perNumeroCelular,

p\_perUbiLatitud,

p\_perUbiLongitud

)

ON DUPLICATE KEY UPDATE

perNombres = p\_perNombres,

perApellidos = p\_perApellidos,

perTipo = p\_perTipo,

perNumeroCelular = p\_perNumeroCelular,

perUbiLatitud = p\_perUbiLatitud,

perUbiLongitud = p\_perUbiLongitud;

-- Inserción en la tabla tblusuario

INSERT INTO tblusuario (

usuCorreo,

usuImagen,

idRolUsuario,

idDocumentoPersona

) VALUES (

p\_usuCorreo,

p\_usuImagen,

p\_idRolUsuario,

p\_idDocumentoPersona

)

ON DUPLICATE KEY UPDATE

usuCorreo = p\_usuCorreo,

usuImagen = p\_usuImagen,

idRolUsuario = p\_idRolUsuario;

-- Devolver el idUsuario y el idRolUsuario para confirmar la inserción

SELECT idUsuario, idRolUsuario

FROM tblusuario

WHERE usuCorreo = p\_usuCorreo;

END

**---------------------------------------------------------------------**

**sp\_c\_consultar\_categoria\_aprendizaje**

BEGIN

SELECT DISTINCT apreCategoriaRecurso

FROM tblAprendizaje;

END

**sp\_c\_consultar\_eventos\_aprendizaje**

BEGIN

SELECT

idEventosV,

eveFechaPublicacion AS evFechaPublicacion,

eveEstado AS evEstado,

evTitulo,

evDescripcion,

evLink

FROM tbleventosvivo;

END

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-**

BEGIN

DECLARE chatExists INT;

DECLARE newChatId INT;

-- Check if the chat exists

SELECT COUNT(\*) INTO chatExists

FROM tblchat

WHERE idChat = p\_id\_orden;

-- If the chat does not exist, create a new chat

IF chatExists = 0 THEN

-- Insert into tblchat and create a new chat record

INSERT INTO tblchat (chFecha, idUsuarioTendero, idUsuarioCliente, idUsuarioRepartidor, idOrden)

VALUES (NOW(), NULL, p\_id\_usuario, NULL, p\_id\_orden); -- Adjust fields as necessary

-- Get the new idChat value

SET newChatId = LAST\_INSERT\_ID();

ELSE

-- If the chat exists, use the existing chat ID

SET newChatId = p\_id\_orden;

END IF;

-- Insert the message into the tblmensaje table using the new or existing chat ID

INSERT INTO tblmensaje (

menContenido,

menEstado,

menImagen,

menFechaEnvio,

idUsuario,

idChat

)

VALUES (

p\_mensaje,

'enviado', -- You can adjust this to the default status

p\_imagen,

NOW(), -- Automatically inserts the current timestamp

p\_id\_usuario,

newChatId -- Use the new or existing chat ID

);

**---------------------------------------------------------------------------------**

BEGIN

INSERT INTO tblListadoProductoTienda(

iptDescripcionProductoTienda,

iptStock,

lptStockMinimo,

lptPrecioCompra,

lptPrecioVenta,

lpEstado,

idProducto,

idTienda

) VALUES (

p\_lptDescripcionProductoTienda,

p\_lptStock,

p\_lptStockMinimo,

p\_lptPrecioCompra,

p\_lptPrecioVenta,

'Activo', -- Assuming a default status of 'Activo'

p\_idProducto,

p\_idTienda

);

END

**-----------------------------------------------------**

BEGIN

INSERT INTO tbllistadoproductotienda (

lptStock,

lptStockMinimo,

lptImagen1,

lptImagen2,

lptImagen3,

lptPrecioCompra,

lptPrecioVenta,

lpEstado,

idProducto,

idTienda

)

VALUES (

p\_lptStock,

p\_lptStockMinimo,

p\_lptImagen1,

p\_lptImagen2,

p\_lptImagen3,

p\_lptPrecioCompra,

p\_lptPrecioVenta,

1, -- Suponiendo que 'lpEstado' es 1 para productos activos

1,

p\_idTienda

);

END

**----------------------------------------------------**

BEGIN

-- Declarar variables para almacenar el idPersona y el idRepartidor generado

DECLARE v\_idPersona INT;

DECLARE v\_idRepartidor INT;

-- Buscar el idPersona en la tabla tblpersona utilizando el documento proporcionado

SELECT idPersona INTO v\_idPersona

FROM tblpersona

WHERE idPersona = p\_documento;

-- Insertar el nuevo repartidor en la tabla tblrepartidor

INSERT INTO tblrepartidor (

repEstado, repAntecedentes, repAntecedentesEstado,

repLicencia, repLicenciaEstado, idVehiculo, idDocumentoPersona

)

VALUES (

'activo', '', 'sin antecedentes',

'', 'sin licencia', NULL, v\_idPersona

);

-- Obtener el último idRepartidor insertado

SET v\_idRepartidor = LAST\_INSERT\_ID();

-- Devolver la información del repartidor junto con la imagen del usuario (de la tabla tblpersona)

SELECT CONCAT(perApellidos, ' ', perNombres) AS Nombre,

usuImagen,

v\_idRepartidor AS idRepartidor,

v\_idPersona AS idUsuario

FROM tblpersona

WHERE idPersona = v\_idPersona;

END

**----------------------------------------------------------**

**Rafael**

**c\_lista\_repartidor\_orden\_mis\_ordenes.php**

BEGIN

SELECT

r.idRepartidor,

CONCAT(p.perNombre, ' ', p.perApellidos) AS perNombreCompleto

FROM tblrepartidor r

JOIN tbltienda t ON t.idRepartidor = r.idRepartidor

JOIN tblpersona p ON r.idDocumentoPersona = p.idDocumentoPersona

WHERE t.idTienda = sp\_idTienda;

END

**-----------------------------------------------------------------------------------------------------------------------------------**

**c\_listar\_favoritos\_aprendizaje.php**

BEGIN

SELECT

p.perapEstado,

p.perapFechaInscripcion,

p.idAprendizaje,

p.idDocumentoPersona,

a.apreDescripcionRecurso,

a.apreContenido

FROM

tblpersonaaprendizaje p

INNER JOIN

tblaprendizaje a

ON

p.idAprendizaje = a.idAprendizaje

WHERE

p.idDocumentoPersona = p\_id\_DocumentoPersona;

END

**c\_listar\_ultimos\_favoritos\_aprendizaje.php**

BEGIN

SELECT

p.perapEstado,

p.perapFechaInscripcion,

p.idAprendizaje,

p.idDocumentoPersona,

a.apreDescripcionRecurso,

a.apreContenido

FROM tblpersonaaprendizaje p

INNER JOIN tblaprendizaje a ON p.idAprendizaje = a.idAprendizaje

WHERE p.idDocumentoPersona = p\_id\_DocumentoPersona

ORDER BY p.perapFechaInscripcion DESC;

END

**-----------------------------------------------------------------------------------------------------------------------------------**

**c\_mostrar\_detalle\_publicidad.php**

BEGIN

SELECT

p.pubImagen,

p.pubTitulo,

p.pubDescripcion,

p.pubFechaInicio,

p.pubCantVistas,

f.factpubMontoTotal,

DATEDIFF(CURDATE(), p.pubFechaInicio) AS fecha\_Dias, -- Calcula los días desde la fecha de inicio

f.tasa

FROM tblpublicidad p

LEFT JOIN tblfacturapublicidad f ON p.idPublicidad = f.idPublicidad

WHERE p.idTienda = p\_id\_Tienda

ORDER BY p.pubFechaInicio DESC;

END

**-----------------------------------------------------------------------------------------------------------------------------------**

**c\_productos\_globales\_mitienda.php**

BEGIN

SELECT

p.idProducto,

p.proImagen,

p.proDescripcion,

p.proPrecioCostoPromedio,

p.proPrecioVentaRecomendado,

p.proUnidadMedida,

c.cpNombre

FROM

tblproductos AS p

LEFT JOIN

tblcategoriaproducto AS c ON p.idCategoriaProducto = c.idCategoriaProducto

WHERE

(p.proDescripcion LIKE CONCAT('%', p\_proDescripcion, '%') OR p\_proDescripcion = ' ')

AND p.idTienda = p\_idTienda;

END

**-----------------------------------------------------------------------------------------------------------------------------------**

**c\_tiendas\_por\_persona\_registrotienda**

BEGIN

SELECT

t.idTienda,

t.tieNombre

FROM tbltienda t

-- Aquí deberías agregar un JOIN si necesitas relacionar tbltienda con otra tabla

-- que contenga el p\_idDocumentoPersona. Ejemplo:

-- JOIN tblpersona p ON t.idTienda = p.idTienda

-- WHERE p.idDocumentoPersona = p\_idDocumentoPersona

WHERE t.idDocumentoPersona = p\_idDocumentoPersona; -- Cambia esta línea si necesitas una relación real

END

**Adriann**

**sp\_c\_existencia\_persona\_inicio\_sesion**

DELIMITER $$

CREATE PROCEDURE sp\_c\_existencia\_persona\_inicio\_sesion(

IN sp\_idDocumentoPersona VARCHAR(20)

)

BEGIN

-- Verifica si la persona existe en la tabla tblpersona

SELECT

p.idDocumentoPersona, u.usuCorreo, u.usuImagen

FROM

tblpersona p

LEFT JOIN

tblusuario u ON p.idDocumentoPersona = u.idDocumentoPersona

WHERE

p.idDocumentoPersona = sp\_idDocumentoPersona;

END$$

DELIMITER ;

**sp\_c\_consultar\_lista\_producto**

DELIMITER $$

CREATE PROCEDURE sp\_c\_consultar\_lista\_producto(

IN p\_idTienda INT

)

BEGIN

-- Consulta para obtener la lista de productos por tienda

SELECT

p.proDescripcion,

lpt.lptStock

FROM

tbllistadoproductotienda lpt

INNER JOIN

tblproductos p ON lpt.idProducto = p.idProducto

WHERE

lpt.idTienda = p\_idTienda;

END$$

DELIMITER ;

**sp\_c\_consultar\_materiales\_por\_categoria\_aprendizaje**

DELIMITER $$

CREATE PROCEDURE sp\_c\_consultar\_materiales\_por\_categoria\_aprendizaje(

IN p\_nombre\_categoria VARCHAR(200), -- Nombre de la categoría

IN p\_idPersona INT -- ID de la persona

)

BEGIN

SELECT

a.idAprendizaje,

a.apreTituloRecurso, -- Ajusta el nombre si es necesario

a.apreContenido,

pa.perapLike -- Asegúrate de que el nombre es correcto

FROM

tblAprendizaje a

LEFT JOIN

tblPersonaAprendizaje pa ON a.idAprendizaje = pa.idAprendizaje AND pa.idPersona = p\_idPersona

INNER JOIN

tblCategoriaAprendizaje ca ON a.idCategoria = ca.idCategoria

WHERE

ca.catNombre = p\_nombre\_categoria; -- Filtrar por el nombre de la categoría

END$$

DELIMITER ;

**sp\_c\_consultar\_mensajes\_chat\_x\_orden**

DELIMITER $$

CREATE PROCEDURE sp\_c\_consultar\_mensajes\_chat\_x\_orden(

IN p\_id\_orden INT

)

BEGIN

SELECT

c.idChat,

m.idMensaje,

m.menContenido,

m.menEstado,

m.menImagen,

m.menFechaEnvio,

m.idUsuario,

u.usuImagen,

u.idUsuario,

r.ruNombre,

p.perNombres

FROM

tblchat c

INNER JOIN

tblmensaje m ON c.idChat = m.idChat

INNER JOIN

tblusuario u ON m.idUsuario = u.idUsuario

INNER JOIN

tblrolusuario r ON u.idRolUsuario = r.idRolUsuario

INNER JOIN

tblpersona p ON u.idDocumentoPersona = p.idDocumentoPersona

WHERE

c.idOrden = p\_id\_orden;

END$$

DELIMITER ;

**sp\_c\_consultar\_stock\_producto\_x\_nombre\_codigo\_almacen**

DELIMITER $$

CREATE PROCEDURE sp\_c\_consultar\_stock\_producto\_x\_nombre\_codigo\_almacen(

IN p\_idTienda INT,

IN p\_idProducto INT,

IN p\_proDescripcion VARCHAR(255),

IN xp\_modbusc INT

)

BEGIN

-- Si xp\_modbusc es 0, buscar por proDescripcion (nombre del producto)

IF xp\_modbusc = 0 THEN

SELECT

t.idTienda,

p.idProducto,

p.proDescripcion,

lpt.lptStock,

lpt.idListadoProductoTienda

FROM

tbllistadoproductotienda lpt

INNER JOIN

tblproductos p ON lpt.idProducto = p.idProducto

WHERE

lpt.idTienda = p\_idTienda

AND p.proDescripcion LIKE CONCAT('%', p\_proDescripcion, '%');

-- Si xp\_modbusc es 1, buscar por idProducto

ELSEIF xp\_modbusc = 1 THEN

SELECT

t.idTienda,

p.idProducto,

p.proDescripcion,

lpt.lptStock,

lpt.idListadoProductoTienda

FROM

tbllistadoproductotienda lpt

INNER JOIN

tblproductos p ON lpt.idProducto = p.idProducto

WHERE

lpt.idTienda = p\_idTienda

AND p.idProducto = p\_idProducto;

END IF;

END$$

DELIMITER ;

**sp\_c\_detalle\_cobro\_orden**

DELIMITER $$

CREATE PROCEDURE sp\_c\_detalle\_cobro\_orden(

IN p\_idOrden INT

)

BEGIN

SELECT

o.odGananciaTienda,

o.odGananciaRepartidor,

t.tieNombre

FROM

tblorden o

INNER JOIN

tbltienda t ON o.idTienda = t.idTienda

WHERE

o.idOrden = p\_idOrden;

END$$

DELIMITER ;

**sp\_c\_detalle\_orden\_x\_id\_mis\_ordenes**

DELIMITER $$

CREATE PROCEDURE sp\_c\_detalle\_orden\_x\_id\_mis\_ordenes(

IN p\_idOrden INT

)

BEGIN

SELECT

lpt.idListadoProductoTienda,

p.proDescripcion AS lptDescripcionProductoTienda,

do.doPrecioVenta,

lpt.lptImagen1,

p.proUnidadMedida AS lptUnidadMedida,

do.doCantidad,

do.doSubTotal

FROM

tbllistadoproductotienda lpt

INNER JOIN

tblproductos p ON lpt.idProducto = p.idProducto

INNER JOIN

tbldetalleorden do ON lpt.idListadoProductoTienda = do.idListadoProductoTienda

WHERE

do.idOrden = p\_idOrden;

END$$

DELIMITER ;

**sp\_c\_documentos\_repartidor**

DELIMITER $$

CREATE PROCEDURE sp\_c\_documentos\_repartidor(

IN p\_idDocumentoPersona VARCHAR(20)

)

BEGIN

SELECT

r.idDocumentoPersona,

r.repEstado,

v.vrTipoVehiculo AS repTipoVehiculo,

v.vrPlaca AS repPlaca,

r.repAntecedentes,

r.repAntecedentesEstado,

r.repLicencia,

r.repLicenciaEstado,

v.vrDocumentoVehiculo AS repDocumentoVehiculo,

v.vrDocumentoVehiculoEstado AS repDocumentoVehiculoEstado,

v.vrTarjetaPropiedad AS repTarjetaPropiedad,

v.vrTarjetaPropiedadEstado AS repTarjetaPropiedadEstado

FROM

tblrepartidor r

INNER JOIN

tblvehiculorepartidor v ON r.idVehiculo = v.idVehiculo

WHERE

r.idDocumentoPersona = p\_idDocumentoPersona;

END$$

DELIMITER ;

**sp\_c\_estado\_repartidor**

DELIMITER $$

CREATE PROCEDURE sp\_c\_estado\_repartidor(

IN p\_idDocumentoPersona VARCHAR(20)

)

BEGIN

SELECT

p.idDocumentoPersona,

p.perNombres,

p.perApellidos,

r.repEstado

FROM

tblrepartidor r

INNER JOIN

tbldocumentopersona p ON r.idDocumentoPersona = p.idDocumentoPersona

WHERE

r.idDocumentoPersona = p\_idDocumentoPersona;

END$$

DELIMITER ;

**sp\_c\_estado\_usuario\_inicio\_sesion**

DELIMITER $$

CREATE PROCEDURE sp\_c\_estado\_usuario\_inicio\_sesion(

IN p\_codvCorreo VARCHAR(255)

)

BEGIN

SELECT

u.codvCorreo,

u.codvCodigo

FROM

tblusuario u

WHERE

u.codvCorreo = p\_codvCorreo;

END$$

DELIMITER ;

**sp\_c\_existe\_repartidor**

DELIMITER $$

CREATE PROCEDURE sp\_c\_existe\_repartidor(

IN p\_documento VARCHAR(20)

)

BEGIN

DECLARE v\_existe INT DEFAULT 0;

-- Se verifica si el repartidor con el documento existe

SELECT COUNT(\*) INTO v\_existe

FROM tblrepartidor

WHERE idDocumentoPersona = p\_documento;

-- Devolver el resultado

SELECT v\_existe AS EXISTE;

END$$

DELIMITER ;

**sp\_c\_informacion\_por\_tienda\_registrotienda**

DELIMITER $$

CREATE PROCEDURE `sp\_c\_informacion\_por\_tienda\_registrotienda`(IN `p\_idTienda` INT)

BEGIN

SELECT

tieNombre,

tieImagen AS tieImagenurl,

tieURLWeb,

tieDescripcion,

tieCorreo,

tieTelefono,

tieDireccion,

tieCiudad,

tieEstado,

tieVentasMensuales,

tieInventarioEstimado,

tieLatitud,

tieLongitud,

idRubroTienda

FROM

tbltienda

WHERE

idTienda = p\_idTienda;

END$$

DELIMITER ;

**sp\_c\_lista\_estados\_orden\_mis\_ordenes**

DELIMITER $$

CREATE PROCEDURE `sp\_c\_lista\_estados\_orden\_mis\_ordenes`(IN `p\_idOrden` INT)

BEGIN

SELECT

he.esorEstado AS esorEstado,

o.odFechaPedido AS esorFecha,

he.esorHora AS esorHora

FROM

tblhistorialestadoorden he

INNER JOIN

tblorden o ON he.idOrden = o.idOrden

WHERE

he.idOrden = p\_idOrden;

END$$

DELIMITER ;

**sp\_c\_lista\_ordenes\_mis\_ordenes**

DELIMITER $$

CREATE PROCEDURE `sp\_c\_lista\_ordenes\_mis\_ordenes`(

IN `p\_idTienda` INT,

IN `p\_idOrden` INT,

IN `p\_odEstado` VARCHAR(50),

IN `p\_odFechaPedido` DATE

)

BEGIN

SELECT

o.idOrden,

o.odFechaPedido,

o.odHoraPedido,

CONCAT(p.perNombre, ' ', p.perApellidos) AS perNombreCompleto,

o.odEstado,

o.idRepartidor,

r.repNombres AS repNombres

FROM

tblorden o

INNER JOIN

tblpersona p ON o.idPersona = p.idPersona

LEFT JOIN

tblrepartidor r ON o.idRepartidor = r.idRepartidor

WHERE

o.idTienda = p\_idTienda

AND (o.idOrden = p\_idOrden OR p\_idOrden IS NULL)

AND (o.odEstado = p\_odEstado OR p\_odEstado IS NULL)

AND (o.odFechaPedido = p\_odFechaPedido OR p\_odFechaPedido IS NULL);

END$$

DELIMITER ;

**sp\_c\_lista\_ordenes\_por\_idrepartidor**

DELIMITER $$

CREATE PROCEDURE `sp\_c\_lista\_ordenes\_por\_idrepartidor`(IN `p\_idRepartidor` INT)

BEGIN

SELECT

o.idOrden,

o.odFechaPedido,

o.odHoraPedido,

o.odEstado,

o.odLatitudEntrega,

o.odLongitudEntrega,

o.odFechaEntrega,

o.odHoraEntrega,

t.tieNombre AS NombreTienda,

CONCAT(p.perNombre, ' ', p.perApellidos) AS Cliente

FROM

tblorden o

INNER JOIN

tbltienda t ON o.idTienda = t.idTienda

INNER JOIN

tblpersona p ON o.idPersona = p.idPersona

WHERE

o.idRepartidor = p\_idRepartidor;

END$$

DELIMITER ;