

Instituto Tecnológico de Costa Rica
Escuela de Computación
Bases de Datos II

Examen 1: CorreosTEC

Elaborado Por:
Juan Pablo Martínez Brenes

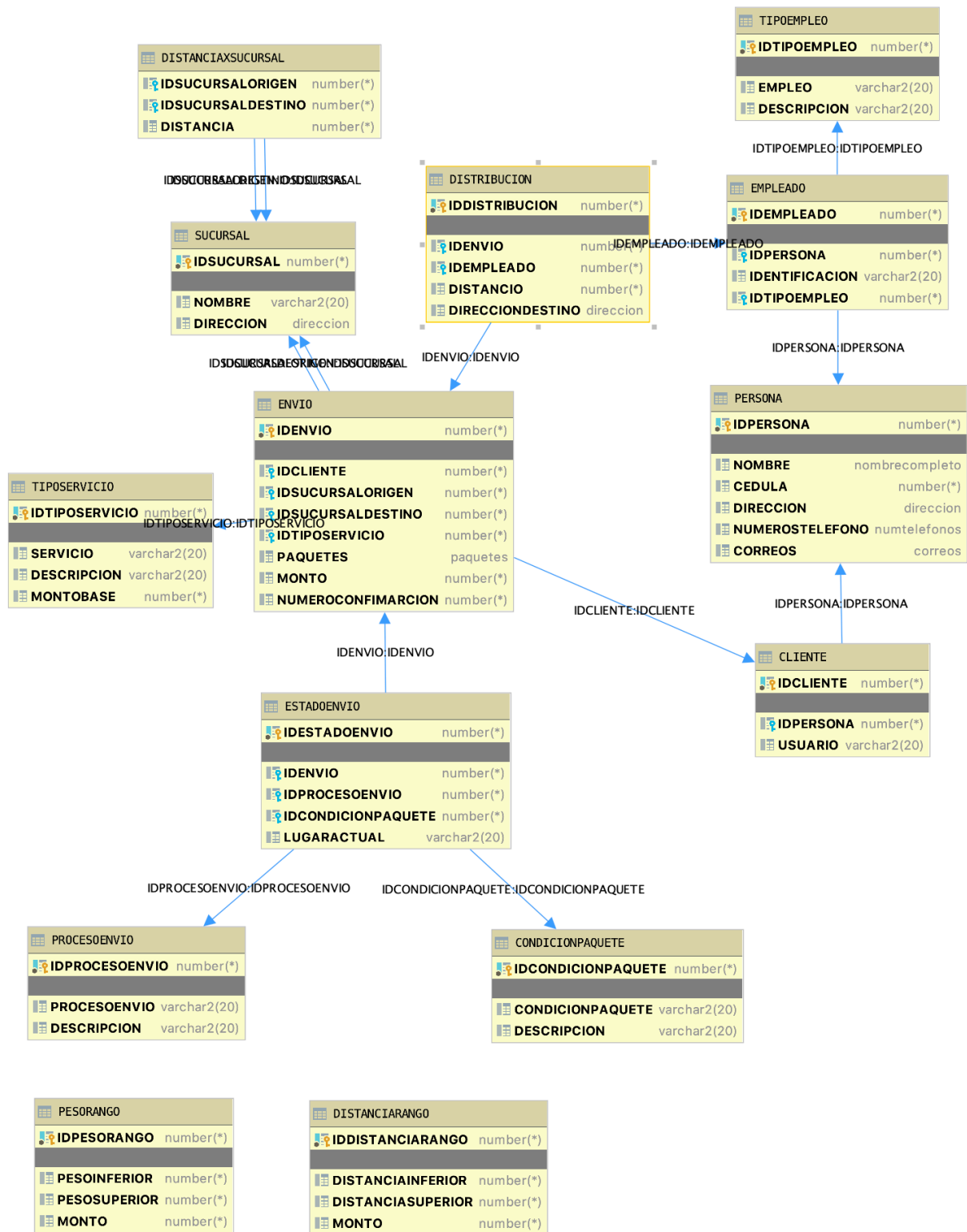
Profesora:
Alicia Salazar Hernandez

I semestre - 2020

Contenido

Diagrama Entidad-Relación	3
Especificaciones de la base de datos	4
Tablas y Datos	4
Procedimientos y Funciones	10
Consulta Usuario	10
Modificar Usuario	11
Montos Facturados por tipo de servicio y sucursal	12
Cantidad de Paquetes enviados por tipo de servicio y sucursal	13
Lista de distribución de paquetes para un mensajero	14
Envíos de n cantidad de paquetes	14
Biografía	15
Anexo	15

Diagrama Entidad-Relación



Especificaciones de la base de datos

La base de datos se desarrolló en Oracle SQL.

Para poner en funcionamiento la base de datos se deben seguir los siguientes pasos:

- Se debe ejecutar el archivo Tablas-Objetos/Objetos.sql para que las tablas puedan acceder a los tipos de datos creados
- Luego se debe ejecutar el archivo Tablas-Objetos/Tablas.sql para crear las tablas
- Posteriormente se llena de datos de la base ejecutando el archivo Data/Data.sql
- Por último, se ejecutan las funciones y se realizan las pruebas.

Tablas y Datos

Tabla sucursal:

```
CREATE TABLE Sucursal
(
  IdSucursal NUMBER PRIMARY KEY NOT NULL,
  Nombre VARCHAR2(30),
  Direccion DIRECCION
);
```

	IDSUCURSAL	NOMBRE	DIRECCION
1	1	Sucursal Costa Rica	{Cartago,Costa Rica,Calle TEC}
2	2	Sucursal Estados Unidos	{New York,Estado Unidos,Calle 11, Ave 4}
3	3	Sucursal China	{Shangai,China,Calle 9, Ave 2}

Tabla DistanciaXSucursal:

```
CREATE TABLE DistanciaXSucursal
(
  IdSucursalOrigen NUMBER,
  IdSucursalDestino NUMBER,
  Distancia NUMBER, -- Distancias en Kilometros
  CONSTRAINT fk_SucursalOrigen FOREIGN KEY (IdSucursalOrigen) REFERENCES Sucursal (IdSucursal),
  CONSTRAINT fk_SucursalDestino FOREIGN KEY (IdSucursalDestino) REFERENCES Sucursal (IdSucursal)
);
```

	IDSUCURSALORIGEN	IDSUCURSALDESTINO	DISTANCIA
1	1	2	1965
2	1	3	15429
3	2	3	15820

Tabla Persona:

```
CREATE TABLE Persona
(
  IdPersona      NUMBER PRIMARY KEY NOT NULL,
  Nombre         NOMBRECOMPLETO,
  Cedula         NUMBER,
  Direccion      DIRECCION,
  NumerosTelefono NUMTELEFONOS,
  Correos        CORREOS
);
```

	IDPERSONA	NOMBRE	CEDULA	DIRECCION	NUMEROSTELEFONO	CORREOS
1	1	{Lammond,Godle...	339885860	{Afghanistan,Qalah...	{{Personal,85028613},{Tra...	{{Personal,lhar...
2	2	{Charmine,Gayn...	649081282	{Philippines,Buala...	{{Personal,84887043},{Tra...	{{Personal,ccog...
3	3	{Emlen,Hawkswo...	241984360	{China,Yanglong,53...	{{Personal,89569081},{Tra...	{{Personal,eric...
4	4	{Wileen,Boswoo...	501581242	{Kazakhstan,Ust-Ka...	{{Personal,85139288},{Tra...	{{Personal,wgri...
5	5	{Lurette,Flind...	217038219	{Sweden,Nykvarn,16...	{{Personal,86377772},{Tra...	{{Personal,ldjo...
6	6	{Brendis,Bangh...	206941093	{Philippines,Corcu...	{{Personal,82578894},{Tra...	{{Personal,brom...
7	7	{Lenka,Eronie,...	887918852	{United States,Des...	{{Personal,87825096},{Tra...	{{Personal,lbat...
8	8	{Karlie,Woller...	601297712	{Vietnam,Thị Trón,...	{{Personal,83842618},{Tra...	{{Personal,kske...
9	9	{Thibaut,Ailsb...	982012283	{Czech Republic,Ve...	{{Personal,83256113},{Tra...	{{Personal,tlan...
10	10	{Pip,Remington...	278023830	{Botswana,Kopong,0...	{{Personal,85910153},{Tra...	{{Personal,ptre...
11	11	{Gayelord,Fair...	223602096	{Serbia,Kladovo,5 ...	{{Personal,89388664},{Tra...	{{Personal,gfis...
12	12	{Jimmy,Nobriga...	467784252	{Lithuania,Kudirko...	{{Personal,85511317},{Tra...	{{Personal,jmig...
13	13	{Dur,Pavek,Dun...	304794379	{Mauritius,Ecroign...	{{Personal,85009300},{Tra...	{{Personal,ddun...
14	14	{Berna,Island,...	474425956	{Greece,Mýrina,098...	{{Personal,87230477},{Tra...	{{Personal,bros...
15	15	{Nady,Brice,Al...	135410912	{Russia,Nyuksenits...	{{Personal,87242495},{Tra...	{{Personal,nall...
16	16	{Ave,Begin,Fra...	317056973	{Czech Republic,St...	{{Personal,84464215},{Tra...	{{Personal,afra...
17	17	{Colly,Iiannon...	217283403	{Sweden,Luleå,5635...	{{Personal,83223879},{Tra...	{{Personal,clef...
18	18	{Aida,Iskov,Do...	482353605	{Peru,Ferreñafe,46...	{{Personal,85611864},{Tra...	{{Personal,adow...
19	19	{Lars,Whooley,...	806236737	{Portugal,Mamede d...	{{Personal,89048753},{Tra...	{{Personal,lmad...
20	20	{Skip,Straffor...	150646318	{Portugal,Charneca...	{{Personal,85392177},{Tra...	{{Personal,ster...
21	21	{Jard,Abram,Ca...	773353427	{Brazil,Rio Pardo,...	{{Personal,86446618},{Tra...	{{Personal,jcat...
22	22	{Paul,Gress,Mu...	375236216	{Philippines,San J...	{{Personal,88614271},{Tra...	{{Personal,pmul...
23	23	{Elwira,Gockel...	803971774	{Indonesia,Pondokw...	{{Personal,89425107},{Tra...	{{Personal,epic...
24	24	{Phillip,Short...	660533465	{China,Bozhou,707 ...	{{Personal,88107729},{Tra...	{{Personal,pdol...
25	25	{Lincoln,Filip...	871146433	{Brazil,Itaperuna,...	{{Personal,83747630},{Tra...	{{Personal,lwha...

Tabla Cliente:

```
CREATE TABLE Cliente
(
  IdCliente      NUMBER PRIMARY KEY NOT NULL,
  IdPersona      NUMBER,
  Usuario        VARCHAR2(20),
  CONSTRAINT fk_PersonaCliente FOREIGN KEY (IdPersona) REFERENCES Persona (IdPersona)
);
```

	IDCLIENTE	IDPERSONA	USUARIO
1	1	1	Usuario 1
2	2	2	Usuario 2
3	3	3	Usuario 3
4	4	4	Usuario 4
5	5	5	Usuario 5
6	6	6	Usuario 6
7	7	7	Usuario 7
8	8	8	Usuario 8
9	9	9	Usuario 9
10	10	10	Usuario 10
11	11	11	Usuario 11
12	12	12	Usuario 12
13	13	13	Usuario 13
14	14	14	Usuario 14
15	15	15	Usuario 15
16	16	16	Usuario 16

Tabla TipoEmpleo:

```
CREATE TABLE TipoEmpleo
(
  IdTipoEmpleo NUMBER PRIMARY KEY NOT NULL,
  Empleo VARCHAR2(20),
  Descripcion VARCHAR2(50)
);
```

	IDTIPOEMPLEO	EMPLEO	DESCRIPCION
1	1	Mensajero	Distribuye paquetes
2	2	Chequeador	Chequea paquetes
3	3	Oficinista	Rastrea paquetes
4	4	Atencion al cliente	Recibe consultas
5	5	Personal de sucursal	Atiende clientes

Tabla Empleado:

```
CREATE TABLE Empleado
(
  IdEmpleado NUMBER PRIMARY KEY NOT NULL,
  IdPersona NUMBER,
  Identificacion VARCHAR2(20),
  IdTipoEmpleo NUMBER,
  CONSTRAINT fk_PersonaEmpleado FOREIGN KEY (IdPersona) REFERENCES Persona (IdPersona),
  CONSTRAINT fk_TipoEmpleoEmpleado FOREIGN KEY (IdTipoEmpleo) REFERENCES TipoEmpleo (IdTipoEmpleo)
);
```

	IDEMPLEADO	IDPERSONA	IDENTIFICACION	IDTIPOEMPLEO
1	1	17	132	1
2	2	18	242	1
3	3	19	353	1
4	4	20	453	1
5	5	21	557	1
6	6	22	667	2
7	7	23	797	3
8	8	24	834	4
9	9	25	957	5

Tabla TipoServicio:

```
CREATE TABLE TipoServicio
(
    IdTipoServicio NUMBER PRIMARY KEY NOT NULL,
    Servicio VARCHAR2(20),
    Descripcion VARCHAR2(50),
    MontoBase Number -- Monto en Dolares
);
```

	IDTIPOSERVICIO	SERVICIO	DESCRIPCION	MONTOBASE
1	1	Expres	Sin retraso	10000
2	2	Avion	Puede tener escalas	8000
3	3	Barco	Tarda más	5000

Tabla PesoRango:

```
CREATE TABLE PesoRango
(
    IdPesoRango NUMBER PRIMARY KEY NOT NULL,
    PesoInferior Number, -- Peso en kilos
    PesoSuperior Number,
    Monto Number -- Monto en Dolares
);
```

	IDPESORANGO	PESOSUPERIOR	PESOSUPERIOR	MONTO
1	1	0	1	0
2	2	1	2	1000
3	3	2	5	2000
4	4	5	10	4000
5	5	10	20	5000
6	6	20	50	10000
7	7	50	100	100000

Tabla DistanciaRango:

```
CREATE TABLE DistanciaRango
(
    IdDistanciaRango NUMBER PRIMARY KEY NOT NULL,
    DistanciaInferior Number,
    DistanciaSuperior Number,
    Monto Number -- Monto en Dolares
);
```

	IDDISTANCIARANGO	DISTANCIASUPERIOR	DISTANCIASUPERIOR	MONTO
1	1	0	100	1000
2	2	100	1000	8000
3	4	1000	8000	15000
4	5	8000	20000	25000

Tabla ProcesoEnvio:

```
CREATE TABLE ProcesoEnvio
(
    IdProcesoEnvio NUMBER PRIMARY KEY NOT NULL,
    ProcesoEnvio VARCHAR2(20),
    Descripcion VARCHAR2(50)
);
```

	IDPROCESOENVIO	PROCESOENVIO	DESCRIPCION
1	1	Pendiente	En espera de ser enviado
2	2	Cerrado	Ya el paquete fue entregado
3	3	En distribucion	En distribucion
4	4	Viajando a sucursal	Viajando a sucursal
5	5	En revision	En revision

Tabla CondicionPaquete:

```
CREATE TABLE CondicionPaquete
(
    IdCondicionPaquete NUMBER PRIMARY KEY NOT NULL,
    CondicionPaquete VARCHAR2(20),
    Descripcion VARCHAR2(50)
);
```

	IDCONDICIONPAQUETE	CONDICIONPAQUETE	DESCRIPCION
1	1	Excelente estado	Excelente estado
2	2	Regular	Regular
3	3	Mal estado	Mal estado

Tabla Envio:

```
CREATE TABLE Envio
(
    IdEnvio NUMBER PRIMARY KEY NOT NULL,
    IdCliente NUMBER,
    IdSucursalOrigen NUMBER,
    IdSucursalDestino NUMBER,
    IdTipoServicio NUMBER,
    Paquetes PAQUETES,
    Monto NUMBER,
    NumeroConfirmacion NUMBER,
    CONSTRAINT fk_ClienteEnvio FOREIGN KEY (IdCliente) REFERENCES Cliente (IdCliente),
    CONSTRAINT fk_SucursalOrigenEnvio FOREIGN KEY (IdSucursalOrigen) REFERENCES Sucursal (IdSucursal),
    CONSTRAINT fk_SucursalDestinoEnvio FOREIGN KEY (IdSucursalDestino) REFERENCES Sucursal (IdSucursal),
    CONSTRAINT fk_TipoServicioEnvio FOREIGN KEY (IdTipoServicio) REFERENCES TipoServicio (IdTipoServicio)
);
```


	IDENVIO	IDCLIENTE	IDSUCURSALORIGEN	IDSUCURSALDESTINO	IDTIPOSERVICIO	PAQUETES	MONTO	NUMEROCONFIMARCION
1	1	1	1	2	1	{{paquete,1}}	10244	1342
2	2	1	3	2	2	{{paquete,2},{paquete,14}}	42000	2345
3	3	2	1	3	3	{{paquete,14}}	53000	3453
4	4	3	1	2	1	{{paquete,15}}	11000	4354
5	5	4	1	3	2	{{paquete,6}}	15000	5533
6	6	4	3	2	3	{{paquete,9},{paquete,153},{paquete,8}}	16400	6422
7	7	5	2	1	1	{{paquete,100}}	16400	7432
8	8	8	3	2	2	{{paquete,121}}	23000	8442
9	9	10	2	3	3	{{paquete,14},{paquete,25}}	16200	9342
10	10	12	1	2	1	{{paquete,15}}	11600	10342
11	11	16	3	2	2	{{paquete,17},{paquete,11}}	10700	11342

Tabla EstadoEnvio:

```

CREATE TABLE EstadoEnvio
(
    IdEstadoEnvio    NUMBER PRIMARY KEY NOT NULL,
    IdEnvio           NUMBER,
    IdProcesoEnvio    NUMBER,
    IdCondicionPaquete NUMBER,
    LugarActual       VARCHAR2(20),
    CONSTRAINT fk_Envio FOREIGN KEY (IdEnvio) REFERENCES Envio (IdEnvio),
    CONSTRAINT fk_ProcesoEnvioEstadoEnvio FOREIGN KEY (IdProcesoEnvio) REFERENCES ProcesoEnvio (IdProcesoEnvio),
    CONSTRAINT fk_CondicionPaqueteEstadoEnvio FOREIGN KEY (IdCondicionPaquete) REFERENCES CondicionPaquete (IdCondicionPaquete)
);

```

	IDESTADOENVIO	IDENVIO	IDPROCESOENVIO	IDCONDICIONPAQUETE	LUGARACTUAL
1	1	1	2	1	Entregado
2	2	2	4	1	mar
3	3	3	1	1	aduanas
4	4	4	5	2	cartago
5	5	5	3	1	san jose
6	6	6	3	1	los angeles
7	7	7	3	1	china
8	8	8	3	2	panama
9	9	9	3	1	san jose
10	10	10	3	1	usa
11	11	11	3	3	alemania

Tabla Distribución:

```

CREATE TABLE Distribucion
(
    IdDistribucion    NUMBER PRIMARY KEY NOT NULL,
    IdEnvio           NUMBER,
    IdEmpleado        NUMBER,
    Distancio         NUMBER,
    DireccionDestino  DIRECCION,
    CONSTRAINT fk_EnvioDistribucion FOREIGN KEY (IdEnvio) REFERENCES Envio (IdEnvio),
    CONSTRAINT fk_EmpleadoDistribucion FOREIGN KEY (IdEmpleado) REFERENCES Empleado (IdEmpleado)
);

```

	IDDISTRIBUCION	IDENVIO	IDEMPLEADO	DISTANCIO	DIRECCIONDESTINO
1	1	5	1	100	{Costa Rica, Cartago,calle 1}
2	2	6	2	312	{Costa Rica, san jose,calle 3}
3	3	7	2	56	{Costa Rica, alajuela,calle 14}
4	4	8	2	746	{China, tokiio,calle 12}
5	5	9	3	764	{China, tokiio,calle 16}
6	6	10	3	876	{Usa, LA,calle 11}
7	7	11	4	945	{Usa, miami,calle 7}

Procedimientos y Funciones

Consulta Usuario

```
-- ConsultarUsuario: consulta cliente por nombre de usuario
CREATE OR REPLACE PROCEDURE ConsultarUsuario(nombreUsuario IN VARCHAR2, nombre_ OUT NOMBRECOMPLETO) AS
BEGIN
    SELECT P.NOMBRE
    INTO nombre_
    FROM Cliente C
    |         INNER JOIN PERSONA P on C.IDPERSONA = P.IDPERSONA
    WHERE C.USUARIO = nombreUsuario;
END;

-- Prueba consulta usuario
DECLARE
    nombre_ NOMBRECOMPLETO;
BEGIN
    ConsultarUsuario( nombreUsuario: 'Usuario 5', nombre_: nombre_);
    DBMS_OUTPUT.PUT( A: nombre_.NOMBRE);
end;
```

Modificar Usuario

```
-- ModificarUsuario: modificar cliente por nombre de usuario
CREATE OR REPLACE PROCEDURE ModificarUsuario(IdCliente_ IN NUMBER, user IN VARCHAR2 DEFAULT NULL, nombre_ IN NOMBRECOMPL
cedula_ IN NUMBER DEFAULT NULL, direct IN DIRECCION DEFAULT NULL, numsTel_
correos_ IN CORREOS DEFAULT NULL) AS

    IdPersona_ NUMBER;

BEGIN
    SELECT P1.IDPERSONA
    INTO IdPersona_
    FROM PERSONA P1
        INNER JOIN CLIENTE C on P1.IDPERSONA = C.IDPERSONA
    WHERE C.IDCLIENTE = IdCliente_;

    if user is not null then
        UPDATE CLIENTE C
        SET USUARIO = user
        WHERE C.IDCLIENTE = IdCliente_;
    end if;

    if nombre_ is not null then
        UPDATE PERSONA P2
        SET P2.NOMBRE = nombre_
        WHERE IDPERSONA = IdPersona_;
    end if;

    if cedula_ is not null then
        UPDATE PERSONA P2
        SET P2.CEDULA = cedula_
        WHERE IDPERSONA = IdPersona_;
    end if;

    if direct is not null then
        UPDATE PERSONA P2
        SET P2.DIRECCION = direct
        WHERE IDPERSONA = IdPersona_;
    end if;

    if numsTel_ is not null then
        UPDATE PERSONA P2
        SET P2.NUMEROSTELEFONO= numsTel_
        WHERE IDPERSONA = IdPersona_;
    end if;

    if correos_ is not null then
        UPDATE PERSONA P2
        SET P2.CORREOS = correos_
        WHERE IDPERSONA = IdPersona_;
    end if;
END ModificarUsuario;
```

```
-- Prueba ModificarUsuario
BEGIN
    ModificarUsuario( IdCliente_: 1, user: 'User 1111'); -- Cambia el nombre de usuario
end;
```

Montos Facturados por tipo de servicio y sucursal

Montos facturados por servicio:

```
-- MontoXServicio: consulta montos facturados por servicio
CREATE OR REPLACE PROCEDURE MontoXServicio IS
BEGIN
    SELECT T.SERVICIO, SUM(Monto) AS Montos
    FROM ENVIO E
        INNER JOIN TIPOSERVICIO T on E.IDTIPOSERVICIO = T.IDTIPOSERVICIO
    GROUP BY T.SERVICIO;
END;
```

	SERVICIO	MONTOS
1	Expres	49244
2	Avion	90700
3	Barco	85600

Montos facturados por sucursal:

```
-- MontoXSucursal: consulta montos facturados por Sucursal
CREATE OR REPLACE PROCEDURE MontoXSucursal IS
BEGIN
    SELECT S.NOMBRE, SUM(Monto) AS Montos
    FROM ENVIO E
        INNER JOIN SUCURSAL S on E.IDSUCURSALDESTINO = S.IDSUCURSAL
    GROUP BY S.NOMBRE;
END;
```

	NOMBRE	MONTOS
1	Sucursal Costa Rica	16400
2	Sucursal China	84200
3	Sucursal Estados Unidos	124944

Cantidad de Paquetes enviados por tipo de servicio y sucursal

Número de paquetes enviados por servicio:

```
-- PaquetesXServicio: consulta paquetes enviados por servicio
CREATE OR REPLACE PROCEDURE PaquetesXServicio IS
BEGIN
    SELECT T.SERVICIO, COUNT(E.IDENVIO) AS CantidaPaquetes
    FROM ENVIO E
        INNER JOIN TIPOSERVICIO T on E.IDTIPOSERVICIO = T.IDTIPOSERVICIO
    GROUP BY T.SERVICIO;
END;
```

	SERVICIO	CANTIDAPAQUETES
1	Expres	4
2	Avion	4
3	Barco	3

Número de paquetes enviados por servicio:

```
-- PaquetesXSucursal: consulta paquetes enviados por Sucursal
CREATE OR REPLACE PROCEDURE PaquetesXSucursal IS
BEGIN
    SELECT S.NOMBRE, COUNT(E.IDENVIO) AS CantidaPaquetes
    FROM ENVIO E
        INNER JOIN SUCURSAL S on E.IDSUCURSALDESTINO = S.IDSUCURSAL
    GROUP BY S.NOMBRE;
END;
```

	NOMBRE	CANTIDAPAQUETES
1	Sucursal Costa Rica	1
2	Sucursal China	3
3	Sucursal Estados Unidos	7

Lista de distribución de paquetes para un mensajero

Lista de distribución para un mensajero según distancia

```
-- PaquetesMensajero: consulta lista de distribucion de un mensajero
CREATE OR REPLACE PROCEDURE PaquetesMensajero IS
BEGIN
    SELECT E.IDENTIFICACION AS IdentificacionMensajero, D.DIRECCIONDESTINO AS DireccionDestino, D.DISTANCIA AS Distancia
    FROM DISTRIBUCION D
        INNER JOIN EMPLEADO E on D.IDEMPLEADO = E.IDEMPLEADO
    WHERE E.IDEMPLEADO = 2
    ORDER BY Distancia DESC ;
END;
```

	IDENTIFICACIONMENSAJERO	DIRECCIONDESTINO	DISTANCIA
1	242	{China, tokiro,calle 12}	746
2	242	{Costa Rica, san jose,calle 3}	312
3	242	{Costa Rica, alajuela,calle 14}	56

Envíos de n cantidad de paquetes

```
-- RealizarEnvio: Realiza envio de un paquete
CREATE OR REPLACE PROCEDURE RealizarEnvio(IdEnvio_ IN Number, IdCliente_ IN NUMBER, IdSucOrigen IN NUMBER,
                                           IdSucDestino IN NUMBER,
                                           IdTipoPaq IN NUMBER, Paquetes_ IN PAQUETES,
                                           NumConf OUT NUMBER) AS
    Monto_    NUMBER := 0;
    MontoPre  Number;
    NumConf   NUMBER := DBMS_RANDOM.RANDOM;
BEGIN
    -- Calcula monto
    SELECT Monto_
    INTO MontoPre
    FROM TIPOSERVICIO
    WHERE IDTIPOSERVICIO = IdTipoPaq;
    Monto_ := + MontoPre;

    for i in 1..Paquetes_.COUNT
    loop
        SELECT P.MONTO
        INTO MontoPre
        FROM PESORANGO P
        WHERE PESOINFERIOR < Paquetes_(i)(1)
            AND Paquetes_(i)(1) > PESOSUPERIOR;
        Monto_ := + MontoPre;
    end loop;

    INSERT INTO ENVIO (IDENVIO, IDCLIENTE, IDSUCSALORIGEN, IDSUCSALDESTINO, IDTIPOSERVICIO, PAQUETES, MONTO,
                      NUMEROCONFIMARCION)
    VALUES (IdEnvio_, IdCliente_, IdSucOrigen, IdSucDestino, IdTipoPaq, Paquetes_, Monto_, NumConf);
END;

-- Prueba
BEGIN
    RealizarEnvio( IdEnvio_: 110, IdCliente_: 1, IdSucOrigen: 1, IdSucDestino: 2, IdTipoPaq: 1,
                  Paquetes_: PAQUETES(PAQUETE('carro', 12)), NumConf: 1234);
END;
```

Biografía

[1]

“CREATE FUNCTION Statement.” [Online]. Available: https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/create_function.htm#LNPLS01370.

[2]

“Database Application Developer’s Guide - Object-Relational Features.” [Online]. Available: https://docs.oracle.com/cd/B19306_01/appdev.102/b14260/adobjint.htm.

[3]

“PL/SQL Object Types.” [Online]. Available: https://docs.oracle.com/cd/A97630_01/appdev.920/a96624/10_objs.htm.

Anexo

Repositorio en GitHub: <https://github.com/juanpa1220/CorreosTEC.git>