

PROYECTO I BASES DE DATOS II

APLICACIÓN WEB DE APOYO PARA LAS PYMES EN TIEMPOS DE COVID-19

Instituto Tecnológico de Costa Rica

IC4302 Bases de Datos II - GR 20

Prof. Alberto Shum Chan

Integrantes:

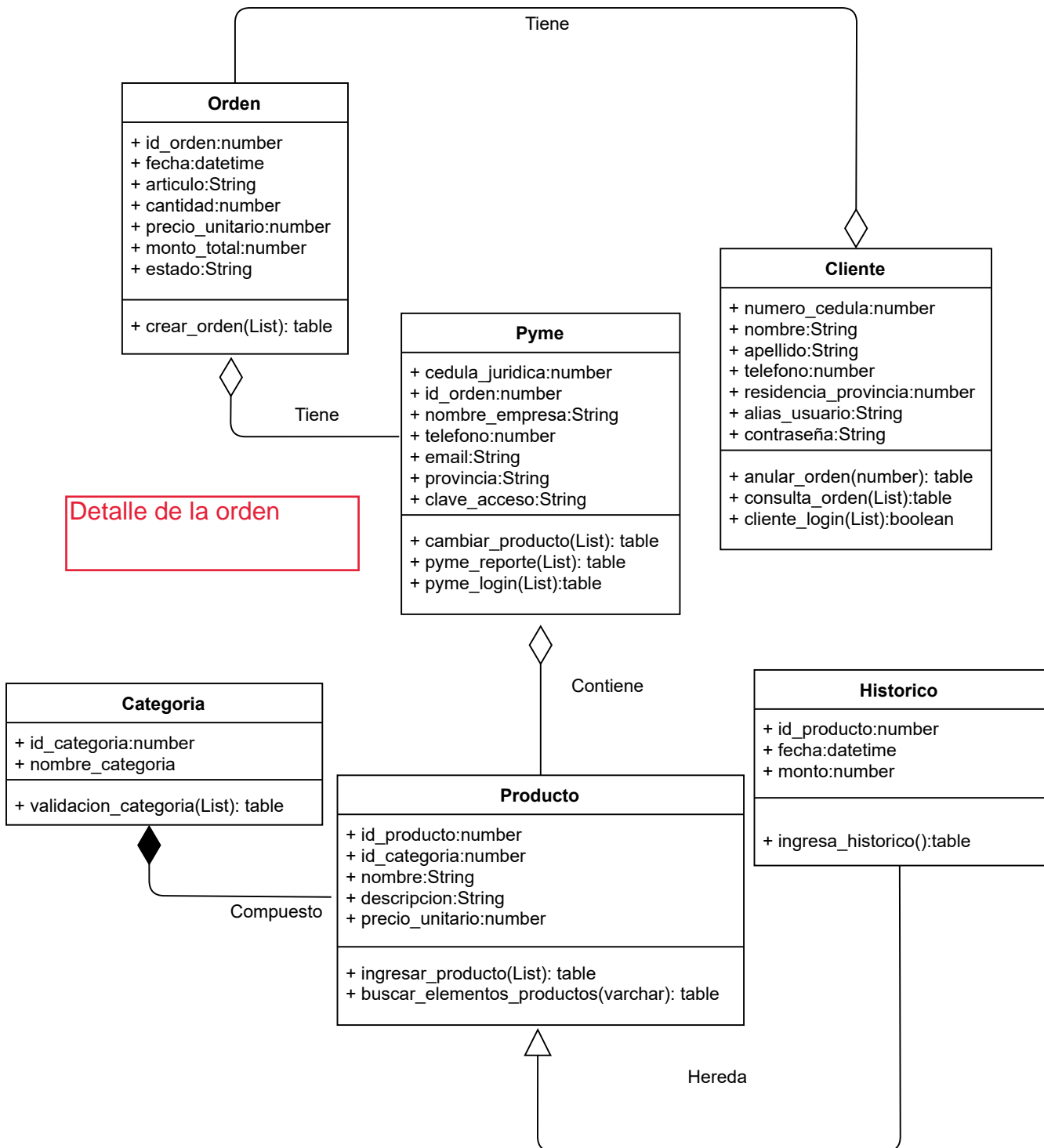
Jonathan Quesada Salas

Alberto Zumbado Abarca



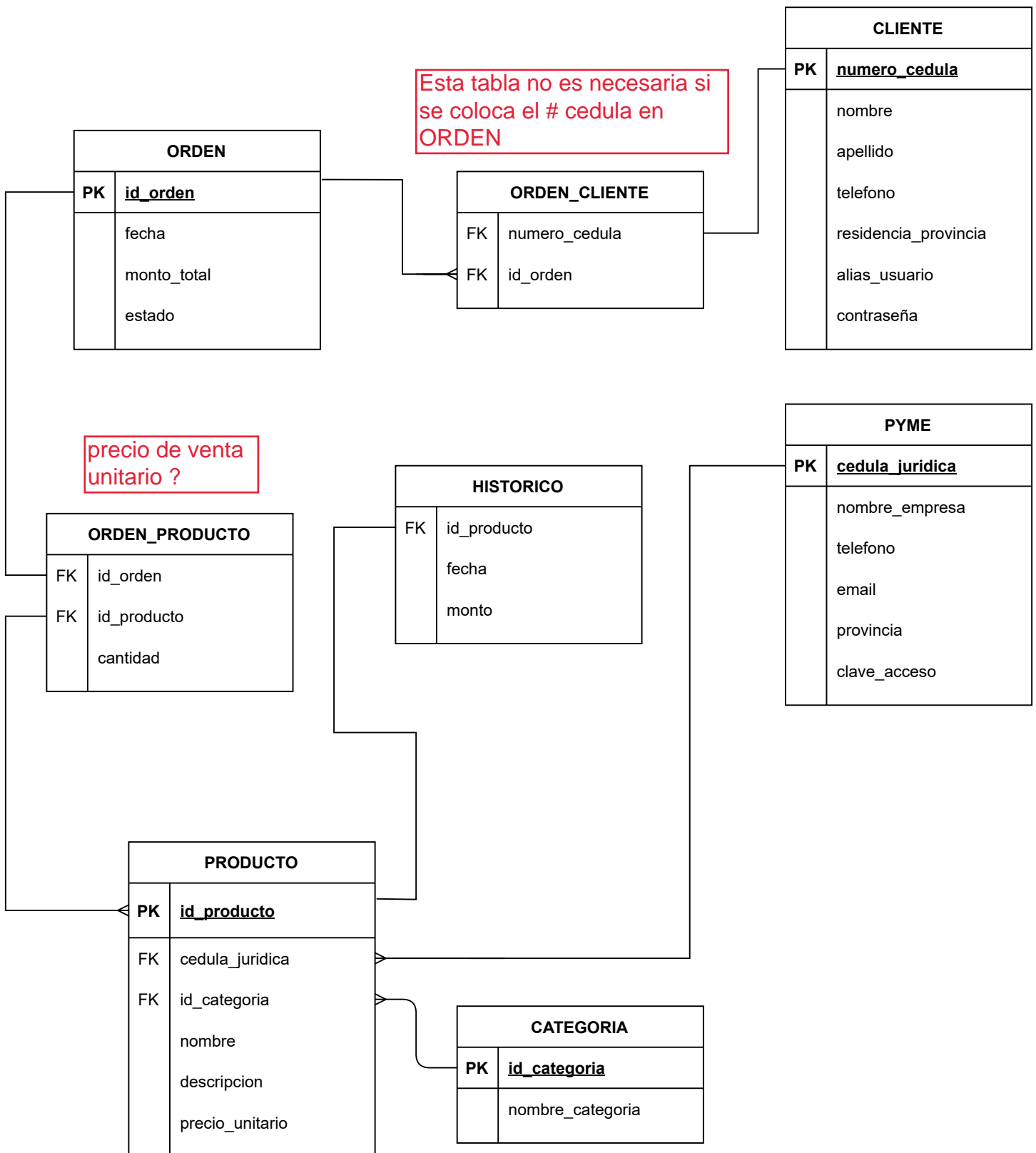
I. MODELADO DE OBJETOS (UML)

3/4



II. MODELO ENTIDAD-RELACION

3/4



III. SCRIPT ORACLE PL/SQL

```
-- -----  
-- PROYECTO 1 - BASES DE DATOS II  
-- -----  
-- Alberto Zumbado Abarca  
-- Jonathan Quesada Salas  
-- -----  
-- SE BORRAN LAS TABLAS  
-- -----  
drop table orden cascade constraints; --  
drop table cliente cascade constraints; --  
drop table producto cascade constraints; --  
drop table pyme cascade constraints; --  
drop table categoria cascade constraints; --  
drop table historico cascade constraints; --  
drop table orden_cliente cascade constraints; --  
drop table orden_producto cascade constraints; --  
  
-- -----  
-- SE CREAN LAS TABLAS  
-- -----  
create table orden(  
    id_orden number not null,  
    fecha date not null,  
    monto_total number not null,  
    estado varchar2(20) not null,  
    constraint orden_pk primary key(id_orden)  
);  
  
create table cliente(  
    numero_cedula varchar2(20) not null,  
    nombre varchar2(20) not null,  
    apellido varchar2(20) not null,
```

```
telefono varchar2(20) unique not null,  
residencia_provincia varchar2(20) not null,  
alias_usuario varchar2(20) unique not null,  
contraseña varchar2(30) not null,  
constraint cliente_pk primary key(numero_cedula)  
);
```

```
create table categoria(  
id_categoria number not null,  
nombre_categoria varchar2(20) not null,  
constraint categoria_pk primary key(id_categoria)  
);
```

```
create table pyme(  
cedula_juridica varchar2(20) not null,  
nombre_empresa varchar2(20) unique not null,  
telefono varchar(20) unique not null,  
email varchar2(50) unique not null,  
clave_acceso varchar2(20) not null,  
constraint pyme_pk primary key (cedula_juridica)  
);
```

```
create table producto(  
id_producto number not null,  
id_categoria number not null,  
pyme_cedula varchar2(20) not null,  
nombre varchar2(20) not null,  
descripcion varchar2(50) not null,  
precio_unitario number not null,  
constraint producto_pk primary key (id_producto),  
constraint pyme_cedula_fk foreign key (pyme_cedula) references  
pyme(cedula_juridica),  
constraint producto_fk foreign key(id_categoria) references categoria(id_categoria)
```

);

```
create table orden_cliente(  
    numero_cedula varchar2(20) not null,  
    id_orden number not null,  
    constraint orden_cliente_orden_fk foreign key (id_orden) references  
orden(id_orden),  
    constraint orden_cliente_cliente_fk foreign key (numero_cedula) references  
cliente(numero_cedula)  
);
```

```
create table orden_producto(  
    id_orden number not null,  
    id_producto number not null,  
    cantidad number not null,  
    constraint orden_producto_orden_fk foreign key (id_orden) references  
orden(id_orden),  
    constraint orden_producto_producto_fk foreign key (id_producto) references  
producto(id_producto)  
);
```

```
create table historico(  
    id_producto number not null,  
    fecha date not null,  
    monto number not null,  
    constraint historico_fk foreign key (id_producto) references producto(id_producto)  
);
```

-- -----

-- SE BORRAN LAS SECUENCAS

-- -----

```
drop sequence orden_sq;  
drop sequence cliente_sq;  
drop sequence producto_sq;
```

```
drop sequence pyme_sq;  
drop sequence categoria_sq;  
drop sequence historico_sq;
```

```
-- -----  
-- SECUENCIAS  
-- -----
```

```
create sequence orden_sq  
  start with 1  
  increment by 1  
  maxvalue 9000  
  minvalue 0;
```

```
create sequence cliente_sq  
  start with 1  
  increment by 1  
  maxvalue 9000  
  minvalue 0;
```

```
create sequence producto_sq  
  start with 1  
  increment by 1  
  maxvalue 9000  
  minvalue 0;
```

```
create sequence pyme_sq  
  start with 1  
  increment by 1  
  maxvalue 9000  
  minvalue 0;
```

```
create sequence categoria_sq
```

```
start with 1
increment by 1
maxvalue 9000
minvalue 0;
```

```
create sequence historico_sq
```

```
start with 1
increment by 1
maxvalue 9000
minvalue 0;
```

```
-- -----
-- INSERTAR DATOS A LAS TABLAS PARA PRUEBAS
-- -----
```

```
insert into orden values (orden_sq.nextval, '01-01-2021', 1000, 'Procesada');
insert into orden values (orden_sq.nextval, '02-02-2021', 2000, 'Procesada');
insert into orden values (orden_sq.nextval, '03-03-2021', 3000, 'Anulada');
insert into orden values (orden_sq.nextval, '04-04-2021', 4000, 'Procesada');
insert into orden values (orden_sq.nextval, '05-05-2021', 5000, 'Anulada');
insert into orden values (orden_sq.nextval, '05-05-2021', 5000, 'Procesada');
insert into orden values (orden_sq.nextval, '05-05-2021', 5000, 'Procesada');
```

```
insert into cliente values ('1', 'Alberto', 'Zumbado', 81238327, 'Alajuela', 'Beto',
'KAS142');
```

```
insert into cliente values ('2', 'Aurelio', 'Sanabria', 81264788, 'Puntarenas', 'Aurelion
Sol', 'qiA14');
```

```
insert into cliente values ('3', 'Jeaustin', 'Obando', 81237327, 'Cartago', 'Suero',
'ASnc12');
```

```
insert into cliente values ('4', 'Maria', 'Mora', 81235427, 'Heredia', 'Maria', 'AKHD14');
```

```
insert into cliente values ('5', 'Andres', 'Viquez', 81941327, 'San Jose', 'Andre',
'POJB1');
```

```
insert into categoria values (categoria_sq.nextval, 'Salud');
```

```
insert into categoria values (categoria_sq.nextval, 'Educacion');
```


insert into categoria values (categoria_sq.nextval, 'Reposteria');

insert into categoria values (categoria_sq.nextval, 'Deporte');

insert into categoria values (categoria_sq.nextval, 'Ejercicio');

insert into pyme values (pyme_sq.nextval, 'PALI', '2770142', 'pali@gmail.com', 'SNQ14A');

insert into pyme values (pyme_sq.nextval, 'masXmenos', '2740142', 'masXmenos@gmail.com', 'CAQ14A');

insert into pyme values (pyme_sq.nextval, 'PERIMERCADOS', '2570142', 'perimercados@gmail.com', 'POQ14A');

insert into pyme values (pyme_sq.nextval, 'Ekono', '2770172', 'Ekono@gmail.com', 'MASA4A');

insert into pyme values (pyme_sq.nextval, 'Pricemark', '2756142', 'pricemark@gmail.com', 'VYQY78');

insert into producto values (producto_sq.nextval, 3, 1, 'Pastel', 'Pastel de chocolate', 1000);

insert into producto values (producto_sq.nextval, 5, 2, 'Pesas', 'Pesas de 5 kilogramos', 1000);

insert into producto values (producto_sq.nextval, 4, 3, 'Balon', 'Balon de futbol', 1000);

insert into producto values (producto_sq.nextval, 1, 4, 'Pastillas', 'Pastillas de dormir', 1000);

insert into producto values (producto_sq.nextval, 2, 5, 'Libros', 'Libros de lectura de niños', 1000);

insert into orden_cliente values ('1',1);

insert into orden_cliente values ('2',3);

insert into orden_cliente values ('3',2);

insert into orden_cliente values ('4',5);

insert into orden_cliente values ('5',4);

insert into orden_producto values (1,1,1);

insert into orden_producto values (2,2,1);

insert into orden_producto values (3,3,1);

insert into orden_producto values (4,4,1);

insert into orden_producto values (5,5,1);

```
insert into historico values (1, '01-01-2020', 800);
insert into historico values (2, '02-02-2020', 700);
insert into historico values (3, '03-03-2020', 900);
insert into historico values (4, '04-04-2020', 500);
insert into historico values (5, '05-05-2020', 950);
```

```
-- -----
-- DEFINICION DEL PAQUETE PARA CONSULTAS CRUD DE MODIFICACION
-- -----
```

```
create or replace package consultas_modificacion as
```

```
-- CRUD ORDEN
```

```
procedure insert_orden
```

```
(o_fecha date,
 o_monto_total number,
 o_estado varchar2);
```

```
procedure update_orden
```

```
(o_id_orden number,
 o_fecha date,
 o_monto_total number,
 o_estado varchar2);
```

```
procedure delete_orden (o_id_orden number);
```

Las ordenes no se borran, se cancelan.

```
-- CRUD CLIENTE
```

```
procedure insert_cliente
```

```
(c_numero_cedula varchar2,
 c_nombre varchar2,
 c_apellido varchar2,
 c_telefono varchar2,
```

```
c_residencia_provincia varchar2,  
c_alias_usuario varchar2,  
c_contraseña varchar2);
```

```
procedure update_cliente  
    (c_numero_cedula varchar2,  
    c_nombre varchar2,  
    c_apellido varchar2,  
    c_telefono varchar2,  
    c_residencia_provincia varchar2,  
    c_alias_usuario varchar2,  
    c_contraseña varchar2);
```

```
procedure delete_cliente(c_numero_cedula varchar2);
```

-- CRUD PRODUCTO

```
procedure insert_producto  
    (p_id_producto number,  
    p_id_categoria number,  
    p_id_pyme varchar2,  
    p_nombre varchar2,  
    p_descripcion varchar2,  
    p_precio_unitario number);
```

```
procedure update_producto  
    (p_id_producto number,  
    p_id_categoria number,  
    p_id_pyme varchar2,  
    p_nombre varchar2,  
    p_descripcion varchar2,  
    p_precio_unitario number);
```

```
procedure delete_producto (p_id_producto number);
```

```
-- CRUD PYME
```

```
procedure insert_pyme  
    (py_cedula_juridica varchar2,  
    py_nombre_empresa varchar2,  
    py_telefono varchar2,  
    py_email varchar2,  
    py_clave_acceso varchar2);
```

```
procedure update_pyme  
    (py_cedula_juridica varchar2,  
    py_nombre_empresa varchar2,  
    py_telefono varchar2,  
    py_email varchar2,  
    py_clave_acceso varchar2);
```

```
procedure delete_pyme (py_cedula_juridica varchar2);
```

```
-- CRUD CATEGORIA
```

```
procedure insert_categoria  
    (c_nombre_categoria varchar2);
```

```
procedure update_categoria  
    (c_id_categoria number,  
    c_nombre_categoria varchar2);
```

```
procedure delete_categoria (c_id_categoria number);
```

```
-- CRUD ORDEN_PRODUCTO
```

```
procedure insert_orden_producto  
    (h_id_orden number,
```

```
h_id_producto number,  
h_id_cantidad number);
```

```
procedure update_orden_producto
```

```
(h_id_orden number,  
h_id_producto number,  
h_id_cantidad number);
```

```
procedure delete_orden_producto (h_id_orden number);
```

```
-- CRUD HISTORICO
```

```
procedure insert_historico
```

```
(h_id_producto number,  
h_fecha date,  
h_monto number);
```

```
procedure update_historico
```

```
(h_id_producto number,  
h_fecha date,  
h_monto number);
```

```
procedure delete_historico(h_id_orden number);
```

```
-- CRUD ORDEN_CLIENTE
```

```
procedure insert_orden_cliente
```

```
(oc_numero_cedula varchar2,  
oc_id_orden number);
```

```
procedure update_orden_cliente
```

```
(oc_numero_cedula varchar2,  
oc_id_orden number);
```

```

procedure delete_orden_cliente
    (oc_numero_cedula number);

end consultas_modificacion;

/

-----
-- CUERPO DEL PAQUETE
-----

create or replace package body consultas_modificacion as

    -- CRUD ORDEN
    procedure insert_orden
        (o_fecha date,
        o_monto_total number,
        o_estado varchar2) is
    begin
        insert into orden(id_orden, fecha, monto_total, estado)
        values (orden_sq.nextval, o_fecha, o_monto_total, o_estado);
        commit;
    exception
        when no_data_found then
            DBMS_OUTPUT.PUT_LINE('La orden no existe');
        when too_many_rows then
            DBMS_OUTPUT.PUT_LINE('La orden esta repetida');
        when value_error then
            DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
        when others then
            DBMS_OUTPUT.PUT_LINE('ERROR AL INSERTAR LA ORDEN');
            DBMS_OUTPUT.PUT_LINE(sqlerrm);
            DBMS_OUTPUT.PUT_LINE(sqlcode);
    end insert_orden;

```

Las excepciones no corresponden a lo que puede pasar

```

procedure update_orden
(o_id_orden number,
o_fecha date,
o_monto_total number,
o_estado varchar2) is
begin
    update orden set id_orden = o_id_orden, fecha = o_fecha, monto_total =
o_monto_total, estado = o_estado
    where id_orden = o_id_orden;
    commit;
exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('La orden no existe');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('La orden esta repetida');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR AL ACTUALIZAR LA ORDEN');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end update_orden;

```

procedure delete_orden

```

(o_id_orden number) is
begin
    delete from orden where id_orden = o_id_orden;
    commit;
exception
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR: No se pudo eliminar la orden');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);

```

```

        DBMS_OUTPUT.PUT_LINE(sqlcode);
    end delete_orden;

-- CRUD CLIENTE

procedure insert_cliente
    (c_numero_cedula varchar2,
    c_nombre varchar2,
    c_apellido varchar2,
    c_telefono varchar2,
    c_residencia_provincia varchar2,
    c_alias_usuario varchar2,
    c_contraseña varchar2) is
begin
    insert into cliente(numero_cedula, nombre, apellido, telefono,
    residencia_provincia, alias_usuario, contraseña)

    values (c_numero_cedula, c_nombre, c_apellido, c_telefono,
    c_residencia_provincia, c_alias_usuario, c_contraseña);

    commit;
exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('El cliente no existe');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('El cliente esta repetido');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR AL INSERTAR EI CLIENTE');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end insert_cliente;

procedure update_cliente
    (c_numero_cedula varchar2,

```



```

c_nombre varchar2,
c_apellido varchar2,
c_telefono varchar2,
c_residencia_provincia varchar2,
c_alias_usuario varchar2,
c_contraseña varchar2) is
begin
    update cliente set numero_cedula = c_numero_cedula, nombre = c_nombre,
apellido = c_apellido,
    telefono = c_telefono, residencia_provincia = c_residencia_provincia,
alias_usuario = c_alias_usuario, contraseña = c_contraseña
    where numero_cedula = c_numero_cedula;
    commit;
exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('El cliente no existe');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('El cliente esta repetido');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR AL ACTUALIZAR EI CLIENTE');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end update_cliente;

```

procedure delete_cliente

```

(c_numero_cedula varchar2) is
begin
    delete from cliente where numero_cedula = c_numero_cedula;
    commit;
exception
    when others then

```

```
        DBMS_OUTPUT.PUT_LINE('ERROR: No se pudo eliminar el cliente');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end delete_cliente;
```

-- CRUD PRODUCTO

```
procedure insert_producto
(p_id_producto number,
p_id_categoria number,
p_id_pyme varchar2,
p_nombre varchar2,
p_descripcion varchar2,
p_precio_unitario number) is

begin
    insert into producto(id_producto, id_categoria, pyme_cedula, nombre,
descripcion, precio_unitario)
        values (p_id_producto, p_id_categoria, p_id_pyme, p_nombre, p_descripcion,
p_precio_unitario);
    commit;
exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('El producto no existe');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('El producto esta repetido');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR AL INSERTAR EI PRODUCTO');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end insert_producto;
```

```

procedure update_producto
(
    p_id_producto number,
    p_id_categoria number,
    p_id_pyme varchar2,
    p_nombre varchar2,
    p_descripcion varchar2,
    p_precio_unitario number) is
begin
    update producto set id_producto = p_id_producto, id_categoria =
p_id_categoria, pyme_cedula = p_id_pyme,
    nombre = p_nombre, descripcion = p_descripcion, precio_unitario =
p_precio_unitario
    where id_producto = p_id_producto;
    commit;
exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('El producto no existe');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('El producto esta repetido');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR AL ACTUALIZAR EI PRODUCTO');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end update_producto;

```

```

procedure delete_producto
(
    p_id_producto number) is
begin
    delete from producto where id_producto = p_id_producto;
    commit;
exception

```

```

        when others then
            DBMS_OUTPUT.PUT_LINE('ERROR: No se pudo eliminar el producto');
            DBMS_OUTPUT.PUT_LINE(sqlerrm);
            DBMS_OUTPUT.PUT_LINE(sqlcode);
        end delete_producto;

-- CRUD PYME
procedure insert_pyme
    (py_cedula_juridica varchar2,
    py_nombre_empresa varchar2,
    py_telefono varchar2,
    py_email varchar2,
    py_clave_acceso varchar2) is

    begin
        insert into pyme(cedula_juridica, nombre_empresa, telefono, email,
        clave_acceso)

            values (py_cedula_juridica , py_nombre_empresa, py_telefono, py_email,
        py_clave_acceso);

        commit;
    exception
        when no_data_found then
            DBMS_OUTPUT.PUT_LINE('El pyme no existe');
        when too_many_rows then
            DBMS_OUTPUT.PUT_LINE('El pyme esta repetido');
        when value_error then
            DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
        when others then
            DBMS_OUTPUT.PUT_LINE('ERROR AL INSERTAR EI PYME');
            DBMS_OUTPUT.PUT_LINE(sqlerrm);
            DBMS_OUTPUT.PUT_LINE(sqlcode);
        end insert_pyme;

```

procedure update_pyme

(py_cedula_juridica varchar2,
py_nombre_empresa varchar2,
py_telefono varchar2,
py_email varchar2,
py_clave_acceso varchar2) is

begin

update pyme set cedula_juridica = py_cedula_juridica, nombre_empresa =
py_nombre_empresa, telefono = py_telefono, email = py_email, clave_acceso =
py_clave_acceso where cedula_juridica = py_cedula_juridica;

commit;

exception

when no_data_found then

DBMS_OUTPUT.PUT_LINE('El pyme no existe');

when too_many_rows then

DBMS_OUTPUT.PUT_LINE('El pyme esta repetido');

when value_error then

DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');

when others then

DBMS_OUTPUT.PUT_LINE('ERROR AL ACTUALIZAR EI PYME');

DBMS_OUTPUT.PUT_LINE(sqlerrm);

DBMS_OUTPUT.PUT_LINE(sqlcode);

end update_pyme;

procedure delete_pyme

(py_cedula_juridica varchar2) is

begin

delete from pyme where cedula_juridica like py_cedula_juridica;

commit;

exception

when others then

DBMS_OUTPUT.PUT_LINE('ERROR: No se pudo eliminar el PYME');

```
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end delete_pyme;
```

-- CRUD CATEGORIA

procedure insert_categoria

(c_nombre_categoria varchar2) is

begin

insert into categoria(id_categoria, nombre_categoria)

values (categoria_sq.nextval, c_nombre_categoria);

commit;

exception

when no_data_found then

DBMS_OUTPUT.PUT_LINE('La categoria no existe');

when too_many_rows then

DBMS_OUTPUT.PUT_LINE('La categoria esta repetido');

when value_error then

DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');

when others then

DBMS_OUTPUT.PUT_LINE('ERROR AL INSERTAR LA CATEGORIA');

DBMS_OUTPUT.PUT_LINE(sqlerrm);

DBMS_OUTPUT.PUT_LINE(sqlcode);

end insert_categoria;

procedure update_categoria

(c_id_categoria number,

c_nombre_categoria varchar2) is

begin

update categoria set id_categoria = c_id_categoria, nombre_categoria =
c_nombre_categoria where id_categoria = c_id_categoria;

commit;

```

exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('La categoria no existe');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('La categoria esta repetido');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR AL ACTUALIZAR LA CATEGORIA');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end update_categoria;

```

```

procedure delete_categoria
(c_id_categoria number) is
begin
    delete from categoria where id_categoria = c_id_categoria;
    commit;
exception
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR: No se pudo eliminar la categoria');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end delete_categoria;

```

-- CRUD ORDEN_PRODUCTO

```

procedure insert_orden_producto
(h_id_orden number,
 h_id_producto number,
 h_id_cantidad number) is

begin

```

```
insert into orden_producto(id_orden, id_producto,cantidad) values (h_id_orden,  
h_id_producto, h_id_cantidad);
```

```
commit;
```

```
exception
```

```
when no_data_found then
```

```
    DBMS_OUTPUT.PUT_LINE('La relacion de orden-producto no existe');
```

```
when too_many_rows then
```

```
    DBMS_OUTPUT.PUT_LINE('La relacion de orden-producto esta repetida');
```

```
when value_error then
```

```
    DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
```

```
when others then
```

```
    DBMS_OUTPUT.PUT_LINE('ERROR AL INSERTAR LA RELACION ENTRE  
ORDEN-PRODUCTO');
```

```
    DBMS_OUTPUT.PUT_LINE(sqlerrm);
```

```
    DBMS_OUTPUT.PUT_LINE(sqlcode);
```

```
end insert_orden_producto;
```

```
procedure update_orden_producto
```

```
(h_id_orden number,
```

```
h_id_producto number,
```

```
h_id_cantidad number) is
```

```
begin
```

```
update orden_producto set id_orden = h_id_orden, id_producto =  
h_id_producto, cantidad = h_id_cantidad where id_orden = h_id_orden;
```

```
commit;
```

```
exception
```

```
when no_data_found then
```

```
    DBMS_OUTPUT.PUT_LINE('La relacion de orden-producto no existe');
```

```
when too_many_rows then
```

```
    DBMS_OUTPUT.PUT_LINE('La relacion de orden-producto esta repetida');
```

```
when value_error then
```

```
    DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
```



```

        when others then

            DBMS_OUTPUT.PUT_LINE('ERROR AL ACTUALIZAR LA RELACION
ENTRE ORDEN-PRODUCTO');

            DBMS_OUTPUT.PUT_LINE(sqlerrm);

            DBMS_OUTPUT.PUT_LINE(sqlcode);

        end update_orden_producto;

procedure delete_orden_producto
(h_id_orden number) is
begin
    delete from orden_producto where id_orden = h_id_orden;
    commit;
exception
    when others then

        DBMS_OUTPUT.PUT_LINE('ERROR: No se pudo eliminar la relacion entre
orden-producto');

        DBMS_OUTPUT.PUT_LINE(sqlerrm);

        DBMS_OUTPUT.PUT_LINE(sqlcode);

    end delete_orden_producto;

```

-- **CRUD HISTORICO**

Es una tabla de auditoria. Los cambios solo ingresan por medio del TRIGGER

```

procedure insert_historico
(h_id_producto number,
h_fecha date,
h_monto number) is
begin
    insert into historico(id_producto, fecha, monto)
    values (h_id_producto, h_fecha, h_monto);
    commit;
exception
    when no_data_found then

        DBMS_OUTPUT.PUT_LINE('El historico no existe');

    when too_many_rows then

```

```

        DBMS_OUTPUT.PUT_LINE('El historico esta repetido');
when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
when others then
        DBMS_OUTPUT.PUT_LINE('ERROR AL INSERTAR HISTORICO');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end insert_historico;

```

procedure **update_historico**

Es una tabla de auditoria.

```

(h_id_producto number,
h_fecha date,
h_monto number) is
begin
        update historico set id_producto = h_id_producto, fecha = h_fecha, monto =
h_monto where id_producto = h_id_producto;
        commit;
exception
when no_data_found then
        DBMS_OUTPUT.PUT_LINE('El historico no existe');
when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('El historico esta repetido');
when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
when others then
        DBMS_OUTPUT.PUT_LINE('ERROR AL ACTUALIZAR HISTORICO');
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end update_historico;

```

procedure **delete_historico**

```

(h_id_orden number) is
begin

```

```

delete from historico where id_producto = h_id_orden;

commit;

exception

when others then

    DBMS_OUTPUT.PUT_LINE('ERROR: No se pudo eliminar el historico');

    DBMS_OUTPUT.PUT_LINE(sqlerrm);

    DBMS_OUTPUT.PUT_LINE(sqlcode);

end delete_historico;

```

-- ORDEN ORDEN_CLIENTE

```

procedure insert_orden_cliente
(oc_numero_cedula varchar2,
 oc_id_orden number) is
begin
    insert into orden_cliente(numero_cedula, id_orden) values (oc_numero_cedula,
oc_id_orden);

    commit;

exception

when no_data_found then

    DBMS_OUTPUT.PUT_LINE('La relacion de orden-cliente no existe');

when too_many_rows then

    DBMS_OUTPUT.PUT_LINE('La relacion de orden-cliente esta repetida');

when value_error then

    DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');

when others then

    DBMS_OUTPUT.PUT_LINE('ERROR AL INSERTAR LA RELACION ENTRE
ORDEN-CLIENTE');

    DBMS_OUTPUT.PUT_LINE(sqlerrm);

    DBMS_OUTPUT.PUT_LINE(sqlcode);

end insert_orden_cliente;

```

procedure update_orden_cliente

```

(oc_numero_cedula varchar2,
oc_id_orden number) is
begin
    update orden_cliente set numero_cedula = oc_numero_cedula, id_orden =
oc_id_orden where numero_cedula = oc_numero_cedula;

    commit;
exception
    when no_data_found then

        DBMS_OUTPUT.PUT_LINE('La relacion de orden-cliente no existe');
    when too_many_rows then

        DBMS_OUTPUT.PUT_LINE('La relacion de orden-cliente esta repetida');
    when value_error then

        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then

        DBMS_OUTPUT.PUT_LINE('ERROR AL ACTUALIZAR LA RELACION
ENTRE ORDEN-CLIENTE');

        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end update_orden_cliente;

```

```

procedure delete_orden_cliente
(oc_numero_cedula number) is
begin
    delete from orden_cliente where numero_cedula = oc_numero_cedula;

    commit;
exception
    when others then

        DBMS_OUTPUT.PUT_LINE('ERROR: No se pudo eliminar la relacion entre
orden-cliente');

        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        DBMS_OUTPUT.PUT_LINE(sqlcode);
end delete_orden_cliente;

```

```
end consultas_modificacion;
```

```
/
```

```
-- -----
```

```
-- DEFINICION DEL PAQUETE PARA CONSULTAS CRUD DE SELECCION
```

```
-- -----
```

```
create or replace package consulta_datos is
```

```
-- Seleccionar orden
```

```
procedure select_orden(id orden.id_orden%TYPE ,c1 OUT SYS_REFCURSOR);
```

```
-- Seleccionar cliente
```

```
procedure select_cliente(id cliente.numero_cedula%TYPE, c1 OUT  
SYS_REFCURSOR);
```

```
-- Seleccionar producto
```

```
procedure select_producto(id producto.id_producto%TYPE, c1 OUT  
SYS_REFCURSOR);
```

```
-- Seleccionar pyme
```

```
procedure select_pyme(id pyme.cedula_juridica%TYPE, c1 OUT  
SYS_REFCURSOR);
```

```
-- Seleccionar categoria
```

```
procedure select_categoria(id categoria.id_categoria%TYPE ,c1 OUT  
SYS_REFCURSOR);
```

```
-- Seleccionar historico
```

```
procedure select_historico(id historico.id_producto%TYPE ,c1 OUT  
SYS_REFCURSOR);
```

```
-- Seleccionar orden_cliente
```

```
procedure select_orden_cliente(id_orden orden_cliente.id_orden%TYPE, id_cliente  
orden_cliente.numero_cedula%TYPE ,c1 OUT SYS_REFCURSOR);
```

```
-- Seleccionar orden_producto
```

```
procedure select_orden_producto(id_orden orden_producto.id_orden%TYPE,  
id_producto orden_producto.id_producto%TYPE ,c1 OUT SYS_REFCURSOR);
```

```
end consulta_datos;
```

```
/
```

```
-- -----
```

```
-- CUERPO DEL PAQUETE
```

```
-- -----
```

```
create or replace package body consulta_datos is
```

-- ORDEN

procedure select_orden(id orden.id_orden%TYPE ,c1 OUT SYS_REFCURSOR) as

begin

open c1 for

select * from orden where orden.id_orden = id ;

exception

when no_data_found then

DBMS_OUTPUT.PUT_LINE('Orden no existe');

when too_many_rows then

DBMS_OUTPUT.PUT_LINE('Orden esta repetida');

when value_error then

DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');

when others then

DBMS_OUTPUT.PUT_LINE('ERROR 7n7');

end select_orden;

-- CLIENTE

procedure select_cliente(id cliente.numero_cedula%TYPE, c1 OUT
SYS_REFCURSOR) as

begin

open c1 for

select * from cliente where cliente.numero_cedula = id;

exception

when no_data_found then

DBMS_OUTPUT.PUT_LINE('Cliente no existe');

when too_many_rows then

DBMS_OUTPUT.PUT_LINE('Cliente esta repetido');

when value_error then

DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');

when others then

```

        DBMS_OUTPUT.PUT_LINE('ERROR 7n7');
    end select_cliente;

-- PRODUCTO

    procedure select_producto(id producto.id_producto%TYPE, c1 OUT
SYS_REFCURSOR) as
    begin

        open c1 for
        select * from producto where producto.id_producto = id;
    exception
        when no_data_found then
            DBMS_OUTPUT.PUT_LINE('Producto no existe');
        when too_many_rows then
            DBMS_OUTPUT.PUT_LINE('Producto esta repetido');
        when value_error then
            DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
        when others then
            DBMS_OUTPUT.PUT_LINE('ERROR 7n7');
    end select_producto;

-- PYME

    procedure select_pyme(id pyme.cedula_juridica%TYPE, c1 OUT
SYS_REFCURSOR) as
    begin

        open c1 for
        select * from pyme where pyme.cedula_juridica = id;
    exception
        when no_data_found then
            DBMS_OUTPUT.PUT_LINE('PYME no existe');
        when too_many_rows then
            DBMS_OUTPUT.PUT_LINE('PYME esta repetido');

```

```

        when value_error then
            DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
        when others then
            DBMS_OUTPUT.PUT_LINE('ERROR 7n7');
    end select_pyme;

-- CATEGORIA

procedure select_categoria(id categoria.id_categoria%TYPE ,c1 OUT
SYS_REFCURSOR) as
begin

    open c1 for
        select * from categoria where categoria.id_categoria = id;
exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('Categoria no existe');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('Categoria esta repetida');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR 7n7');
end select_categoria;

-- HISTORICO

procedure select_historico(id historico.id_producto%TYPE ,c1 OUT
SYS_REFCURSOR) as
begin

    open c1 for
        select * from historico where historico.id_producto = id;
exception
    when no_data_found then

```



```

        DBMS_OUTPUT.PUT_LINE('Historico no existe');
when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('Historico esta repetido');
when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
when others then
        DBMS_OUTPUT.PUT_LINE('ERROR 7n7');
end select_historico;

```

-- ORDEN_CLIENTE

```

procedure select_orden_cliente(id_orden orden_cliente.id_orden%TYPE, id_cliente
orden_cliente.numero_cedula%TYPE ,c1 OUT SYS_REFCURSOR) as

```

```

begin

```

```

        open c1 for
        select * from orden_cliente where orden_cliente.id_orden = id_orden and
orden_cliente.numero_cedula = id_cliente;

```

```

exception

```

```

        when no_data_found then
                DBMS_OUTPUT.PUT_LINE('Orden-Cliente no existe');
        when too_many_rows then
                DBMS_OUTPUT.PUT_LINE('Orden-Cliente esta repetido');
        when value_error then
                DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
        when others then
                DBMS_OUTPUT.PUT_LINE('ERROR 7n7');
end select_orden_cliente;

```

-- ORDEN_PRODUCTO

```

procedure select_orden_producto(id_orden orden_producto.id_orden%TYPE,
id_producto orden_producto.id_producto%TYPE ,c1 OUT SYS_REFCURSOR) as

```

```

begin

```

```

        open c1 for
        select * from orden_producto where orden_producto.id_orden = id_orden and
orden_producto.id_producto = id_producto;
    exception
        when no_data_found then
            DBMS_OUTPUT.PUT_LINE('Orden-Producto no existe');
        when too_many_rows then
            DBMS_OUTPUT.PUT_LINE('Orden-Producto esta repetido');
        when value_error then
            DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
        when others then
            DBMS_OUTPUT.PUT_LINE('ERROR 7n7');
    end select_orden_producto;

end consulta_datos;

/

```

```

-----
-- PROCEDIMIENTOS Y FUNCIONES
-----

```

Por qué no están dentro de un PACKAGE?

```

create or replace procedure anular_orden (p_id_orden number) as
begin
    update orden set estado = 'Anulada' where p_id_orden = id_orden;
exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('La orden no existe');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('La orden esta repetida');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR 7n7');

```

end;

/

-- -----

create or replace procedure consultar_orden (c_numero_cedula varchar2, estado
number, c1 IN OUT SYS_REFCURSOR) as

begin

if estado = 0 then

open c1 for

select * from orden inner join orden_cliente

on orden.id_orden = orden_cliente.id_orden

inner join cliente on cliente.numero_cedula = orden_cliente.numero_cedula

where c_numero_cedula = orden_cliente.numero_cedula and orden.estado =
'Anulada';

elsif estado = 1 then

open c1 for

select * from orden inner join orden_cliente

on orden.id_orden = orden_cliente.id_orden

inner join cliente on cliente.numero_cedula = orden_cliente.numero_cedula

where c_numero_cedula = orden_cliente.numero_cedula and orden.estado =
'Procesada';

elsif estado = 2 then

open c1 for

select * from orden inner join orden_cliente

on orden.id_orden = orden_cliente.id_orden

inner join cliente on cliente.numero_cedula = orden_cliente.numero_cedula

where c_numero_cedula = orden_cliente.numero_cedula;

end if;

exception

when no_data_found then

DBMS_OUTPUT.PUT_LINE('La orden no existe, nada que consultar');

when too_many_rows then

DBMS_OUTPUT.PUT_LINE('Ejecucion repetida');

```

        when value_error then

            DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');

        when others then

            DBMS_OUTPUT.PUT_LINE('ERROR 7n7');

end;

/

-----

create or replace function buscar_elementos_productos(descripcion_producto
varchar2) return number

is
    id_producto_parametro number;
    condicion_auxiliar number;

begin

    select count(*) into condicion_auxiliar from producto indice where
descripcion_producto = indice.nombre or indice.descripcion like concat('%',
concat(descripcion_producto, '%'));

    if condicion_auxiliar > 1 then return -1;

    end if;

    if condicion_auxiliar = 1 then

        select id_producto into id_producto_parametro from producto indice where
descripcion_producto = indice.nombre or indice.descripcion like concat('%',
concat(descripcion_producto, '%'));

        return id_producto_parametro;

    end if;

    return 0;

exception

    when no_data_found then

        DBMS_OUTPUT.PUT_LINE('Sin elementos que buscar');

    when too_many_rows then

        DBMS_OUTPUT.PUT_LINE('Ejecucion repetida');

    when value_error then

        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');

    when others then

```

```

        DBMS_OUTPUT.PUT_LINE('ERROR 7n7');

    end;

/

-----

create or replace procedure pyme_reporte (p_pyme_cedula varchar2, mes number, c1
OUT SYS_REFCURSOR) as
    articulos_vendidos number;
    pmonto_total number;
begin
    select sum(cantidad) into articulos_vendidos from orden inner join orden_producto
on orden.id_orden = orden_producto.id_orden inner join producto
    on orden_producto.id_producto = producto.id_producto
    inner join pyme on producto.pyme_cedula = pyme.cedula_juridica
    where pyme.cedula_juridica = p_pyme_cedula and to_char(orden.fecha, 'MM') =
mes and orden.estado like 'Procesada';

    select sum(orden_producto.cantidad * producto.precio_unitario) into pmonto_total
from orden inner join orden_producto on orden.id_orden = orden_producto.id_orden
inner join producto
    on orden_producto.id_producto = producto.id_producto
    inner join pyme on producto.pyme_cedula = pyme.cedula_juridica
    where pyme.cedula_juridica = p_pyme_cedula and to_char(orden.fecha, 'MM') =
mes and orden.estado like 'Procesada';

    open c1 for

    select mes as reporte_mes, articulos_vendidos as articulos_vendidos_mes,
    pmonto_total as monto_total from dual;

exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('Sin reportes que hacer');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('Ejecucion repetida');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');

```

```

        when others then

            DBMS_OUTPUT.PUT_LINE('ERROR 7n7');

end;

/

-----

create or replace function cliente_login(id cliente.alias_usuario%TYPE , pass
cliente.contraseña%type)

    return varchar2 as

    resultado number;

begin

    select count(*) into resultado from cliente where cliente.alias_usuario = id and
cliente.contraseña = pass;

    if resultado = 1 then

        return 'true';

    else

        return 'false';

    end if;

exception

    when no_data_found then

        DBMS_OUTPUT.PUT_LINE('El registro de cliente no existe');

    when too_many_rows then

        DBMS_OUTPUT.PUT_LINE('Ejecucion repetida');

    when value_error then

        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');

    when others then

        DBMS_OUTPUT.PUT_LINE('ERROR 7n7');

end;

/

-----

create or replace function pyme_login(cedula pyme.cedula_juridica%TYPE , pass
pyme.clave_acceso%type)

```

```

        return varchar2 as
        resultado number;
begin
    select count(*) into resultado from pyme where pyme.cedula_juridica = cedula and
pyme.clave_acceso = pass;

    if resultado = 1 then
        return 'true';
    else
        return 'false';
    end if;

exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('El registro de pyme no existe');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('Ejecucion repetida');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR 7n7');
end;
/
-- -----

create or replace procedure validacion_categoria(
    vc_id_producto number,
    vc_pyme_cedula varchar2,
    vc_nombre_categoria varchar2,
    vc_nombre varchar2,
    vc_descripcion varchar2,
    vc_precio_unitario number)
as
    id_categoria_aux number;

```

```

condicion_auxiliar number;
begin
    select count(*) into condicion_auxiliar from categoria vc
    where nombre_categoria = vc_nombre_categoria;

    if condicion_auxiliar = 0
    then
        consultas_modificacion.insert_categoria(vc_nombre_categoria);
        select vc.id_categoria into id_categoria_aux from categoria vc
        where vc.nombre_categoria = vc_nombre_categoria;
    else
        select vc.id_categoria into id_categoria_aux from categoria vc
        where vc.nombre_categoria = vc_nombre_categoria;
    end if;

    consultas_modificacion.insert_producto(vc_id_producto, id_categoria_aux,
    vc_pyme_cedula, vc_nombre, vc_descripcion, vc_precio_unitario);

exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('Categoria no encontrada');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('Ejecucion repetida');
    when value_error then
        DBMS_OUTPUT.PUT_LINE('Parametros mal ingresados');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR 7n7');

end;
/
-- -----
-- TRIGGER
-- -----

create or replace trigger ingresar_historico after update of precio_unitario on producto
for each row

```



```

declare
begin
    consultas_modificacion.insert_historico (:old.id_producto, SYSDATE,
:old.precio_unitario);
    commit;
exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('No se encontro resultados compatibles');
    when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('Ejecucion repetida');
    when others then
        DBMS_OUTPUT.PUT_LINE('ERROR 7n7');
end;
/

```

-- Encender los prints

set serveroutput ON;

commit;

Descripción	Valor	G11
Captura los objetos necesarios y sus atributos, comportamientos y relaciones	4	3
Entidades, atributos, relaciones. Al menos en 4ta forma normal	4	3
Agradable e intuitiva	2	2
Inserción	2	2
Modificación	2	2
Borrado	2	2
Consulta y reportes	4	2
Nombres significativos	2	1
Definición de PKs	2	2
Definición de FKs y restricciones de no nulidad	2	2
Secuencias	2	2
Funcionamiento	5	5
Funcionamiento y excepciones	10	7
Funcionamiento y excepciones	5	3
Implementación y llamado	5	3
Permite operaciones CRUD	4	3
Cumple con la funcionalidad solicitada	3	1
	60	45
		75