**LOGIN**

PROCEDIMIENTO PARA Verificar si el usuario exste

-----------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_verificarUsuario(IN \_usuario varchar(20),

IN \_contraseña varchar(20))

BEGIN

SELECT usuario.idUsuario, empleado.codEmpleado, usuario.estado, concat(empleado.nombres, ' ' ,empleado.apellidos) as empleado

from usuario INNER JOIN empleado USING(codEmpleado)

WHERE userName = \_usuario AND password = \_contraseña AND usuario.estado = 1;

END;

$$

**AUDILOG**

Procedimiento para mostrar el contenido de audiLog

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_mostrarAudiLog()

BEGIN

SELECT audilog.idAudiLog, audiLog.fechaEvento, usuario.userName, tipotransaccion.tipoTransaccion, audiLog.descripcion FROM audilog INNER JOIN usuario USING (idUsuario) INNER JOIN tipotransaccion USING(idTipoTransaccion);

END;

$$

------------------------------Solicitud matenimiento--------------------------------------------

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_insertarSolicitudMantenimiento

AFTER INSERT ON solicitudmantenimiento

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = new.idUsuario;

SET idRegistro = new.idSolicitudMantenimiento;

SET descrip = CONCAT('Se insertó el registro en solicitudmantenimiento con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip, 1);

END$$

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_actualizarSolicitudMantenimiento

AFTER UPDATE ON solicitudmantenimiento

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = new.idUsuario;

SET idRegistro = old.idSolicitudMantenimiento;

SET descrip = CONCAT('Se actualizó el registro en solicitudmantenimiento con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip , 2);

END$$

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_eliminarSolicitudMantenimiento

AFTER DELETE ON solicitudmantenimiento

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = old.idUsuario;

SET idRegistro = old.idSolicitudMantenimiento;

SET descrip = CONCAT('Se eliminó el registro en solicitudmantenimiento con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip, 3);

END$$

------------------------------control matenimiento--------------------------------------------

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_insertarControlMantenimiento

AFTER INSERT ON controlmantenimiento

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = new.idUsuario;

SET idRegistro = new.idControlMantenimiento;

SET descrip = CONCAT('Se insertó el registro en controlmantenimiento con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip, 1);

END$$

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_actualizarControlMantenimiento

AFTER UPDATE ON controlmantenimiento

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = new.idUsuario;

SET idRegistro = old.idControlMantenimiento;

SET descrip = CONCAT('Se actualizó el registro en controlmantenimiento con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip , 2);

END$$

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_eliminarControlMantenimiento

AFTER DELETE ON controlmantenimiento

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = old.idUsuario;

SET idRegistro = old.idControlMantenimiento;

SET descrip = CONCAT('Se eliminó el registro en controlmantenimiento con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip, 3);

END$$

------------------------------asignación equipo--------------------------------------------

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_insertarAsignacionEquipo

AFTER INSERT ON asignacionequipo

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = new.idUsuario;

SET idRegistro = new.idAsignacionEquipo;

SET descrip = CONCAT('Se insertó el registro en asignacionequipo con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip, 1);

END$$

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_actualizarAsignacionEquipo

AFTER UPDATE ON asignacionequipo

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = new.idUsuario;

SET idRegistro = old.idAsignacionEquipo;

SET descrip = CONCAT('Se actualizó el registro en asignacionequipo con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip , 2);

END$$

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_eliminarAsignacionEquipo

AFTER DELETE ON asignacionequipo

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = old.idUsuario;

SET idRegistro = old.idAsignacionEquipo;

SET descrip = CONCAT('Se eliminó el registro en asignacionequipo con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip, 3);

END$$

------------------------------control garantia--------------------------------------------

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_insertarControlGarantia

AFTER INSERT ON controlgarantia

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = new.idUsuario;

SET idRegistro = new.idControlGarantia;

SET descrip = CONCAT('Se insertó el registro en controlgarantia con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip, 1);

END$$

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_actualizarControlGarantia

AFTER UPDATE ON controlgarantia

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = new.idUsuario;

SET idRegistro = old.idControlGarantia;

SET descrip = CONCAT('Se actualizó el registro en controlgarantia con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip , 2);

END$$

DELIMITER $$

CREATE OR REPLACE TRIGGER tx\_eliminarControlGarantia

AFTER DELETE ON controlgarantia

FOR EACH ROW

BEGIN

DECLARE idUser INTEGER;

DECLARE idRegistro INTEGER;

DECLARE descrip VARCHAR(100);

SET idUser = old.idUsuario;

SET idRegistro = old.idControlGarantia;

SET descrip = CONCAT('Se eliminó el registro en controlgarantia con Id', ' ', idRegistro);

INSERT INTO audilog(fechaEvento, idUsuario, descripcion, idTipoTransaccion)

VALUES (NOW(), idUser, descrip, 3);

END$$

**CARGO**

PROCEDIMIENTO PARA MOSTRAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_mostrarCargo()

BEGIN

SELECT cargo.idCargo, cargo.cargo, departamento.departamento FROM cargo INNER JOIN departamento USING (idDepartamento);

END;

$$

PROCEDIMIENTO PARA INSERTAR

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

DELIMITER $$

CREATE PROCEDURE sp\_insertarCargo(IN cargo varchar(30),

IN idDepartamento int)

BEGIN

INSERT INTO cargo (cargo, idDepartamento ) VALUES(cargo, idDepartamento );

END;

$$

PROCEDIMIENTO PARA ACTUALIZAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarCargo(IN \_cargo varchar(30),

IN \_idDepartamento INT,

IN \_idCargo int)

BEGIN

UPDATE cargo

SET cargo = \_cargo,

idDepartamento = \_idDepartamento

WHERE idCargo = \_idCargo;

END;

$$

PROCEDIMIENTO PARA ELIMINAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarCargo(IN \_idCargo int)

BEGIN

DELETE FROM cargo WHERE idCargo=\_idCargo;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeCargo(IN \_cargo varchar(30))

BEGIN

SELECT count(\*) from cargo

WHERE cargo = \_cargo;

END;

$$

**CAPACIDAD**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarCapacidad()

BEGIN

SELECT \* FROM capacidad;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_insertarCapacidad(IN \_capacidad varchar(20))

BEGIN

INSERT INTO capacidad (capacidad) VALUES(\_capacidad);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarCapacidad(IN \_idCapacidad INT,

IN \_capacidad varchar(20))

BEGIN

UPDATE capacidad

SET capacidad = \_capacidad

WHERE idCapacidad = \_idCapacidad;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarCapacidad(IN \_idCapacidad int)

BEGIN

DELETE FROM capacidad WHERE idCapacidad = \_idCapacidad;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeCapacidad(IN \_capacidad varchar(20))

BEGIN

SELECT count(\*) from capacidad

WHERE capacidad = \_capacidad;

END;

$$

**DEPARTAMENTO**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarDepartamento()

BEGIN

SELECT \* FROM departamento;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_insertarDepartamento(IN \_departamento varchar(40))

BEGIN

INSERT INTO departamento (departamento) VALUES(\_departamento);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarDepartamento(IN \_idDepartamento INT,

IN \_departamento varchar(40))

BEGIN

UPDATE departamento

SET departamento = \_departamento

WHERE idDepartamento = \_idDepartamento;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarDepartamento(IN \_idDepartamento int)

BEGIN

DELETE FROM departamento WHERE idDepartamento = \_idDepartamento;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeDepartamento(IN \_departamento varchar(40))

BEGIN

SELECT count(\*) from departamento

WHERE departamento = \_departamento;

END;

$$

**ESTADO CONTROL MANTENIMIENTO**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarEstadoControlMantenimiento()

BEGIN

SELECT \* FROM estadoControlMantenimiento;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_insertarEstadoControlMantenimiento(IN \_estadoControlMantenimiento varchar(50))

BEGIN

INSERT INTO estadoControlMantenimiento (estadoControlMantenimiento) VALUES(\_estadoControlMantenimiento);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarEstadoControlMantenimiento(IN \_id INT,

IN \_estadoControlMantenimiento varchar(50))

BEGIN

UPDATE estadoControlMantenimiento

SET estadoControlMantenimiento = \_estadoControlMantenimiento

WHERE idEstadoControlMantenimiento = \_id;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarEstadoControlMantenimiento(IN \_id int)

BEGIN

DELETE FROM estadoControlMantenimiento WHERE idEstadoControlMantenimiento = \_id;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeEstadoControlMantenimiento(IN \_estadoControlMantenimiento varchar(50))

BEGIN

SELECT count(\*) from estadocontrolmantenimiento

WHERE estadoControlMantenimiento = \_estadoControlMantenimiento;

END;

$$

**MARCA**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarMarca()

BEGIN

SELECT \* FROM marca;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_insertarMarca(IN \_marca varchar(20))

BEGIN

INSERT INTO marca (marca) VALUES(\_marca);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarMarca(IN \_idMarca INT,

IN \_marca varchar(20))

BEGIN

UPDATE marca

SET marca = \_marca

WHERE idMarca = \_idMarca;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarMarca(IN \_idMarca int)

BEGIN

DELETE FROM marca WHERE idMarca = \_idMarca;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeMarca(IN \_marca varchar(20))

BEGIN

SELECT count(\*) from marca

WHERE marca = \_marca;

END;

$$

**EMPRESA GARANTIA**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarEmpresaGarantia()

BEGIN

SELECT \* FROM empresagarantia;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_insertarEmpresaGarantia(IN \_nombreEmpresa varchar(50))

BEGIN

INSERT INTO empresagarantia (nombreEmpresa) VALUES(\_nombreEmpresa);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_actualizarEmpresaGarantia(IN \_idEmpresa int,

IN \_nombreEmpresa varchar(50))

BEGIN

UPDATE empresagarantia

SET nombreEmpresa = \_nombreEmpresa

WHERE idEmpresa = \_idEmpresa;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_eliminarEmpresaGarantia (IN \_idEmpresa int)

BEGIN

DELETE FROM empresagarantia

WHERE idEmpresa = \_idEmpresa;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeEmpresaGarantia(IN \_nombreEmpresa varchar(50))

BEGIN

SELECT count(\*) from empresagarantia

WHERE nombreEmpresa = \_nombreEmpresa;

END;

$$

**TIPO DISCO**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarTipoDisco()

BEGIN

SELECT \* FROM tipoDisco;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_insertarTipoDisco(IN \_tipoDisco varchar(10))

BEGIN

INSERT INTO tipoDisco (tipoDisco) VALUES(\_tipoDisco);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarTipoDisco(IN \_idTipoDisco INT,

IN \_TipoDisco varchar(10))

BEGIN

UPDATE tipoDisco

SET tipoDisco = \_TipoDisco

WHERE idTipoDisco = \_idTipoDisco;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarTipoDisco(IN \_idTipoDisco int)

BEGIN

DELETE FROM tipoDisco WHERE idTipoDisco = \_idTipoDisco;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeTipoDisco(IN \_tipoDisco varchar(10))

BEGIN

SELECT count(\*) from tipodisco

WHERE tipoDisco = \_tipoDisco;

END;

$$

**TIPO EQUIPO**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarTipoEquipo()

BEGIN

SELECT \* FROM tipoEquipo;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_insertarTipoEquipo(IN \_tipoEquipo varchar(30))

BEGIN

INSERT INTO tipoEquipo (tipoEquipo) VALUES(\_tipoEquipo);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarTipoEquipo(IN \_idTipoEquipo INT,

IN \_TipoEquipo varchar(30))

BEGIN

UPDATE tipoEquipo

SET tipoEquipo = \_TipoEquipo

WHERE idTipoEquipo = \_idTipoEquipo;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarTipoEquipo(IN \_idTipoEquipo int)

BEGIN

DELETE FROM tipoEquipo WHERE idTipoEquipo = \_idTipoEquipo;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeTipoEquipo(IN \_tipoEquipo varchar(30))

BEGIN

SELECT count(\*) from tipoequipo

WHERE tipoEquipo = \_tipoEquipo;

END;

$$

**TIPO MANTENIMIENTO**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarTipoMantenimiento()

BEGIN

SELECT \* FROM tipoMantenimiento;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_insertarTipoMantenimiento(IN \_tipoMantenimiento varchar(10))

BEGIN

INSERT INTO tipoMantenimiento (tipoMantenimiento) VALUES(\_tipoMantenimiento);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarTipoMantenimiento(IN \_idTipoMantenimiento INT,

IN \_TipoMantenimiento varchar(10))

BEGIN

UPDATE tipoMantenimiento

SET tipoMantenimiento = \_TipoMantenimiento

WHERE idTipoMantenimiento = \_idTipoMantenimiento;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarTipoMantenimiento(IN \_idTipoMantenimiento int)

BEGIN

DELETE FROM tipoMantenimiento WHERE idTipoMantenimiento = \_idTipoMantenimiento;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeTipoMantenimiento(IN \_tipoMantenimiento varchar(10))

BEGIN

SELECT count(\*) from tipomantenimiento

WHERE tipoMantenimiento = \_tipoMantenimiento;

END;

$$

**TIPO RAM**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarTipoRam()

BEGIN

SELECT \* FROM tiporam;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_insertarTipoRam(IN \_tipoRam varchar(20))

BEGIN

INSERT INTO tipoRam (tipoRam) VALUES(\_tipoRam);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarTipoRam(IN \_idTipoRam INT,

IN \_tipoRam varchar(20))

BEGIN

UPDATE tipoRam

SET tipoRam = \_tipoRam

WHERE idTipoRam = \_idTipoRam;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarTipoRam(IN \_idTipoRam int)

BEGIN

DELETE FROM tiporam WHERE idTipoRam = \_idTipoRam;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeTipoRam(IN \_tipoRam varchar(20))

BEGIN

SELECT count(\*) from tiporam

WHERE tipoRam = \_tipoRam;

END;

$$

**VELOCIDAD RAM**

DELIMITER $$

CREATE PROCEDURE sp\_mostrarVelocidadRam()

BEGIN

SELECT \* FROM velocidadram;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_insertarVelocidadRam(IN \_velocidadRam varchar(20))

BEGIN

INSERT INTO velocidadram (velocidadRam) VALUES(\_velocidadRam);

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarVelocidadRam(IN \_idVelocidadRam INT,

IN \_velocidadRam varchar(20))

BEGIN

UPDATE velocidadram

SET velocidadRam = \_velocidadRam

WHERE idVelocidadRam = \_idVelocidadRam;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarVelocidadRam(IN \_idVelocidadRam int)

BEGIN

DELETE FROM velocidadram WHERE idVelocidadRam = \_idVelocidadRam;

END;

$$

---------------------------------------------------------------------------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_existeVelocidadRam(IN \_velocidadRam varchar(20))

BEGIN

SELECT count(\*) from velocidadram

WHERE velocidadRam = \_velocidadRam;

END;

$$

**MODELO**

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_mostrarModelo()

BEGIN

SELECT modelo.idModelo, modelo.modelo, marca.marca FROM modelo INNER JOIN marca USING (idMarca);

END;

$$

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

DELIMITER $$

CREATE PROCEDURE sp\_insertarModelo(IN \_modelo varchar(20),

IN \_idMarca int)

BEGIN

INSERT INTO modelo (modelo, idMarca ) VALUES(\_modelo, \_idMarca);

END;

$$

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarModelo(IN \_modelo varchar(20),

IN \_idMarca INT,

IN \_idModelo int)

BEGIN

UPDATE modelo SET modelo = \_modelo, idMarca = \_idMarca

WHERE idModelo = \_idModelo;

END;

$$

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarModelo(IN \_idModelo int)

BEGIN

DELETE FROM modelo WHERE idModelo=\_idModelo;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeModelo(IN \_modelo varchar(20))

BEGIN

SELECT count(\*) from modelo

WHERE modelo = \_modelo;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarMarca()

BEGIN

SELECT \* FROM marca;

END;

$$

**USUARIO**

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_mostrarUsuario()

BEGIN

SELECT usuario.idUsuario, concat(empleado.nombres, ' ', empleado.apellidos) as empleado, usuario.fechaCreacion, usuario.userName, usuario.password, usuario.fechaBaja, CASE usuario.estado WHEN 0 THEN 'INACTIVO' WHEN 1 THEN 'ACTIVO' END AS estado

FROM usuario INNER JOIN empleado USING (codEmpleado);END;

$$

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

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_insertarUsuario(IN \_codEmpleado int(10),

IN \_fechaC datetime, IN \_userName varchar(20), IN \_password varchar(20) , IN \_fechaB datetime )

BEGIN

INSERT INTO usuario (codEmpleado, fechaCreacion, userName, password, fechaBaja) VALUES(\_codEmpleado, \_fechaC, \_userName, \_password, \_fechaB);

END;

$$

-----------------------------------------

call sp\_actualizarUsuario(77, 'jamz', 'JAMZ', null, 0);

DELIMITER $$

CREATE PROCEDURE sp\_actualizarUsuario(IN \_idUsuario int ,

IN \_userName varchar(20),

IN \_password varchar(20),

IN \_fechaB date,

IN \_estado Tinyint)

BEGIN

UPDATE usuario SET userName = \_userName,

password = \_password,

fechaBaja= \_fechaB,

estado= \_estado

WHERE idUsuario = \_idUsuario;

END;

$$

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarUsuario(IN \_idUsuario int)

BEGIN

DELETE FROM usuario WHERE idUsuario=\_idUsuario;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeUsuario(IN \_userName varchar(20))

BEGIN

SELECT count(\*) as cantidad from usuario

WHERE userName = \_userName;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarEmpleado1()

BEGIN

SELECT empleado.codEmpleado, concat(empleado.nombres, ' ', empleado.apellidos) as empleado, empleado.telefono, empleado.celular, sexo.sexo, cargo.cargo, empleado.estado FROM empleado INNER JOIN cargo USING (idCargo) INNER JOIN sexo USING (idSexo);

END;

$$

**SOLICITUD MANTENIMIENTO**

PROCEDIMIENTO PARA MOSTRAR

-----------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_mostrarSolicitudM(IN \_idUsuario int)

BEGIN

SELECT solicitudmantenimiento.idSolicitudMantenimiento, solicitudmantenimiento.fechaSolicitudMantenimiento, equipo.codEquipo, equipo.descripcionEquipo, tipoequipo.tipoEquipo, modelo.modelo, equipo.serviceTag, solicitudMantenimiento.preDiagnostico, usuario.userName

FROM solicitudmantenimiento INNER JOIN equipo

ON solicitudmantenimiento.codEquipo = equipo.codEquipo

INNER JOIN tipoequipo

ON equipo.idTipoEquipo = tipoequipo.idTipoEquipo

INNER JOIN modelo

ON equipo.idModelo = modelo.idModelo

INNER JOIN usuario

ON solicitudmantenimiento.idUsuario = usuario.idUsuario

WHERE solicitudmantenimiento.idUsuario = \_idUsuario;

END;

$$

PROCEDIMIENTO PARA INSERTAR

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

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_insertarSolicitudM(IN \_fechaSolicitudMantenimiento date,

IN \_codEquipo varchar(9), IN \_preDiagnostico varchar(100), IN \_idUsuario int)

BEGIN

INSERT INTO solicitudmantenimiento (fechaSolicitudMantenimiento, codEquipo, preDiagnostico, idUsuario ) VALUES(\_fechaSolicitudMantenimiento, \_codEquipo, \_preDiagnostico, \_idUsuario );

END;

$$

PROCEDIMIENTO PARA ACTUALIZAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarSolicitudM(IN \_fechaSM date,

IN \_codEquipo varchar(9),

IN \_preDiagnostico varchar(100),

IN \_idSM int)

BEGIN

UPDATE solicitudmantenimiento

SET fechaSolicitudMantenimiento = \_fechaSM,

codEquipo = \_codEquipo,

preDiagnostico = \_preDiagnostico

WHERE idSolicitudMantenimiento = \_idSM;

END;

$$

PROCEDIMIENTO PARA ELIMINAR -----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarSolicitudM(IN \_idSolicitud int)

BEGIN

DELETE FROM solicitudMantenimiento WHERE idSolicitudMantenimiento = \_idSolicitud;

END;

$$

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_obtenerCorreoEmpleado(IN \_codEmpleado varchar(10))

BEGIN

SELECT CONCAT(empleado.nombres, ' ', empleado.apellidos) as NombreEmpleado,empleado.correoElectronico

from usuario INNER JOIN solicitudmantenimiento USING(idUsuario)

INNER JOIN empleado USING(codEmpleado)

WHERE codEmpleado = \_codEmpleado;

END

$$

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_mostrarEquipoSolicitud(IN \_idUsuario int)

BEGIN

SELECT equipo.codEquipo, tipoequipo.tipoEquipo, equipo.descripcionEquipo,

equipo.serie, equipo.serviceTag, modelo.modelo

FROM equipo INNER JOIN tipoequipo

USING (idTipoEquipo)

INNER JOIN modelo

USING (idModelo)

INNER JOIN asignacionequipo

USING (codEquipo)

WHERE asignacionequipo.idUsuario = \_idUsuario;

END;

$$

**EMPLEADO**

PROCEDIMIENTO PARA MOSTRAR

-----------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_mostrarEmpleado()

BEGIN

SELECT empleado.codEmpleado, empleado.nombres, empleado.apellidos, empleado.telefono, empleado.celular, empleado.extension, sexo.sexo, cargo.cargo, empleado.correoElectronico, CASE empleado.estado WHEN 0 THEN 'INACTIVO' WHEN 1 THEN 'ACTIVO' END AS estado

FROM empleado INNER JOIN cargo USING (idCargo) INNER JOIN sexo USING (idSexo);

END;

$$

PROCEDIMIENTO PARA INSERTAR

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

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_insertarEmpleado(IN \_codEmpleado int(10), IN \_nombres varchar(50) ,IN \_apellidos varchar(50), IN \_telefono varchar(9), IN \_celular varchar(9), IN \_extension varchar(9), IN \_select\_Sexo int, IN \_select\_Cargo int, IN \_eMail varchar(40))

BEGIN

INSERT INTO empleado (codEmpleado, nombres, apellidos, telefono, celular, extension, idSexo, idCargo, correoElectronico) VALUES(\_codEmpleado, \_nombres, \_apellidos, \_telefono, \_celular, \_extension, \_select\_Sexo, \_select\_Cargo, \_eMail);

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarSexo()

BEGIN

SELECT idSexo, sexo FROM sexo ORDER BY sexo ASC;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarCargo()

BEGIN

SELECT cargo.idCargo, cargo.cargo, departamento.departamento FROM cargo INNER JOIN departamento USING (idDepartamento);

END;

$$

PROCEDIMIENTO PARA ACTUALIZAR

----------------------------------------- call sp\_actualizarEmpleado("0203", "Rom", "Zambra", "27804050", null, 1, 1, "ronalMendoza@hotmail.es", 1)

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_actualizarEmpleado(IN \_codEmpleado int(10),

IN \_nombres varchar(50),

IN \_apellidos varchar(50),

IN \_telefono varchar(9),

IN \_celular varchar(9),

IN \_extension varchar(9),

IN \_idSexo int,

IN \_idCargo int,

IN \_eMail varchar(40),

IN \_estado tinyint)

BEGIN

UPDATE empleado

SET nombres = \_nombres,

apellidos = \_apellidos,

telefono = \_telefono,

celular = \_celular,

extension = \_extension,

idSexo = \_idSexo,

idCargo = \_idCargo,

correoElectronico = \_eMail,

estado = \_estado

WHERE codEmpleado = \_codEmpleado;

END;

$$

PROCEDIMIENTO PARA ELIMINAR

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_eliminarEmpleado(IN \_codEmpleado int(10))

BEGIN

DELETE FROM empleado WHERE codEmpleado = \_codEmpleado;

END;

$$

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_existeEmpleado(IN \_codEmpleado int(10))

BEGIN

SELECT count(\*) from empleado

WHERE codEmpleado = \_codEmpleado;

END;

$$

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_existeEmpleadoUsuario(IN \_codEmpleado int(10))

BEGIN

SELECT count(\*) from usuario

WHERE codEmpleado = \_codEmpleado;

END;

$$

**CONTROL MANTENIMIENTO**

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_mostrarControlMantenimiento()

BEGIN

SELECT controlmantenimiento.idControlMantenimiento, solicitudmantenimiento.idSolicitudMantenimiento, controlmantenimiento.fechaControlMantenimiento, equipo.codEquipo, equipo.descripcionEquipo, tipomantenimiento.tipoMantenimiento,

solicitudmantenimiento.fechaSolicitudMantenimiento,controlmantenimiento.observacion, estadocontrolmantenimiento.estadoControlMantenimiento, concat(empleado.nombres, ' ', empleado.apellidos) as empleado

FROM controlmantenimiento INNER JOIN solicitudmantenimiento

ON controlmantenimiento.idSolicitudMantenimiento = solicitudmantenimiento.idSolicitudMantenimiento

INNER JOIN equipo

ON solicitudmantenimiento.codEquipo = equipo.codEquipo

INNER JOIN tipomantenimiento

ON controlmantenimiento.idTipoMantenimiento = tipomantenimiento.idTipoMantenimiento

INNER JOIN usuario

ON controlmantenimiento.idUsuario = usuario.idUsuario

INNER JOIN empleado

ON usuario.codEmpleado = empleado.codEmpleado

INNER JOIN estadocontrolmantenimiento

ON controlmantenimiento.idEstadoControlMantenimiento = estadocontrolmantenimiento.idEstadoControlMantenimiento;

END;

$$

PROCEDIMIENTO PARA INSERTAR

**CALL sp\_insertarControlMantenimiento (2021-02-30, 2031, 1, 6, "SQL", 1)**

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_insertarControlMantenimiento (IN \_fechaControlMant date, IN \_usuario int(11) , IN \_tipoMant int, IN \_idSolMant int, IN \_observacion varchar(100), IN \_idEstadoControlMant int)

BEGIN

INSERT INTO controlmantenimiento (fechaControlMantenimiento, idUsuario, idTipoMantenimiento, idSolicitudMantenimiento, observacion, idEstadoControlMantenimiento)

VALUES( \_fechaControlMant, \_usuario, \_tipoMant, \_idSolMant, \_observacion, \_idEstadoControlMant);

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarEmpleado()

BEGIN

SELECT empleado.codEmpleado, concat(empleado.nombres, ' ', empleado.apellidos) as empleado, empleado.telefono, empleado.celular, sexo.sexo, cargo.cargo, empleado.estado FROM empleado INNER JOIN cargo USING (idCargo) INNER JOIN sexo USING (idSexo);

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarTipoMantenimiento()

BEGIN

SELECT \* FROM tipoMantenimiento;

END;

$$

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_mostrarSolicitudMControl()

BEGIN

SELECT solicitudmantenimiento.idSolicitudMantenimiento, solicitudmantenimiento.fechaSolicitudMantenimiento, equipo.descripcionEquipo, solicitudMantenimiento.preDiagnostico, concat(empleado.nombres, ' ', empleado.apellidos) as empleado

FROM solicitudmantenimiento INNER JOIN equipo USING (codEquipo)

INNER JOIN usuario USING (idUsuario)

INNER JOIN empleado USING(codEmpleado);

END;

$$

PROCEDIMIENTO PARA ACTUALIZAR

-----------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_actualizarControlMantenimiento (IN \_idControlMant int,

IN \_fechaControlMant date,

IN \_usuario int(11),

IN \_tipoMant int,

IN \_idSolMant int,

IN \_observacion varchar(100),

IN \_estadoControlMant int)

BEGIN

UPDATE controlmantenimiento

SET fechaControlMantenimiento = \_fechaControlMant,

idUsuario = \_usuario,

idTipoMantenimiento = \_tipoMant,

idSolicitudMantenimiento = \_idSolMant,

observacion = \_observacion,

idEstadoControlMantenimiento = \_estadoControlMant

WHERE idControlMantenimiento = \_idControlMant;

END;

$$

PROCEDIMIENTO PARA ELIMINAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarControlMantenimiento (IN \_idControlMant int)

BEGIN

DELETE FROM controlmantenimiento WHERE idControlMantenimiento = \_idControlMant;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeIdSolMant (IN \_idSolMant int)

BEGIN

SELECT count(\*)

from controlmantenimiento

WHERE idSolicitudMantenimiento = \_idSolMant;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarUsuario()

BEGIN

SELECT usuario.idUsuario, concat(empleado.nombres, ' ', empleado.apellidos) as empleado, usuario.fechaCreacion, usuario.userName, usuario.password, usuario.fechaBaja, CASE usuario.estado WHEN 0 THEN 'INACTIVO' WHEN 1 THEN 'ACTIVO' END AS estado

FROM usuario INNER JOIN empleado USING (codEmpleado);

END;

$$

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_obtenerCorreoEmpleadoControlMant(IN \_codEquipo varchar(9))

BEGIN

SELECT CONCAT(empleado.nombres, ' ', empleado.apellidos) as NombreEmpleado,empleado.correoElectronico

from usuario INNER JOIN asignacionequipo USING(idUsuario)

INNER JOIN empleado USING(codEmpleado)

WHERE codEquipo = \_codEquipo;

END

$$

**BITACORA MANTENIMIENTO**

PROCEDIMIENTO PARA MOSTRAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_mostrarBitacoraMantenimiento()

BEGIN

SELECT bitacoramantenimiento.idBitacora, bitacoramantenimiento.fechaCambioEstado, solicitudmantenimiento.fechaSolicitudMantenimiento, estadocontrolmantenimiento.estadoControlMantenimiento FROM bitacoramantenimiento INNER JOIN estadocontrolmantenimiento USING (idEstadoControlMantenimiento) INNER JOIN solicitudmantenimiento USING (idSolicitudMantenimiento);

END;

$$

PROCEDIMIENTO PARA INSERTAR

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

DELIMITER $$

CREATE PROCEDURE sp\_insertarBitacoraMantenimiento (IN \_fechaCambioEstado date, IN \_solicitudMant int , IN \_estadoControlMant int)

BEGIN

INSERT INTO bitacoramantenimiento (fechaCambioEstado, idSolicitudMantenimiento, idEstadoControlMantenimiento) VALUES( \_fechaCambioEstado, \_solicitudMant, \_estadoControlMant);

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarSolicitudM()

BEGIN

SELECT solicitudmantenimiento.idSolicitudMantenimiento, solicitudmantenimiento.fechaSolicitudMantenimiento, equipo.descripcionEquipo, solicitudMantenimiento.preDiagnostico FROM solicitudmantenimiento INNER JOIN equipo USING (codEquipo);

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarEstadoControlMant()

BEGIN

SELECT idEstadoControlMantenimiento, estadoControlMantenimiento FROM estadocontrolmantenimiento;

END;

$$

PROCEDIMIENTO PARA ACTUALIZAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarBitacoraMantenimiento (IN \_idBitacora int,

IN \_fechaCambioEstado date,

IN \_idSolMant int,

IN \_idEstadoCMant int)

BEGIN

UPDATE bitacoramantenimiento

SET fechaCambioEstado = \_fechaCambioEstado,

idSolicitudMantenimiento = \_idSolMant,

idEstadoControlMantenimiento = \_idEstadoCMant

WHERE idBitacora = \_idBitacora;

END;

$$

PROCEDIMIENTO PARA ELIMINAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarBitacoraMantenimiento (IN \_idBitacoraMant int)

BEGIN

DELETE FROM bitacoramantenimiento WHERE idBitacora = \_idBitacoraMant;

END;

$$

**EQUIPO**

-----------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_mostrarEquipo1()

BEGIN

SELECT equipo.codEquipo, equipo.numInventario, tipoequipo.tipoEquipo, modelo.modelo, equipo.serie, equipo.serviceTag, equipo.descripcionEquipo

FROM equipo INNER JOIN tipoequipo USING (idTipoEquipo)

INNER JOIN modelo USING(idModelo) ;

END;

$$

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

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_insertarEquipo (IN \_codEquipo varchar(9), IN \_numInventario varchar(8),

IN \_idTipoEquipo int, IN \_idModelo int, IN \_serie varchar(20) , IN \_serviceTag varchar(30), IN \_descripcionEquipo varchar(50) )

BEGIN

INSERT INTO equipo (codEquipo, numInventario, idTipoEquipo, idModelo, serie, serviceTag, descripcionEquipo) VALUES(\_codEquipo, \_numInventario, \_idTipoEquipo, \_idModelo, \_serie, \_serviceTag, \_descripcionEquipo);

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarTipoEquipo()

BEGIN

SELECT \* FROM tipoEquipo;

END;

$$

call sp\_mostrarModeloSelect(2);//2 = dell

DELIMITER $$

CREATE PROCEDURE sp\_mostrarModeloSelect(IN \_marca int )

BEGIN

SELECT modelo.idModelo, modelo.modelo, marca.marca FROM modelo INNER JOIN marca USING (idMarca)

WHERE modelo.idMarca = \_marca;

END;

$$

-----------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_actualizarEquipo (IN \_codEquipo varchar(9), IN \_numInventario varchar(8),

IN \_idTipoEquipo int, IN \_idModelo int, IN \_serie varchar(20) , IN \_serviceTag varchar(30), IN \_descripcionEquipo varchar(50) )

BEGIN

UPDATE equipo SET numInventario = \_numInventario,

idTipoEquipo = \_idTipoEquipo,

idModelo = \_idModelo,

serie = \_serie,

serviceTag= \_serviceTag,

descripcionEquipo= \_descripcionEquipo

WHERE codEquipo = \_codEquipo ;

END;

$$

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarEquipo(IN \_codEquipo int)

BEGIN

DELETE FROM equipo WHERE codEquipo=\_codEquipo;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeCodigoEquipo(IN \_codigo varchar(20))

BEGIN

SELECT count(\*) from equipo

WHERE codEquipo = \_codigo;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeSerie(IN \_serie varchar(20))

BEGIN

SELECT count(\*) from equipo

WHERE serie = \_serie;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeServiceTag(IN \_serviceTag varchar(30))

BEGIN

SELECT count(\*) from equipo

WHERE serviceTag = \_serviceTag;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarMarca()

BEGIN

SELECT \* FROM marca;

END;

$$

**DETALLE EQUIPO COMPUTADORA**

PROCEDIMIENTO PARA MOSTRAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_mostrarDetalleEquipoC()

BEGIN

SELECT detequipocomputadora.codEquipo, tiporam.tipoRam, capacidad.capacidad, velocidadram.velocidadRam, tipodisco.tipoDisco, capacidad.capacidad FROM detequipocomputadora INNER JOIN velocidadram USING (idVelocidadRam) INNER JOIN capacidad USING (idCapacidad) INNER JOIN tipodisco USING (idTipoDisco) INNER JOIN tiporam USING (idTipoRam) INNER JOIN equipo USING (codEquipo);

END;

$$

PROCEDIMIENTO PARA INSERTAR

DELIMITER $$

CREATE PROCEDURE sp\_insertarDetalleEquipoC (IN \_codEquipo varchar(9), IN \_idTipoRam int , IN \_idCapacidadRam int, IN \_idVelocidadRam int, IN \_idTipoDisco int, IN \_idCapacidadDisco int)

BEGIN

INSERT INTO detequipocomputadora (codEquipo, idTipoRam, idCapacidad, idVelocidadRam, idTipoDisco, idCapacidadDisco) VALUES( \_codEquipo, \_idTipoRam, \_idCapacidadRam, \_idVelocidadRam, \_idTipoDisco, \_idCapacidadDisco);

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarVelocidadRam()

BEGIN

SELECT \* FROM velocidadram;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarCapacidad()

BEGIN

SELECT \* FROM capacidad;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarTipoDisco()

BEGIN

SELECT \* FROM tipoDisco;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarTipoRam()

BEGIN

SELECT \* FROM tiporam;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarEquipo1()

BEGIN

SELECT equipo.codEquipo, tipoequipo.tipoEquipo, modelo.modelo, equipo.serie, equipo.serviceTag, equipo.descripcionEquipo FROM equipo INNER JOIN tipoequipo USING (idTipoEquipo) INNER JOIN modelo USING(idModelo) ;

END;

$$

PROCEDIMIENTO PARA ACTUALIZAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_actualizarDetalleEquipoC (IN \_codEquipo varchar(9), IN \_idTipoRam int , IN \_idCapacidadRam int, IN \_idVelocidadRam int, IN \_idTipoDisco int, IN \_idCapacidadDisco int)

BEGIN

UPDATE detequipocomputadora

SET idTipoRam = \_idTipoRam,

idCapacidad = \_idCapacidadRam,

idVelocidadRam = \_idVelocidadRam,

idTipoDisco = \_idTipoDisco,

idCapacidadDisco = \_idCapacidadDisco

WHERE CodEquipo = \_codEquipo;

END;

$$

PROCEDIMIENTO PARA ELIMINAR

-----------------------------------------

DELIMITER $$

CREATE PROCEDURE sp\_eliminarDetalleEquipoC (IN \_codEquipo varchar(9))

BEGIN

DELETE FROM detequipocomputadora WHERE codEquipo = \_codEquipo;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeDetCodigoEquipo(IN \_codigo varchar(9))

BEGIN

SELECT count(\*) from detequipocomputadora

WHERE codEquipo = \_codigo;

END;

$$

**CONTROL GARANTÍA**

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_mostrarControlGarantia()

BEGIN

SELECT controlgarantia.idControlGarantia, equipo.codEquipo, equipo.descripcionEquipo, tipoequipo.tipoEquipo, modelo.modelo, equipo.serviceTag,

controlgarantia.fechaInicio, controlgarantia.fechaVencimiento, CASE controlgarantia.estado WHEN 0 THEN 'INACTIVO'

WHEN 1 THEN 'ACTIVO'

END AS estado,

empresagarantia.idEmpresa, empresagarantia.nombreEmpresa, controlgarantia.telefonoContacto, controlgarantia.correoElectronico, usuario.userName

FROM controlgarantia INNER JOIN equipo

ON controlgarantia.codEquipo = equipo.codEquipo

INNER JOIN tipoequipo

ON equipo.idTipoEquipo = tipoequipo.idTipoEquipo

INNER JOIN modelo

ON equipo.idModelo = modelo.idModelo

INNER JOIN empresagarantia

ON controlgarantia.idEmpresa = empresagarantia.idEmpresa

INNER JOIN usuario

ON controlgarantia.idUsuario = usuario.idUsuario;

END;

$$

PROCEDIMIENTO PARA INSERTAR

**call sp\_insertarControlGarantia(031, 2016-01-01, 2020-12-12, 1, 116)**

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_insertarControlGarantia (IN \_codEquipo varchar(9), IN \_fechaInicio date , IN \_fechaVencimiento date, IN \_idEmpresa int, IN \_telefono varchar(9), IN \_correo varchar(40), IN \_idUsuario int)

BEGIN

INSERT INTO controlgarantia (codEquipo, fechaInicio, fechaVencimiento, idEmpresa, telefonoContacto, correoElectronico, idUsuario)

VALUES(\_codEquipo, \_fechaInicio, \_fechaVencimiento, \_idEmpresa, \_telefono, \_correo, \_idUsuario);

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarEmpresaGarantia()

BEGIN

SELECT \* FROM empresagarantia;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarEquipo1()

BEGIN

SELECT equipo.codEquipo, tipoequipo.tipoEquipo, modelo.modelo, equipo.serie, equipo.serviceTag, equipo.descripcionEquipo FROM equipo INNER JOIN tipoequipo USING (idTipoEquipo) INNER JOIN modelo USING(idModelo) ;

END;

$$

----------------------------------------- PROCEDIMIENTO PARA ACTUALIZAR

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_actualizarControlGarantia (IN \_idControlGarantia int,

IN \_codEquipo varchar(9),

IN \_fechaInicio date,

IN \_fechaVencimiento date,

IN \_estado tinyint,

IN \_idEmpresa int,

IN \_telefonoContacto varchar(9),

IN \_correo varchar(40),

IN \_idUsuario int)

BEGIN

UPDATE controlgarantia

SET codEquipo = \_codEquipo,

fechaInicio = \_fechaInicio,

fechaVencimiento = \_fechaVencimiento,

estado = \_estado,

idEmpresa = \_idEmpresa,

telefonoContacto = \_telefonoContacto,

correoElectronico = \_correo,

idUsuario = \_idUsuario

WHERE idControlGarantia = \_idControlGarantia;

END;

----------------------------------------- PROCEDIMIENTO PARA ELIMINAR

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_eliminarControlGarantia (IN \_idControlGarantia int)

BEGIN

DELETE FROM controlgarantia WHERE idControlGarantia = \_idControlGarantia;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeControlGarantia (IN \_codEquipo varchar(9))

BEGIN

SELECT count(\*)

from controlgarantia

WHERE codEquipo = \_codEquipo;

END;

$$

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_obtenerCorreoEmpleadoControlGarantia(IN \_codEquipo varchar(9))

BEGIN

SELECT CONCAT(empleado.nombres, ' ', empleado.apellidos) as NombreEmpleado,empleado.correoElectronico

from usuario INNER JOIN asignacionequipo USING(idUsuario)

INNER JOIN empleado USING(codEmpleado)

INNER JOIN equipo USING(codEquipo)

WHERE codEquipo = \_codEquipo;

END

$$

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_obtenerCorreoEmpleadoControlGarantia(IN \_idEmpresa int)

BEGIN

SELECT nombreEmpresa

from empresagarantia

WHERE idEmpresa = \_idEmpresa;

END

$$

**ASIGNACIÓN EQUIPO**

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_mostrarAsignacionEquipo()

BEGIN

SELECT asignacionequipo.idAsignacionEquipo, equipo.codEquipo, equipo.descripcionEquipo, tipoequipo.tipoEquipo, modelo.modelo, equipo.serviceTag,

asignacionequipo.fechaAsignacion, asignacionequipo.fechaBaja, asignacionequipo.observacion, CASE asignacionequipo.estado WHEN 0 THEN 'INACTIVO'

WHEN 1 THEN 'ACTIVO'

END AS estado, concat(empleado.nombres, ' ', empleado.apellidos) as empleado

FROM asignacionequipo INNER JOIN equipo

ON asignacionequipo.codEquipo = equipo.codEquipo

INNER JOIN tipoequipo

ON equipo.idTipoEquipo = tipoequipo.idTipoEquipo

INNER JOIN modelo

ON equipo.idModelo = modelo.idModelo

INNER JOIN usuario

ON asignacionequipo.idUsuario = usuario.idUsuario

INNER JOIN empleado

ON usuario.codEmpleado = empleado.codEmpleado;

END;

$$

PROCEDIMIENTO PARA INSERTAR

**call sp\_insertarAsignacionEquipo( 2016-01-01, 031, "mysql",2020-12-12, 116)**

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_insertarAsignacionEquipo( IN \_fechaAsignacion date, IN \_codEquipo varchar(9), IN \_observacion varchar(50), IN \_fechaBaja date, IN \_idUsuario int)

BEGIN

INSERT INTO asignacionequipo (fechaAsignacion, codEquipo, observacion, fechaBaja, idUsuario)

VALUES(\_fechaAsignacion, \_codEquipo, \_observacion, \_fechaBaja, \_idUsuario);

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarEquipo1()

BEGIN

SELECT equipo.codEquipo, tipoequipo.tipoEquipo, modelo.modelo, equipo.serie, equipo.serviceTag, equipo.descripcionEquipo FROM equipo INNER JOIN tipoequipo USING (idTipoEquipo) INNER JOIN modelo USING(idModelo) ;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_mostrarUsuario()

BEGIN

SELECT usuario.idUsuario, concat(empleado.nombres, ' ', empleado.apellidos) as empleado, usuario.fechaCreacion, usuario.userName, usuario.password, usuario.fechaBaja, CASE usuario.estado WHEN 0 THEN 'INACTIVO' WHEN 1 THEN 'ACTIVO' END AS estado

FROM usuario INNER JOIN empleado USING (codEmpleado);END;

$$

PROCEDIMIENTO PARA ACTUALIZAR

-----------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_actualizarAsignacionEquipo( IN \_idAsignacion int,

IN \_fechaAsignacion date,

IN \_codEquipo varchar(9),

IN \_observacion varchar(50),

IN \_fechaBaja date,

IN \_estado tinyint,

IN \_idUsuario int)

BEGIN

UPDATE asignacionequipo

SET fechaAsignacion = \_fechaAsignacion,

codEquipo = \_codEquipo,

observacion = \_observacion,

fechaBaja = \_fechaBaja,

estado = \_estado,

idUsuario = \_idUsuario

WHERE idAsignacionEquipo = \_idAsignacion;

END;

PROCEDIMIENTO PARA ELIMINAR

-----------------------------------------

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_eliminarAsignacionEquipo(IN \_idAsignacionEquipo int)

BEGIN

DELETE FROM asignacionequipo WHERE idAsignacionEquipo = \_idAsignacionEquipo;

END;

$$

DELIMITER $$

CREATE PROCEDURE sp\_existeCodEquipo(IN \_codigo varchar(9))

BEGIN

SELECT count(\*)

from asignacionequipo

WHERE codEquipo = \_codigo;

END;

$$

DELIMITER $$

CREATE OR REPLACE PROCEDURE sp\_obtenerCorreoEmpleadoAsigEquipo(IN \_idUsuario int)

BEGIN

SELECT CONCAT(empleado.nombres, ' ', empleado.apellidos) as NombreEmpleado,empleado.correoElectronico

from usuario INNER JOIN asignacionequipo USING(idUsuario)

INNER JOIN empleado USING(codEmpleado)

WHERE idUsuario = \_idUsuario;

END

$$

Posibles Consultas

1. Cantidad de solicitudes pendientes y completos.

2. Cantidad de mantenimientos pendientes y completos.

3. Cantidad de solicitudes y mantenimientos por rangos de fechas.

4. Usuario con más solicitudes.

5. Usuarios (técnicos) con solicitudes pendientes.

6. Departamento con más solicitudes pendientes.

Reportes

7. Reporte de unidades que más solicitudes realizan.

8. Reporte de solicitudes por técnico o usuario asignado.

9. Mantenimientos pendientes por rango de fecha.

10. Mantenimientos pendientes por departamentos.

1. Completos

select COUNT(idControlMantenimiento)

from controlmantenimiento

WHERE idEstadoControlMantenimiento = 3;

Pendientes

select COUNT(idControlMantenimiento)

from controlmantenimiento

WHERE idEstadoControlMantenimiento = 1 OR idEstadoControlMantenimiento = 2;