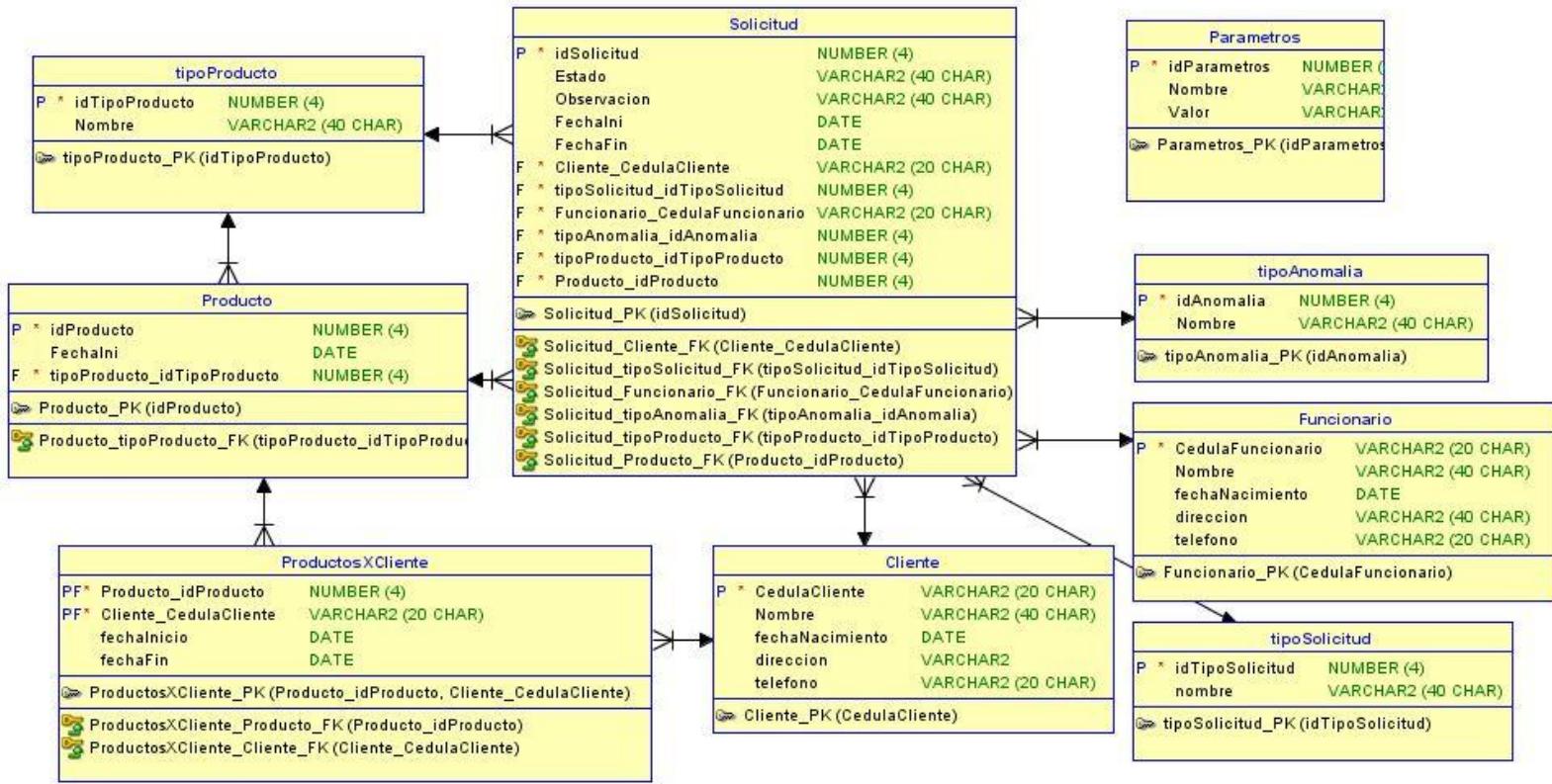


Proyecto final BD

- Diana Torres
- Jesús Paz
- Alejandro Alvarado
- Felipe Cortes
- J. Steven Delgado

Modelo



DDL

- [DDL_Modelo.ddl](#)

Paquetes:

1 Nivel

- [Clientes.sql](#)
- [Funcionario.sql](#)
- [Producto.sql](#)
- [ProductosXCliente.sql](#)
- [Solicitud.sql](#)
- [TipoSolicitud.sql](#)
- [TipoAnomalia.sql](#)
- [TipoProducto.sql](#)

```

CREATE OR REPLACE PACKAGE pkClientesNivel1 IS -- spec
    PROCEDURE pInsertar(ivNombre IN VARCHAR2, ivCedula IN VARCHAR2, ivFechaNacimiento IN DATE, ivDireccion IN VARCHAR2 ,ivTelefono IN VARCHAR2);
    PROCEDURE pEliminar(iCedulaCliente IN NUMBER);
    FUNCTION fConsultar(ivCedula IN VARCHAR2) RETURN cliente%rowtype;
    PROCEDURE pModificar(ivCedula IN VARCHAR2,ivNombre IN VARCHAR2, ivFechaNacimiento IN DATE, ivDireccion IN VARCHAR2 ,ivTelefono IN VARCHAR2);
END pkClientesNivel1;
/
CREATE OR REPLACE PACKAGE BODY pkClientesNivel1 IS -- body
-- Insertar
    PROCEDURE pInsertar(ivNombre IN VARCHAR2, ivCedula IN VARCHAR2, ivFechaNacimiento IN DATE, ivDireccion IN VARCHAR2 ,ivTelefono IN VARCHAR2) IS
        BEGIN
            INSERT INTO cliente
            VALUES ( ivCedula,ivNombre, ivFechaNacimiento, ivDireccion, ivTelefono);
            EXCEPTION
            WHEN DUP_VAL_ON_INDEX THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, este registro ya existe.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pInsertar;
-- Eliminar
    PROCEDURE pEliminar(iCedulaCliente IN NUMBER) IS
        BEGIN
            DELETE FROM cliente
            WHERE cedulacliente = iCedulaCliente;
            EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pEliminar;
-- Actualizar
    PROCEDURE pModificar(ivCedula IN VARCHAR2,ivNombre IN VARCHAR2, ivFechaNacimiento IN DATE, ivDireccion IN VARCHAR2 ,ivTelefono IN VARCHAR2) IS
        BEGIN
            UPDATE cliente
            SET nombre=ivNombre, cedulacliente=ivCedula, fechaNacimiento=ivFechaNacimiento, direccion=ivDireccion, telefono=ivTelefono
            WHERE cedulacliente=ivCedula;
            EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pModificar;
-- Obtener
    FUNCTION fConsultar(ivCedula IN VARCHAR2) RETURN cliente%rowtype
    IS ovCliente cliente%rowtype;
        BEGIN
            SELECT * into ovCliente
            FROM cliente
            WHERE cedulacliente=ivCedula;
            RETURN ovCliente;
            EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END fConsultar;
END_pkClientesNivel1.

```

```

CREATE OR REPLACE PACKAGE pkFuncionarioNivel1 IS |
    PROCEDURE pInsertar(cedulaFuncionario IN VARCHAR2, nombre IN VARCHAR2, fechaNacimiento IN DATE, direccion IN VARCHAR2, telefono IN VARCHAR2 );
    PROCEDURE pEliminar(cedulaFuncionario IN VARCHAR2);
    PROCEDURE pModificar(cedulaFuncionario IN VARCHAR2, inombre IN VARCHAR2, fechaNacimiento IN DATE, direccion IN VARCHAR2, telefono IN VARCHAR2 );
    FUNCTION fConsultar(cedulaFuncionario IN VARCHAR2) RETURN FUNCIONARIO%rowtype;
END pkFuncionarioNivel1;
/
CREATE OR REPLACE PACKAGE BODY pkFuncionarioNivel1 IS
-- Insertar
PROCEDURE pInsertar
    (cedulaFuncionario IN VARCHAR2, nombre IN VARCHAR2, fechaNacimiento IN DATE, direccion IN VARCHAR2, telefono IN VARCHAR2)
IS
BEGIN
    INSERT INTO FUNCIONARIO
    VALUES (cedulaFuncionario, nombre, fechaNacimiento, direccion, telefono );
    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            RAISE_APPLICATION_ERROR(-20001,'Error, este registro ya existe.');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END pInsertar;
-- Eliminar
PROCEDURE pEliminar(cedulaFuncionario IN VARCHAR2) IS
BEGIN
    DELETE FROM FUNCIONARIO T WHERE T.CEDULAFUNCIONARIO= cedulaFuncionario;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20001,'Error, no existe un funcionario con esa cédula.');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END pEliminar;
-- Actualizar
PROCEDURE pModificar(cedulaFuncionario IN VARCHAR2, inombre IN VARCHAR2, fechaNacimiento IN DATE, direccion IN VARCHAR2, telefono IN VARCHAR2)
IS
BEGIN
    UPDATE FUNCIONARIO
    SET NOMBRE=inombre
    WHERE CEDULAFUNCIONARIO=cedulaFuncionario;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20001,'Error,no existe un funcionario con esa cédula.');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END pModificar;
FUNCTION fConsultar(cedulaFuncionario IN VARCHAR2) RETURN FUNCIONARIO%rowtype
IS ovFuncionario FUNCIONARIO%rowtype;
BEGIN
    SELECT * into ovFuncionario
    FROM FUNCIONARIO
    WHERE CEDULAFUNCIONARIO=cedulaFuncionario;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20001,'Error,no existe un funcionario con esa cédula.');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END fConsultar;
END pkFuncionarioNivel1;

```

```

CREATE OR REPLACE PACKAGE pkProductoNivel1 IS
    PROCEDURE pInsertar(ivIdProducto IN PRODUCTO.IDPRODUCTO%type, ivFechaIni IN PRODUCTO.FECHAINI%type, ivTipoProducto IN PRODUCTO.TIPOPRODUCTO_IDTIPOPRODUCTO%type);
    PROCEDURE pEliminar(ivIdProducto IN NUMBER);
    PROCEDURE pModificar(ivIdProducto IN NUMBER, ivFechaIni IN DATE, ivTipoProducto IN NUMBER );
    FUNCTION fConsultar(ivIdProducto IN NUMBER) RETURN PRODUCTO%rowtype;
END pkProductoNivel1;
/
CREATE OR REPLACE PACKAGE BODY pkProductoNivel1 IS
-- Insertar
    PROCEDURE pInsertar
        (ivIdProducto IN PRODUCTO.IDPRODUCTO%type, ivFechaIni IN PRODUCTO.FECHAINI%type, ivTipoProducto IN PRODUCTO.TIPOPRODUCTO_IDTIPOPRODUCTO%type)
    IS
        BEGIN
        INSERT INTO PRODUCTO(IDPRODUCTO, FECHAINI, TIPOPRODUCTO_IDTIPOPRODUCTO)
        VALUES (ivIdProducto, ivFechaIni, ivTipoProducto);
        EXCEPTION
            WHEN DUP_VAL_ON_INDEX THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, este registro ya existe.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
    END pInsertar;
-- Eliminar
    PROCEDURE pEliminar(ivIdProducto IN NUMBER) IS
        BEGIN
        DELETE FROM PRODUCTO P
        WHERE P.IDPRODUCTO = ivIdProducto;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
    END pEliminar;
-- Actualizar
    PROCEDURE pModificar(ivIdProducto NUMBER, ivFechaIni DATE, ivTipoProducto NUMBER ) IS
        BEGIN
        UPDATE PRODUCTO
        SET _idProducto=ivIdProducto, fechaIni=ivFechaIni, TIPOPRODUCTO_IDTIPOPRODUCTO=ivTipoProducto;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
    END pModificar;
    FUNCTION fConsultar(ivIdProducto NUMBER) RETURN PRODUCTO%rowtype
    IS ovProducto PRODUCTO%rowtype;
        BEGIN
        SELECT * into ovProducto
        FROM PRODUCTO
        WHERE idProducto=ivIdProducto;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
    END fConsultar;
END_pkProductoNivel1.

```

```

CREATE OR REPLACE PACKAGE pkProductosXClienteNivel1 IS
    PROCEDURE pInsertar(ivIdProducto IN NUMBER, ivCedula IN VARCHAR2,ivFechaInicio IN DATE,ivFechaFin IN DATE);
    PROCEDURE pEliminar(ivIdProducto IN NUMBER, ivCedula IN VARCHAR2);
    PROCEDURE pModificar(ivIdProducto IN NUMBER, ivCedula IN VARCHAR2,ivFechaInicio IN DATE,ivFechaFin IN DATE);
    FUNCTION fConsultar(ivCedula IN VARCHAR2) RETURN PRODUCTOSXCLIENTE%rowtype;
END pkProductosXClienteNivel1;
/
CREATE OR REPLACE PACKAGE BODY pkProductosXClienteNivel1 IS
    -- Insertar
    PROCEDURE pInsertar (ivIdProducto IN NUMBER, ivCedula IN VARCHAR2,ivFechaInicio IN DATE,ivFechaFin IN DATE) IS
        BEGIN
            INSERT INTO PRODUCTOSXCLIENTE
            VALUES (ivIdProducto,ivCedula,ivFechaInicio,ivFechaFin);
            EXCEPTION
                WHEN DUP_VAL_ON_INDEX THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error, este registro ya existe.');
                WHEN OTHERS THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pInsertar;
    -- Eliminar
    PROCEDURE pEliminar(ivIdProducto IN NUMBER, ivCedula IN VARCHAR2) IS
        BEGIN
            DELETE FROM PRODUCTOSXCLIENTE S WHERE S.PRODUCTO_IDPRODUCTO = ivIdProducto
            AND S.CLIENTE_CEDULACLIENTE=ivCedula;
            EXCEPTION
                WHEN NO_DATA_FOUND THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error, no existe cliente o producto con ese id.');
                WHEN OTHERS THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pEliminar;
    -- Actualizar
    PROCEDURE pModificar(ivIdProducto IN NUMBER, ivCedula IN VARCHAR2,ivFechaInicio IN DATE,ivFechaFin IN DATE) IS
        BEGIN
            UPDATE PRODUCTOSXCLIENTE
            SET FECHAINICIO=ivFechaInicio, FECHAFIN=ivFechaFin
            WHERE PRODUCTO_IDPRODUCTO = ivIdProducto
            AND CLIENTE_CEDULACLIENTE=ivCedula;
            EXCEPTION
                WHEN NO_DATA_FOUND THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error, no existe cliente o producto con ese id.');
                WHEN OTHERS THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pModificar;
    FUNCTION fConsultar(ivCedula in VARCHAR2) RETURN PRODUCTOSXCLIENTE%rowtype
    IS ovProducto PRODUCTOSXCLIENTE%rowtype;
    BEGIN
        SELECT * into ovProducto
        FROM PRODUCTOSXCLIENTE
        WHERE CLIENTE_CEDULACLIENTE=ivCedula;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, no existe Cliente con esa Cedula.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
    END fConsultar;
END pkProductosXClienteNivel1;

```

```

CREATE OR REPLACE PACKAGE pkSolicitudNivel1 IS
  PROCEDURE pInsertar(ivIdSolicitud IN NUMBER, ivEstado IN VARCHAR2,ivObservacion IN VARCHAR2,ivIdCliente IN VARCHAR2,ivIdFuncionario IN VARCHAR2
  ,ivTipoSolicitud IN NUMBER,ivTipoAnomalia IN NUMBER,ivIdTipoProducto IN NUMBER,ivIdProducto IN NUMBER );
  PROCEDURE pEliminar(ivIdSolicitud IN NUMBER);
  PROCEDURE pModificar(ivIdSolicitud IN NUMBER, ivEstado IN VARCHAR2,ivObservacion IN VARCHAR2,ivIdCliente IN VARCHAR2,ivIdFuncionario IN VARCHAR2
  ,ivFechaIn DATE,ivFechaFin DATE,ivTipoSolicitud IN NUMBER,ivTipoAnomalia IN NUMBER,ivIdTipoProducto IN NUMBER,ivIdProducto IN NUMBER );
  FUNCTION fConsultar(ivIdSolicitud IN NUMBER) RETURN SOLICITUD%ROWTYPE;
  PROCEDURE pCambiarEstado(ivIdSolicitud IN NUMBER, ivEstado IN VARCHAR2,ivIdFuncionario IN VARCHAR2);
END pkSolicitudNivel1;
/
CREATE OR REPLACE PACKAGE BODY pkSolicitudNivel1 IS
-- Insertar
  PROCEDURE pInsertar
    (ivIdSolicitud NUMBER, ivEstado VARCHAR2,ivObservacion VARCHAR2,ivIdCliente VARCHAR2,ivIdFuncionario VARCHAR2
    ,ivTipoSolicitud NUMBER,ivTipoAnomalia NUMBER,ivIdTipoProducto NUMBER,ivIdProducto NUMBER )
  IS
  BEGIN
    INSERT INTO SOLICITUD
    VALUES (ivIdSolicitud , ivEstado , ivObservacion ,null,null,ivIdCliente
    ,ivTipoSolicitud ,ivIdFuncionario,ivTipoAnomalia,ivIdTipoProducto ,ivIdProducto );
    EXCEPTION
      WHEN DUP_VAL_ON_INDEX THEN
        RAISE_APPLICATION_ERROR(-20001,'Error, este registro ya existe.');
      WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001,'Error desconocido.' ||SQLERRM||SQLCODE);
  END pInsertar;
-- Eliminar
  PROCEDURE pEliminar(ivIdSolicitud IN NUMBER)
  IS
  BEGIN
    DELETE FROM SOLICITUD S WHERE S.IDSOLICITUD = ivIdSolicitud;
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
      WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001,'Error desconocido.' ||SQLERRM||SQLCODE);
  END pEliminar;
-- Actualizar
  PROCEDURE pModificar(ivIdSolicitud NUMBER, ivEstado VARCHAR2,ivObservacion VARCHAR2,ivIdCliente VARCHAR2,ivIdFuncionario VARCHAR2
  ,ivFechaIn DATE,ivFechaFin DATE,ivTipoSolicitud NUMBER,ivTipoAnomalia NUMBER,ivIdTipoProducto NUMBER,ivIdProducto NUMBER ) IS
  IS
  BEGIN
    UPDATE SOLICITUD
    SET Estado=ivEstado ,Observacion=ivObservacion ,FechaIn=ivFechaIn,FechaFin=ivFechaFin,CLIENTE_CEDULACLIENTE=ivIdCliente
    ,TIPOSOLICITUD_IDTIPOSOLICITUD=ivTipoSolicitud ,FUNCIONARIO_CEDULAFUNCIONARIO=ivIdFuncionario, TIPOANOMALIA_IDANOMALIA=ivTipoAnomalia
    ,TIPOPRODUCTO_IDTIPOPRODUCTO=ivIdTipoProducto ,PRODUCTO_IDPRODUCTO=ivIdProducto
    WHERE IDSOLICITUD=ivIdSolicitud;
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
      WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001,'Error desconocido.' ||SQLERRM||SQLCODE);
  END pModificar;
  FUNCTION fConsultar(ivIdSolicitud IN NUMBER) RETURN SOLICITUD%ROWTYPE
  IS
  BEGIN
    SELECT * into avSolicitud
    FROM Solicitud
    WHERE IDSOLICITUD=ivIdSolicitud;
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
      WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001,'Error desconocido.' ||SQLERRM||SQLCODE);
  END fConsultar;
  PROCEDURE pCambiarEstado(ivIdSolicitud IN NUMBER, ivEstado IN VARCHAR2,ivIdFuncionario IN VARCHAR2)
  IS
  BEGIN
    UPDATE SOLICITUD
    SET Estado=ivEstado,FUNCIONARIO_CEDULAFUNCIONARIO=ivIdFuncionario
    WHERE IDSOLICITUD=ivIdSolicitud;
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
      WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001,'Error desconocido.' ||SQLERRM||SQLCODE);
  END pCambiarEstado;
END pkSolicitudNivel1;

```

```

create or replace PACKAGE pkTipoAnomaliaNivel1 IS
    PROCEDURE pInsertar(LidAnomalia IN NUMBER, LNOMBRE IN VARCHAR2);
    PROCEDURE pEliminar(LidAnomalia IN NUMBER);
    PROCEDURE pModificar(LidAnomalia IN NUMBER, LNOMBRE IN VARCHAR2);
    FUNCTION fConsultar(LidAnomalia in NUMBER) RETURN TIPOANOMALIA%rowtype;
END pkTipoAnomaliaNivel1;
/

create or replace PACKAGE BODY pkTipoAnomaliaNivel1 IS
-- Insertar
    PROCEDURE pInsertar
        (LidAnomalia NUMBER, LNOMBRE VARCHAR2)
    IS
        BEGIN
            INSERT INTO TIPOANOMALIA
            VALUES (LidAnomalia , LNOMBRE);

            EXCEPTION
                WHEN DUP_VAL_ON_INDEX THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error, este registro ya existe.');
                WHEN OTHERS THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pInsertar;

-- Eliminar
    PROCEDURE pEliminar(LidAnomalia IN NUMBER) IS
        BEGIN
            DELETE FROM TIPOANOMALIA S WHERE S.IDANOMALIA = LidAnomalia;
            EXCEPTION
                WHEN NO_DATA_FOUND THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
                WHEN OTHERS THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pEliminar;

-- Actualizar
    PROCEDURE pModificar(LidAnomalia NUMBER, LNOMBRE VARCHAR2) IS
        BEGIN
            UPDATE TIPOANOMALIA
            SET IDANOMALIA=LidAnomalia ,NOMBRE=LNOMBRE
            WHERE IDANOMALIA=LidAnomalia;

            EXCEPTION
                WHEN NO_DATA_FOUND THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
                WHEN OTHERS THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pModificar;

-- OBTENER
    FUNCTION fConsultar(LidAnomalia IN NUMBER) RETURN TIPOANOMALIA%rowtype
    IS ovTipoAnomalia TIPOANOMALIA%rowtype;
    BEGIN
        SELECT * into ovTipoAnomalia
        FROM TIPOANOMALIA
        WHERE IDANOMALIA=LidAnomalia;

        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);

    END fConsultar;
END pkTipoAnomaliaNivel1;

```

```

CREATE OR REPLACE PACKAGE pkTipoProductoNivel1 IS
    PROCEDURE pInsertar(LidTipoProducto IN NUMBER, LNombre IN VARCHAR2);
    PROCEDURE pEliminar(LidTipoProducto IN NUMBER);
    PROCEDURE pModificar(LidTipoProducto IN NUMBER, LNombre IN VARCHAR2);
    FUNCTION fConsultar(LidTipoProducto in NUMBER) RETURN TIPOPRODUCTO%rowtype;
END pkTipoProductoNivel1;
/
CREATE OR REPLACE PACKAGE BODY pkTipoProductoNivel1 IS
-- Insertar
PROCEDURE pInsertar
(LidTipoProducto NUMBER, LNombre VARCHAR2)
IS
BEGIN
INSERT INTO TIPOPRODUCTO
VALUES (LidTipoProducto , LNombre);

EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
RAISE_APPLICATION_ERROR(-20001,'Error, este registro ya existe.');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END pInsertar;

--Eliminar
PROCEDURE pEliminar(LidTipoProducto IN NUMBER) IS
BEGIN
DELETE FROM TIPOPRODUCTO S WHERE S.IDTIOPRODUCTO = LidTipoProducto;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END pEliminar;

-- Actualizar
PROCEDURE pModificar(LidTipoProducto NUMBER, LNombre VARCHAR2) IS
BEGIN
UPDATE TIPOPRODUCTO
SET IDTIOPRODUCTO=LidTipoProducto ,NOMBRE=LNombre
WHERE IDTIOPRODUCTO=LidTipoProducto;

EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END pModificar;

--OBTENER
FUNCTION fConsultar(LidTipoProducto IN NUMBER) RETURN TIPOPRODUCTO%rowtype
IS ovTipoProducto TIPOPRODUCTO%rowtype;
BEGIN
SELECT * into ovTipoProducto
FROM TIPOPRODUCTO
WHERE IDTIOPRODUCTO=LidTipoProducto;

EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);

END fConsultar;
|
END pkTipoProductoNivel1;

```

```

CREATE OR REPLACE PACKAGE pkTipoSolicitudNivel1 IS
    PROCEDURE pInsertar(idTipoSolicitud IN NUMBER, nombre IN VARCHAR2 );
    PROCEDURE pEliminar(idTipoSolicitud IN NUMBER);
    PROCEDURE pModificar(idTipoSolicitud IN NUMBER, inombre IN VARCHAR2);
    FUNCTION fConsultar(idTipoSolicitud in NUMBER) RETURN TIPOSOLICITUD%rowtype;
END pkTipoSolicitudNivel1;
/
CREATE OR REPLACE PACKAGE BODY pkTipoSolicitudNivel1 IS
-- Insertar
-- PROCEDURE pInsertar
-- (idTipoSolicitud NUMBER,nombre VARCHAR2)
IS
BEGIN
    INSERT INTO TIPOSOLICITUD
    VALUES (idTipoSolicitud , nombre );
    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            RAISE_APPLICATION_ERROR(-20001,'Error, este registro ya existe.');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END pInsertar;
-- Eliminar
PROCEDURE pEliminar(idTipoSolicitud IN NUMBER) IS
BEGIN
    DELETE FROM TIPOSOLICITUD T WHERE T.IDTIPOSOLICITUD = idTipoSolicitud;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20001,'Error, no existe un tipo de solicitud con ese id.');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END pEliminar;
-- Actualizar
PROCEDURE pModificar(idTipoSolicitud NUMBER, inombre VARCHAR2) IS
BEGIN
    UPDATE TIPOSOLICITUD
    SET  nombre=inombre
    WHERE IDTIPOSOLICITUD=idTipoSolicitud;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20001,'Error, no existe un tipo de solicitud con ese id.');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END pModificar;
FUNCTION fConsultar(idTipoSolicitud IN NUMBER) RETURN TIPOSOLICITUD%rowtype
IS ovTipoSolicitud TIPOSOLICITUD%rowtype;
BEGIN
    SELECT * into ovTipoSolicitud
    FROM TIPOSOLICITUD
    WHERE IDTIPOSOLICITUD=idTipoSolicitud;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20001,'Error, no existe un tipo de solicitud con ese id.');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
END fConsultar;
END pkTipoSolicitudNivel1;

```

2 Nivel

- [pkAsignacionNivel2.sql](#)
- [pkAtencion2.sql](#)
- [pkInicioSesionNivel2.sql](#)
- [pkRegistroNivel2.sql](#)

```
CREATE OR REPLACE PACKAGE pkAsignacionNivel2 IS
  PROCEDURE pAsignacionAutomatica(ivIdSolicitud IN NUMBER);
  PROCEDURE pAsignacionIndividual(ivIdSolicitud IN NUMBER,ivFuncionario IN VARCHAR2);

END pkAsignacionNivel2;
/
CREATE OR REPLACE PACKAGE BODY pkAsignacionNivel2 IS

  PROCEDURE pAsignacionAutomatica(ivIdSolicitud IN NUMBER)
  IS
    cedulaF VARCHAR2(20);
    cantidad NUMBER;
  BEGIN
    --SELECT P.Valor into cantidad
    --FROM PARAMETROS P
    --WHERE P.idParametros=0001;
    -- FALTA EL ID
    SELECT F.CEDULAFUNCIONARIO into cedulaF
    FROM FUNCIONARIO F
    WHERE ROWNUM = 1;

    PKSOLICITUDNIVEL1.PCAMBIARESTADO(ivIdSolicitud,'Asignada',cedulaF);
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
      WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);

  END pAsignacionAutomatica;

  PROCEDURE pAsignacionIndividual(ivIdSolicitud IN NUMBER,ivFuncionario IN VARCHAR2)
  IS
  BEGIN
    PKSOLICITUDNIVEL1.PCAMBIARESTADO(ivIdSolicitud,'Asignada',ivFuncionario);
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'Error, no existe solicitud con ese id.');
      WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
  END pAsignacionIndividual;

END pkAsignacionNivel2;
```

```

create or replace PACKAGE pkAtencionNivel2 IS
FUNCTION fValidarAtencion(ivCedulaFuncionario IN VARCHAR2,ivIdSolicitud IN NUMBER) RETURN NUMBER;
PROCEDURE pAtenderSolicitud(ivCedulaFuncionario IN VARCHAR2,ivIdSolicitud IN NUMBER ,ivComentario IN VARCHAR2);
PROCEDURE pAtenderReclamoDano(ivCedulaFuncionario IN NUMBER,ivIdSolicitud IN VARCHAR2, ivEstado IN VARCHAR2,ivComentario IN VARCHAR2);
PROCEDURE pAtenderReclamoAutomatico(ivIdSolicitud IN VARCHAR2);
END pkAtencionNivel2;
create or replace PACKAGE BODY pkAtencionNivel2 IS

FUNCTION fValidarAtencion(ivCedulaFuncionario IN VARCHAR2,ivIdSolicitud IN NUMBER) RETURN NUMBER IS
solicitudAct SOLICITUD%rowtype;
retorno NUMBER;
BEGIN
solicitudAct :=pkSOLICITUDNIVEL1.FCONSULTAR(ivIdSolicitud);

IF solicitudAct.FUNCIONARIO_CEDULAFUNCIONARIO=ivCedulaFuncionario THEN
RETURN retorno;
ELSE
RETURN retorno:=0;
END IF;
END fValidarAtencion;

PROCEDURE pAtenderSolicitud(ivCedulaFuncionario IN VARCHAR2,ivIdSolicitud IN NUMBER ,ivComentario IN VARCHAR2) IS
fecha DATE;
solicitudAct SOLICITUD%rowtype;
BEGIN
solicitudAct :=pkSOLICITUDNIVEL1.FCONSULTAR(ivIdSolicitud);

IF solicitudAct.FUNCIONARIO_CEDULAFUNCIONARIO=ivCedulaFuncionario THEN
IF solicitudAct.TIPOSOLICITUDDTIPOSOLICITUDB='001' THEN
IF solicitudAct.MONTO_SALIDA_NETA='1,1';--El primer dia del proximo mes
DEPRODUCTO_IDPRODUCTO=solicitudAct.IDPRODUCTO,solicitudAct.CLIENTE_CEDULACLIENTE,fecha,NULL;
pkSOLICITUDNIVEL1.MODIFICAR(solicitudAct.IDSOLICITUD,'Atendida',solicitudAct.CLIENTE_CEDULACLIENTE,solicitudAct.FUNCIONARIO_CEDULAFUNCIONARIO,ivComentario,
solicitudAct.TIPOSOLICITUDDTIPOSOLICITUDB,solicitudAct.IDTIPOSOLICITUDB,solicitudAct.TIPOANOMALIA_IDANOMALIA,solicitudAct.TIPOPRODUCTO_IDTIPOPRODUCTO,
solicitudAct.PRODUCTO_IDPRODUCTO);
ELSIF solicitudAct.TIPOSOLICITUDDTIPOSOLICITUDB='002' THEN
fecha:=SYSDATE;
DEPRODUCTO_IDPRODUCTO=solicitudAct.CLIENTE_CEDULACLIENTE,fecha,NULL;
pkSOLICITUDNIVEL1.MODIFICAR(solicitudAct.IDSOLICITUD,'Atendida',solicitudAct.CLIENTE_CEDULACLIENTE,solicitudAct.FUNCIONARIO_CEDULAFUNCIONARIO,ivComentario,
solicitudAct.FECHAFIN,solicitudAct.TIPOSOLICITUDDTIPOSOLICITUDB,solicitudAct.TIPOANOMALIA_IDANOMALIA,solicitudAct.TIPOPRODUCTO_IDTIPOPRODUCTO,
solicitudAct.PRODUCTO_IDPRODUCTO);
END IF;
ELSE
RAISE_APPLICATION_ERROR(-20001,'Esta solicitud no esta asociada al funcionario buscado. ');
END IF;
END pAtenderSolicitud;

PROCEDURE pAtenderReclamoDano(ivCedulaFuncionario IN NUMBER,ivIdSolicitud IN VARCHAR2, ivEstado IN VARCHAR2,ivComentario IN VARCHAR2)--p2
IS
solicitudAct SOLICITUD%rowtype;
BEGIN
solicitudAct :=pkSOLICITUDNIVEL1.FCONSULTAR(ivIdSolicitud);

IF solicitudAct.JDSOLICITUDDTIPOSOLICITUDB IS NULL THEN
RAISE_APPLICATION_ERROR(-20001,'No existe una solicitud con el ID buscado.');
END IF;
IF solicitudAct.FUNCIONARIO_CEDULAFUNCIONARIO=ivCedulaFuncionario THEN
IF solicitudAct.TIPOSOLICITUDDTIPOSOLICITUDB='001' THEN
pkSOLICITUDNIVEL1.MODIFICAR(solicitudAct.IDSOLICITUD,'Atendida',solicitudAct.CLIENTE_CEDULACLIENTE,ivEstado,ivComentario,solicitudAct.FECHAINI,solicitudAct.FECHAFIN,
solicitudAct.TIPOSOLICITUDDTIPOSOLICITUDB,solicitudAct.IDTIPOSOLICITUDB,solicitudAct.TIPOANOMALIA_IDANOMALIA,solicitudAct.TIPOPRODUCTO_IDTIPOPRODUCTO,
solicitudAct.PRODUCTO_IDPRODUCTO);
END IF;
END IF;
END pAtenderReclamoDano;

PROCEDURE pAtenderReclamoDanoAutomatico(ivIdSolicitud IN VARCHAR2)--p3
IS
solicitudAct SOLICITUD%rowtype;
tempoLimite NUMBER;
expiradas SOLICITUD%rowtype;
cursor sys_refcursor;
BEGIN
SELECT P.valor into tempolimite
FROM PARAMETROS_P
WHERE P.idParametros=0002;

SELECT S.IdSolicitud into expiradas
FROM SOLICITUD_S
WHERE SOLICITUD_S.FECHAFIN --> tempolimite
AND ROWNUM = 1
AND S.ESTADO='Asignado'
AND (S.TIPOSOLICITUDDTIPOSOLICITUDB=003 OR S.TIPOSOLICITUDDTIPOSOLICITUDB=004);

IF solicitudAct.IDSOLICITUD IS NULL THEN
RAISE_APPLICATION_ERROR(-20001,'No existe una solicitud con el ID buscado.');
END IF;

IF solicitudAct.TIPOSOLICITUDDTIPOSOLICITUDB=003 OR solicitudAct.TIPOSOLICITUDDTIPOSOLICITUDB=004 THEN
RAISE_APPLICATION_ERROR(-20001,'La solicitud pasada por parametro no es de tipo dano o reclamo.');
ELSE
--Inicia del loop
loop
fetch cur1 into expiradas;
exit when cur1%notfound;
solicitudAct :=pkSOLICITUDNIVEL1.FCONSULTAR(expiradas);
pkSOLICITUDNIVEL1.MODIFICAR(expiradas,'Atendida automaticamente por el sistema',solicitudAct.CLIENTE_CEDULACLIENTE,solicitudAct.FUNCIONARIO_CEDULAFUNCIONARIO,
solicitudAct.FECHAINI,solicitudAct.FECHAFIN,solicitudAct.TIPOSOLICITUDDTIPOSOLICITUDB,solicitudAct.IDTIPOSOLICITUDB,solicitudAct.TIPOANOMALIA_IDANOMALIA,solicitudAct.TIPOPRODUCTO_IDTIPOPRODUCTO,
solicitudAct.PRODUCTO_IDPRODUCTO);
END loop;
END IF;
END pAtenderReclamoDanoAutomatico;

```

```
| CREATE OR REPLACE PACKAGE pkInicioSesionNivel2 IS
| 
|     FUNCTION fAcceder(ivIdCliente  VARCHAR2) RETURN CLIENTE%rowtype;
|     PROCEDURE pRegistrar(ivIdCliente IN VARCHAR2,ivNombre in VARCHAR2,ivFechaNacimiento Date,ivDireccion in VARCHAR2, ivTelefono in VARCHAR2);
| 
END pkInicioSesionNivel2;
/
CREATE OR REPLACE PACKAGE BODY pkInicioSesionNivel2 IS
-- Insertar
-- Insertar
    FUNCTION fAcceder(ivIdCliente  VARCHAR2)RETURN CLIENTE%rowtype
    IS
        ovCliente cliente%rowtype;
    BEGIN
        ovCliente:=PKCLIENTESNIVEL1.FCONSULTAR(ivIdCliente);
        RETURN ovCliente;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20001,'Error, Cliente no encontrado');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
    END fAcceder;

    PROCEDURE pRegistrar(ivIdCliente IN VARCHAR2,ivNombre in VARCHAR2,ivFechaNacimiento Date,ivDireccion in VARCHAR2, ivTelefono in VARCHAR2)
    IS
    BEGIN
        PKCLIENTESNIVEL1.pInsertar(ivNombre,ivIdCliente,ivFechaNacimiento ,ivDireccion , ivTelefono);
    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            RAISE_APPLICATION_ERROR(-20001,'Error, este cliente ya existe.');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
    END pRegistrar;
END pkInicioSesionNivel2;
```

```
CREATE OR REPLACE PACKAGE pkRegistroNivel2 IS

    PROCEDURE pRegistrarSolicitud(ivIdSolicitud IN NUMBER,ivObservacion IN VARCHAR2,ivIdCliente IN VARCHAR2,ivTipoSolicitud IN NUMBER
    ,ivTipoAnomalia NUMBER,ivIdTipoProducto IN NUMBER,ivIdProducto IN NUMBER);

END pkRegistroNivel2;
/
CREATE OR REPLACE PACKAGE BODY pkRegistroNivel2 IS
-- Insertar
    PROCEDURE pRegistrarSolicitud(ivIdSolicitud IN NUMBER,ivObservacion IN VARCHAR2,ivIdCliente IN VARCHAR2,ivTipoSolicitud IN NUMBER
    ,ivTipoAnomalia NUMBER,ivIdTipoProducto IN NUMBER,ivIdProducto IN NUMBER)
    IS
        BEGIN
            PKSOLICITUDNIVEL1.pInsertar(ivIdSolicitud , 'Pendiente',ivObservacion,ivIdCliente , '000'
            ,ivTipoSolicitud,ivTipoAnomalia,ivIdTipoProducto ,ivIdProducto );

            EXCEPTION
                WHEN DUP_VAL_ON_INDEX THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error, el id de la solicitud ya existe.');
                WHEN OTHERS THEN
                    RAISE_APPLICATION_ERROR(-20001,'Error desconocido.'||SQLERRM||SQLCODE);
        END pRegistrarSolicitud;

    END pkRegistroNivel2;
```

3 Nivel

- [pkAsignacionNivel3.sql](#)
- [pkAtencion3.sql](#)
- [pkInicioSesionNivel3.sql](#)
- [pkRegistroNivel3.sql](#)

```
CREATE OR REPLACE PACKAGE pkAsignacionNivel3 IS
    PROCEDURE pAsignacionAutomatica(ivIdSolicitud IN NUMBER, ovRetorno out VARCHAR2);
    PROCEDURE pAsignacionIndividual(ivIdSolicitud IN NUMBER,ivFuncionario IN VARCHAR2,ovRetorno out VARCHAR2);

END pkAsignacionNivel3;
/
CREATE OR REPLACE PACKAGE BODY pkAsignacionNivel3 IS

    PROCEDURE pAsignacionAutomatica(ivIdSolicitud IN NUMBER,ovRetorno out VARCHAR2)
    IS
        BEGIN
            PKASIGNACIONNIVEL2.pAsignacionAutomatica(ivIdSolicitud);
            ovRetorno:='Asignacion Automatica Correcta';
            EXCEPTION
                WHEN OTHERS THEN
                    ovRetorno:=SQLERRM;
        END pAsignacionAutomatica;

    PROCEDURE pAsignacionIndividual(ivIdSolicitud IN NUMBER,ivFuncionario IN VARCHAR2,ovRetorno out VARCHAR2)
    IS
        BEGIN
            PKASIGNACIONNIVEL2.pAsignacionIndividual(ivIdSolicitud,ivFuncionario);
            ovRetorno:='Asignacion individual exitosa';
            EXCEPTION
                WHEN OTHERS THEN
                    ovRetorno:=SQLERRM;
        END pAsignacionIndividual;

END pkAsignacionNivel3;
```

```

create or replace PACKAGE pkAtencionNivel3 IS
  PROCEDURE pValidarAtencion(ivCedulaFuncionario IN VARCHAR2,ivIdSolicitud IN NUMBER, ovRetorno out NUMBER, ovRetorno2 out VARCHAR2);
  PROCEDURE pAtenderSolicitud(ivIdCedulaFuncionario VARCHAR2,ivIdSolicitud IN NUMBER, ivComentario IN VARCHAR2, ovRetorno out VARCHAR2);
  PROCEDURE pAtenderReclamoODano(ivIdCedulaFuncionario IN VARCHAR2,ivIdSolicitud IN NUMBER, ivEstado IN VARCHAR2,ivComentario IN VARCHAR2, ovRetorno out VARCHAR2);
  PROCEDURE pAtenderReclamoODanoAutomatico(ivIdSolicitud IN VARCHAR2, ovRetorno out VARCHAR2 );
END pkAtencionNivel3;
/
create or replace PACKAGE pkAtencionNivel3 IS
  PROCEDURE pValidarAtencion(ivCedulaFuncionario IN VARCHAR2,ivIdSolicitud IN NUMBER, ovRetorno out NUMBER, ovRetorno2 out VARCHAR2);
  PROCEDURE pAtenderSolicitud(ivIdCedulaFuncionario VARCHAR2,ivIdSolicitud IN NUMBER, ivComentario IN VARCHAR2, ovRetorno out VARCHAR2);
  PROCEDURE pAtenderReclamoODano(ivIdCedulaFuncionario IN VARCHAR2,ivIdSolicitud IN NUMBER, ivEstado IN VARCHAR2,ivComentario IN VARCHAR2, ovRetorno out VARCHAR2);
  PROCEDURE pAtenderReclamoODanoAutomatico(ivIdSolicitud IN VARCHAR2, ovRetorno out VARCHAR2 );
END pkAtencionNivel3;
/
create or replace PACKAGE BODY pkAtencionNivel3 IS

  PROCEDURE pValidarAtencion(ivCedulaFuncionario VARCHAR2,ivIdSolicitud IN NUMBER, ovRetorno out NUMBER, ovRetorno2 out VARCHAR2)
  IS
  BEGIN
    ovRetorno:=PKATENCIONNIVEL2.fValidarAtencion(ivCedulaFuncionario,ivIdSolicitud);
    ovRetorno2:='La validacion ha sido exitosa.';

    EXCEPTION
      WHEN OTHERS THEN
        ovRetorno:=SQLERRM;
  END pValidarAtencion;

  PROCEDURE pAtenderSolicitud(ivIdCedulaFuncionario VARCHAR2,ivIdSolicitud IN NUMBER, ivComentario IN VARCHAR2, ovRetorno out VARCHAR2)
  IS
  BEGIN
    PKATENCIONNIVEL2.pAtenderSolicitud(ivIdCedulaFuncionario,ivIdSolicitud, ivComentario);
    ovRetorno:='Se atendio la solicitud de manera exitosa.';

    EXCEPTION
      WHEN OTHERS THEN
        ovRetorno:=SQLERRM;
  END pAtenderSolicitud;

  PROCEDURE pAtenderReclamoODano(ivIdCedulaFuncionario IN VARCHAR2,ivIdSolicitud IN NUMBER, ivEstado IN VARCHAR2,ivComentario IN VARCHAR2, ovRetorno out VARCHAR2)
  IS
  BEGIN
    PKATENCIONNIVEL2.pAtenderReclamoODano(ivIdCedulaFuncionario ,ivIdSolicitud, ivEstado,ivComentario);
    ovRetorno:='Se atendio el reclamo o el daño de manera exitosa.';

    EXCEPTION
      WHEN OTHERS THEN
        ovRetorno:=SQLERRM;
  END pAtenderReclamoODano;

  PROCEDURE pAtenderReclamoODanoAutomatico(ivIdSolicitud IN VARCHAR2,ovRetorno out VARCHAR2)
  IS
  BEGIN
    PKATENCIONNIVEL2.pAtenderReclamoODanoAutomatico(ivIdSolicitud);
    ovRetorno:='Se realizo la atencion automatica de manera exitosa.';

    EXCEPTION
      WHEN OTHERS THEN
        ovRetorno:=SQLERRM;
  END pAtenderReclamoODanoAutomatico;

END pkAtencionNivel3;

```

```

CREATE OR REPLACE PACKAGE pkInicioSesionNivel3 IS

    PROCEDURE pAcceder(ivIdCliente VARCHAR2,ovNombre out VARCHAR2,ovFechaNacimiento out Date,ovDireccion out VARCHAR2, ovTelefono out VARCHAR2,
    ovRetorno out VARCHAR2);
    PROCEDURE pRegistrar(ivIdCliente IN VARCHAR2,ivNombre in VARCHAR2,ivFechaNacimiento Date,ivDireccion in VARCHAR2, ivTelefono in VARCHAR2
    ,ovRetorno out VARCHAR2);

END pkInicioSesionNivel3;
/
CREATE OR REPLACE PACKAGE BODY pkInicioSesionNivel3 IS
-- Insertar
    PROCEDURE pAcceder(ivIdCliente VARCHAR2,ovNombre out VARCHAR2,ovFechaNacimiento out Date,ovDireccion out VARCHAR2, ovTelefono out VARCHAR2,
    ovRetorno out VARCHAR2)
    IS
        ovCliente cliente%rowtype;
    BEGIN

        ovCliente:=PKINICIOSESIONNIVEL2.FACCEDER(ivIdCliente);
        ovNombre :=ovCliente.nombre;
        ovFechaNacimiento :=ovCliente.FechaNacimiento;
        ovDireccion:=ovCliente.Direccion;
        ovTelefono:=ovCliente.Telefono;
        ovRetorno:='Exito';

    EXCEPTION
        WHEN OTHERS THEN
            ovRetorno:=SQLERRM;
    END pAcceder;

    PROCEDURE pRegistrar(ivIdCliente IN VARCHAR2,ivNombre in VARCHAR2,ivFechaNacimiento Date,ivDireccion in VARCHAR2, ivTelefono in VARCHAR2
    , ovRetorno out VARCHAR2)
    IS
    BEGIN
        PKINICIOSESIONNIVEL2.PREGISTRAR(ivIdCliente,ivNombre,ivFechaNacimiento ,ivDireccion , ivTelefono);
        ovRetorno:='Registro exitoso';
    EXCEPTION
        WHEN OTHERS THEN
            ovRetorno:=SQLERRM;
    END pRegistrar;

END pkInicioSesionNivel3;

```

```
| CREATE OR REPLACE PACKAGE pkRegistroNivel3 IS
| PROCEDURE pRegistrarSolicitud(ivIdSolicitud IN NUMBER,ivObservacion IN VARCHAR2,ivIdCliente IN VARCHAR2,ivTipoSolicitud IN NUMBER
| ,ivTipoAnomalia NUMBER,ivIdTipoProducto IN NUMBER,ivIdProducto IN NUMBER, ovRetorno out VARCHAR2);
| END pkRegistroNivel3;
|
CREATE OR REPLACE PACKAGE BODY pkRegistroNivel3 IS
-- Insertar
PROCEDURE pRegistrarSolicitud(ivIdSolicitud IN NUMBER,ivObservacion IN VARCHAR2,ivIdCliente IN VARCHAR2,ivTipoSolicitud IN NUMBER
,ivTipoAnomalia NUMBER,ivIdTipoProducto IN NUMBER,ivIdProducto IN NUMBER, ovRetorno out VARCHAR2)
IS
BEGIN
PKSOLICITUDNIVEL1.pInsertar(ivIdSolicitud , 'Pendiente',ivObservacion,ivIdCliente , '000'
,ivTipoSolicitud,ivTipoAnomalia,ivIdTipoProducto ,ivIdProducto );
PKREGISTRONIVEL2.pRegistrarSolicitud(ivIdSolicitud,ivObservacion,ivIdCliente
,ivTipoSolicitud,ivTipoAnomalia,ivIdTipoProducto ,ivIdProducto);
ovRetorno:='Solicitud registrada correctamente.';
EXCEPTION
WHEN OTHERS THEN
ovRetorno:=SQLERRM;
END pRegistrarSolicitud;
END pkRegistroNivel3;
```

Datos

- [DATOS.sql](#)

```

--Clientes
INSERT INTO CLIENTE VALUES ('1.090.427.627','Sofia','10-ENE-1996','Calle 5 #12-321','12');
INSERT INTO CLIENTE VALUES ('1.090.450.726','Karen','10-FEB-1997','Cra 1 #21-432','13');
INSERT INTO CLIENTE VALUES ('13.490.880','Valentina','10-MAR-1995','Cra 19 #45-253','14');
INSERT INTO CLIENTE VALUES ('1.090.423.115','Daniela','10-ABR-1993','Calle 2 #76-345','15');
INSERT INTO CLIENTE VALUES ('1.093.885.333','Carla','10-MAY-1990','Cra 2 #34-123','15');

--funcionario
INSERT INTO FUNCIONARIO VALUES ('37.279.508','Carlos','23-JUN-1991','Carrera 56A No. 51 - 81','22');
INSERT INTO FUNCIONARIO VALUES ('1.090.441.17','Felipe','21-JUL-1994','Carrera 22 No. 17-31','23');
INSERT INTO FUNCIONARIO VALUES ('1.090.305.888','Alejandro','29-AGO-1986','Carrera 54 No. 68 - 80','24');
INSERT INTO FUNCIONARIO VALUES ('91.530.780','Sebastian','12-OCT-1979','Calle 59 No. 27 - 35','25');
INSERT INTO FUNCIONARIO VALUES ('1.036.627.286','Nicolas','06-NOV-1990','Carrera 8 No. 20 - 59','26');

--Telefonia: Voz
--Internet: Datos
--Integrado: Voz y Datos

--tipoProducto
INSERT INTO TIPOPRODUCTO VALUES (1,'Voz');
INSERT INTO TIPOPRODUCTO VALUES (2,'Datos');
INSERT INTO TIPOPRODUCTO VALUES (3,'Voz');
INSERT INTO TIPOPRODUCTO VALUES (4,'Integrado');
INSERT INTO TIPOPRODUCTO VALUES (5,'Datos');

--producto
INSERT INTO PRODUCTO VALUES (1,'06-ENE-2011',1);
INSERT INTO PRODUCTO VALUES (2,'07-FEB-2012',2);
INSERT INTO PRODUCTO VALUES (3,'08-MAR-2013',3);
INSERT INTO PRODUCTO VALUES (4,'09-ABR-2014',4);
INSERT INTO PRODUCTO VALUES (5,'10-MAY-2015',5);

-- TIPO DE ANOMALIA PUEDEN SER MAS
--tipoAnomalia

INSERT INTO TIPOANOMALIA VALUES (1,'Sin Anomalia');
INSERT INTO TIPOANOMALIA VALUES (2,'Baja Velocidad');
INSERT INTO TIPOANOMALIA VALUES (3,'Mala calidad del servicio');

--tipoSolicitud
--1-5 registro
--5-10 asignacion

--registro: tiene estado de No asignada: creacion o modificacion
--Asignacion: tiene estados de asignada y dentro puede estar: solicitud, retiro, daño y reclamo

--Registro
INSERT INTO TIPOSOLICITUD VALUES (1,'Creacion');
INSERT INTO TIPOSOLICITUD VALUES (2,'Modificacion');
--Asignacion
INSERT INTO TIPOSOLICITUD VALUES (3,'Cancelacion');
INSERT INTO TIPOSOLICITUD VALUES (4,'Daño');
INSERT INTO TIPOSOLICITUD VALUES (5,'Reclamo');

--solicitud
--Una solicitud de registro

INSERT INTO SOLICITUD VALUES (1,'Creacion','esta solicitud no ha sido asignada','01-ENE-2018','01-FEB-2018','1.090.427.627',1,'37.279.508',5,1,1);

--solicitud de asignacion

INSERT INTO SOLICITUD VALUES (5,'Solicitud','Pidiendo un nuevo producto','01-FEB-2018','01-MAR-2018','1.090.450.726',5,'1.090.441.17',6,2,2);

--Relacion de clientes x producto
--Solo se crea esta relacion para una solicitud de TipoSolicitud
--Falta revisar lo de la fecha de ini.

INSERT INTO PRODUCTOSXCLIENTE VALUES (2, '1.090.450.726', '01-FEB-2018', '01-MAR-2018');

COMMIT;

```

Demo App

Repositorio: [FinalBD](#)

¡Gracias!