



Universidad Tecnológica de Panamá
Facultad de Ingeniería de Sistemas Computacionales
Licenciatura en Ingeniería de Sistemas de Información

Departamento de Sistemas de Información, Control y Evaluación de
Recursos Informáticos

Sistemas de Bases de Datos II

Equipo #1

Proyecto Final: “Implementación de Programación Almacenada para
Base de Datos diseñada previamente”

Facilitador:

Ing. Henry Lezcano

Integrantes:

Aguilar, Milagros	3-740-771
Atencio, Anel	8-950-868
Márquez, Paola	8-949-1108
Rojas, Reynaldo	8-950-792

Grupo:

1IF131

II Semestre, 2020

TABLA DE CONTENIDO

INTRODUCCIÓN	3
Misión de la Base de Datos	4
Objetivos de la misión de la Base de Datos:	4
Definición del Proyecto.....	4
a. Ámbito	4
b. Alcance o límites	5
Análisis de Requerimientos	5
DIAGRAMA ENTIDAD-RELACIÓN	8
MODELO LÓGICO	9
MODELO LÓGICO DEVELOPER	10
MODELO FÍSICO	11
ANEXO	35
CONCLUSIÓN	59

INTRODUCCIÓN

A lo largo de la historia, la humanidad ha afrontado epidemias de diversas magnitudes, algunas devastadoras, como la llamada plaga de Atenas, en el año 430 a.C., en plena Segunda Guerra del Peloponeso. No se conoce la naturaleza de la plaga, pero según el historiador Tucídides, que la contrajo y sobrevivió, la ciudad de Pericles tardó 50 años en recuperarse y, durante esa época, la desesperanza fue tal que sus habitantes perdieron la fe en los dioses y en las leyes ante una inminente muerte. Según Tucídides, la plaga vino de Etiopía y pasó luego a Egipto, Libia y Grecia. Incluso los espartanos que asediaban Atenas se retiraron por el temor a la epidemia.

Los coronavirus son una extensa familia de virus que pueden causar enfermedades tanto en animales como en humanos. En los humanos, se sabe que varios coronavirus causan infecciones respiratorias que pueden ir desde el resfriado común hasta enfermedades más graves como el síndrome respiratorio de Oriente Medio (MERS) y el síndrome respiratorio agudo severo (SRAS). El coronavirus que se ha descubierto más recientemente causa la enfermedad por coronavirus COVID-19. La COVID-19 es la enfermedad infecciosa causada por el coronavirus que se ha descubierto más recientemente. Tanto el nuevo virus como la enfermedad eran desconocidos antes de que estallara el brote en Wuhan (China) en diciembre de 2019.

La pandemia de coronavirus COVID-19 es la crisis de salud global que define nuestro tiempo y el mayor desafío que hemos enfrentado desde la Segunda Guerra Mundial. Desde su aparición en Asia a finales del año pasado, el virus ha llegado a cada continente, excepto a la Antártida. Los casos aumentan a diario en África, las Américas, y Europa.

Las herramientas que están más cercanas a tratar de resolver los problemas causados por este virus son las ecuaciones diferenciales y los modelos estocásticos, pero también la teoría de juegos, big data, machine learning y, en general, el análisis de datos juega ya un papel relevante, especialmente cuando queremos incluir los aspectos de las conductas sociales.

Misión de la Base de Datos

Organizar correctamente la información de los pacientes infectados, con el fin de automatizar los procesos que se realizan diariamente en los hospitales del país.

Objetivos de la misión de la Base de Datos:

Objetivo General

Proporcionar a los usuarios finales una visión abstracta de los datos acerca de los infectados por COVID-19 que existen en el país para así poder manejarlos y organizarlos de una manera más sencilla y con gran velocidad de respuesta.

Objetivos Específicos

- Servir de ayuda para el registro de casos de COVID-19 del país y todo lo necesario para un mayor control y realizar informes cada cierto tiempo.
- Poder tener un mejor control sobre los cuidados y seguimientos que se le realizarán a cada paciente, dependiendo de si se encuentra en el hospital o en aislamiento domiciliario.
- Dar a conocer en qué hospital están atendiendo a cada paciente y en qué sala se encuentra este para una mayor organización y para tener en cuenta en qué hospital pueden seguir admitiendo pacientes y qué cantidad de salas se encuentran disponibles para este fin.

Definición del Proyecto

a. Ámbito

Los centros de salud, hospitales y demás instituciones se encuentran conscientes de que en estos momentos en los que se está viviendo una situación que nunca antes había ocurrido, es necesario realizar la mayor cantidad de pruebas posibles a todos los pacientes que se presentan con síntomas y proporcionar los resultados de manera rápida para así poder colocar a las personas que lo requieran en aislamiento dentro de sus hogares, o en casos más graves, atender a la mayor cantidad de pacientes posibles dentro del hospital, para que en el mejor de los casos, puedan tener una pronta recuperación y haya espacio para ofrecer una atención de calidad a cada uno de los nuevos pacientes.

Para tener un mejor control dentro de esta problemática, es necesario contar con una base de datos que identifique a los pacientes que se registran tanto para hacerse las pruebas como para internarse en el hospital, debido a que así se cuenta con información verídica acerca de cómo van aumentando los casos de infectados día tras día, qué tantas personas se recuperan y qué tantas no logran sobrevivir al Covid-19.

Esto ayudará a automatizar el trabajo que se encuentran realizando muchas personas para proporcionar los datos y la información correcta a toda la población, y ayudará a contar con un mejor orden y organización, además de que será mucho más fácil y rápido compartir estos datos a nivel interno en los centros de salud y hospitales.

b. Alcance o límites

- La base de datos solo es para el registro de infectados de nuestro país (Panamá).
- Sólo se registrarán los datos de pacientes infectados por COVID-19 confirmados.

Análisis de Requerimientos

Para el análisis de los requerimientos necesarios para crear la base de datos utilizamos la técnica de investigación. Esta técnica nos pareció la más adecuada ya que nos parecía que investigar acerca del problema actual en las distintas páginas web o mediante las noticias nacionales, nos hacía ver cómo se iban relacionando los datos y cuáles eran los datos de más importancia que eran necesarios tener a mano en todo momento. Este método nos ayudó también ya que pudimos ser capaces de investigar cómo otros países estaban tratando los datos relacionados con las personas, siendo capaces de poder tener un control de la situación, por lo que mediante la investigación fuimos capaces de ver cuáles eran las medidas de administración de datos de las personas infectadas que ellos utilizaban y que nos serían muy útiles a la hora de desarrollar nuestra base de datos.

Mediante la investigación y el estudio del funcionamiento de la situación actual, se llegó a la conclusión de que los principales datos que son necesarios para la creación de esta base de datos son los siguientes:

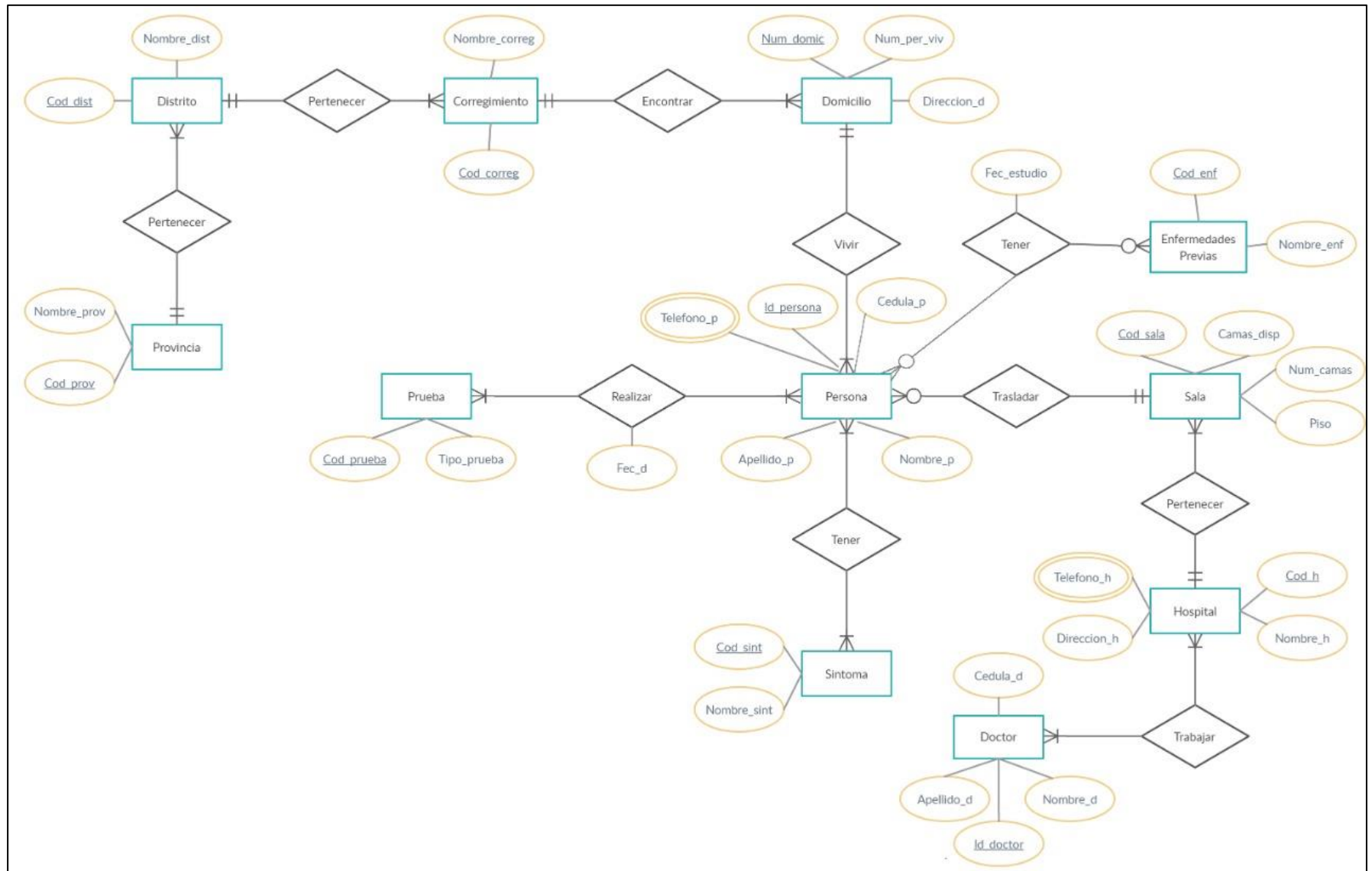
- **Provincia:** Es necesario conocer la provincia en la cual se encuentran las personas que están infectadas con el virus para así poder llevar un control a escala nacional.

- **Distrito:** El conocer la cantidad de personas infectadas en cada distrito permite que las autoridades responsables puedan conocer con exactitud la situación de este para así poder tomar medidas para controlar la expansión del virus.
- **Corregimiento:** sabiendo el corregimiento en el que se encuentran las personas infectadas permite que se tenga un control más preciso por parte de las autoridades para así poder tomar decisiones adecuadas y mantener un control en estas zonas, aplicando medidas de seguridad y sanidad adecuadas para cada una de estas.
- **Domicilio:** Es necesario saber dónde vive cada persona que está infectada para así poder llevar un control total y saber a su vez con cuántas personas se puede relacionar el infectado. Es preciso conocer la dirección del domicilio y cuántas personas viven en este.
- **Persona:** Lo más importante en esta base de datos es saber qué persona está infectada ya que esta es la base para llevar un control preciso de la expansión del virus en el país. Se debe conocer la cédula, nombre, apellido y teléfono de la persona
- **Enfermedades Previas:** Un dato importante para conocer la situación de las personas infectadas, es saber si estas cuentan con enfermedades previas porque con esto se puede realizar tratamientos distintos para cada persona.
- **Sala:** Es necesario saber en qué sala está internado un paciente ya que así se tiene un control preciso del estado del infectado. Es importante conocer el número de camas que la sala posee y el piso del hospital en la que se encuentra.
- **Hospital:** El hospital en el que se encuentra una persona infectada es relevante ya que así se mantiene un orden y a su vez son útiles para conocer la situación del hospital. Conocer el nombre, teléfono y dirección de este es muy importante.
- **Doctor:** Los doctores encargados de realizar las distintas pruebas y llevar el control de los pacientes son importantes para esta base de datos, ya que así se puede conocer qué doctor está realizando las distintas pruebas y análisis de los pacientes. Se debe conocer la cédula, nombre y apellido de cada doctor que se encuentre laborando.
- **Prueba:** Este dato es necesario para conocer con seguridad si una persona está infectada o no del virus. Es importante conocer el resultado de la prueba, y los síntomas que presentaba la persona.

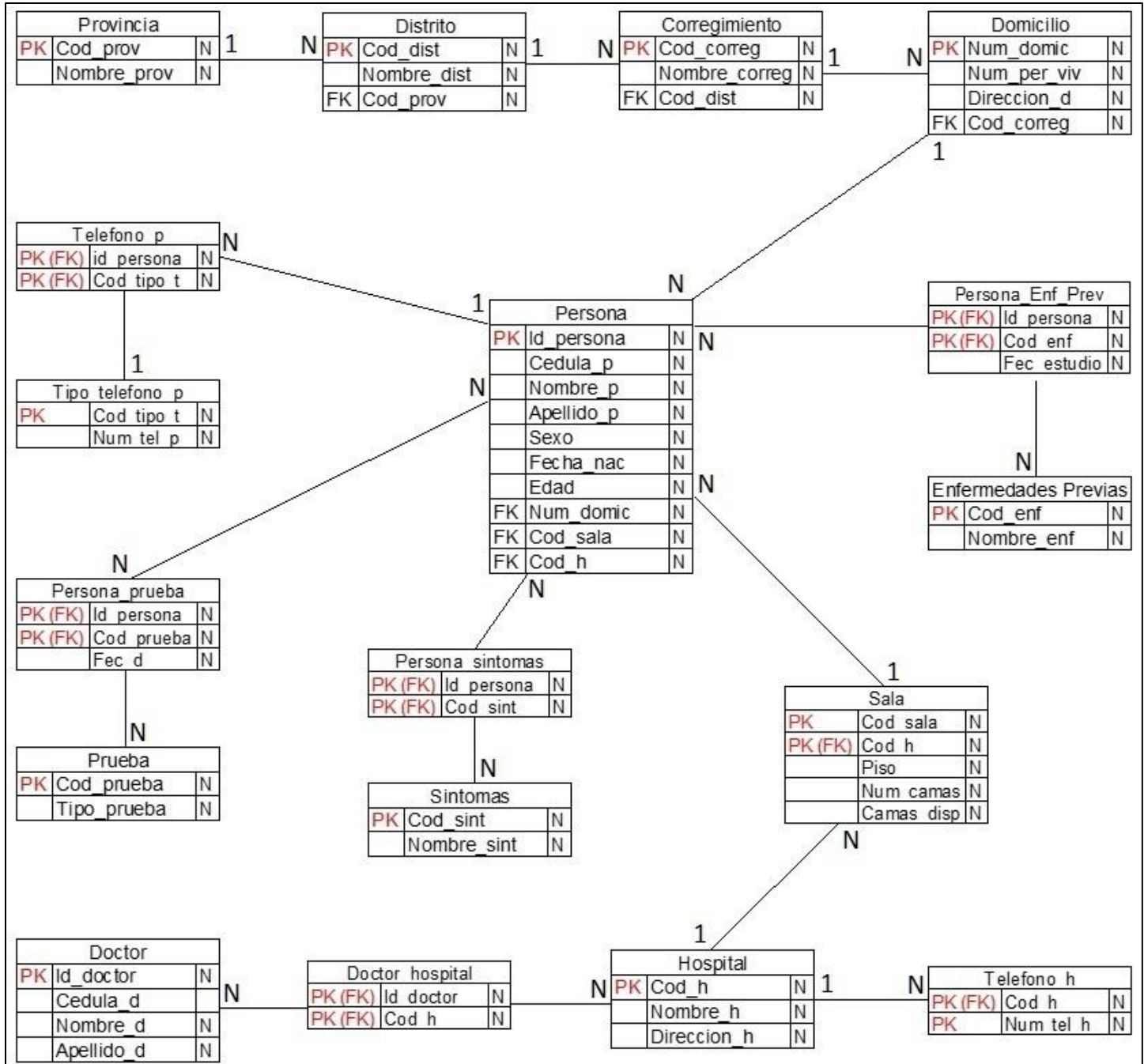
Cómo utilizar o generar los datos

- **Provincia:** Este dato se genera al conocer el distrito en el que vive la persona infectada.
- **Distrito:** Este dato se genera al conocer el corregimiento en el que se encuentra el domicilio de la persona infectada.
- **Corregimiento:** Este dato se genera a partir de la localización del domicilio de la persona, sabiendo en que calle o barrio esta vive.
- **Domicilio:** Este dato se genera al hablar con la persona infectada y tomar su información personal.
- **Persona:** Este dato se genera a medida que más personas se vayan realizando pruebas para conocer si poseen el virus o no.
- **Enfermedades Previas:** Este dato se genera al hablar con la persona infectada y tomar su información personal.
- **Sala:** Este dato se genera al saber en qué situación de salud se encuentra la persona infectada y el hospital al que fue llevado.
- **Hospital:** Este dato se genera al conocer en qué sala se ha trasladado la persona infectada.
- **Doctor:** Este dato se obtiene al hablar con el administrador del hospital y que nos informe de la lista de doctores disponibles en el hospital.
- **Prueba:** Este dato se obtiene una vez que la persona decide ir al hospital y que ahí los doctores le realicen la prueba.

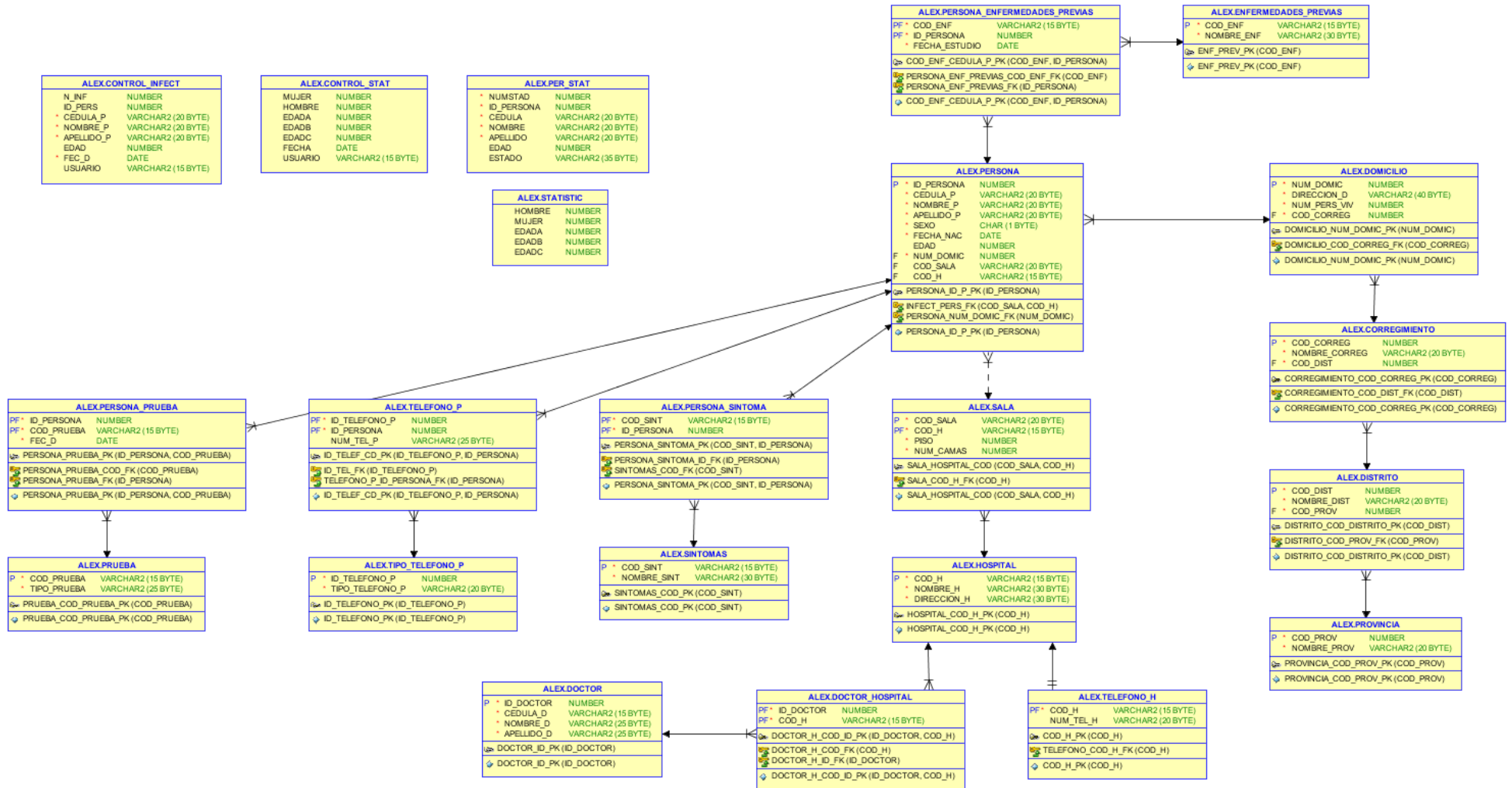
DIAGRAMA ENTIDAD-RELACIÓN



MODELO LÓGICO



MODELO LÓGICO DEVELOPER



MODELO FÍSICO

TABLAS

```
create table Enfermedades_previas (  
  Cod_enf varchar2 (15),  
  constraint Enf_prev_pk primary key (Cod_enf),  
  nombre_enf varchar2 (30) not null  
);
```

```
create table Prueba(  
  Cod_prueba varchar2 (15),  
  constraint Prueba_Cod_prueba_pk primary key (Cod_prueba),  
  tipo_prueba varchar2 (25) not null  
);
```

```
create table Hospital(  
  Cod_h varchar2 (15),  
  constraint Hospital_Cod_h_pk primary key (Cod_h),  
  nombre_h varchar2 (30) not null,  
  direccion_h varchar2 (30) not null  
);
```

```
create table Doctor(  
  id_doctor number,  
  Cedula_d varchar2 (15) not null,  
  constraint Doctor_id_pk primary key (id_doctor),  
  nombre_d varchar2 (25) not null,  
  apellido_d varchar2 (25) not null  
);
```

```
create table Sintomas (  
  Cod_sint varchar2 (15),  
  constraint Sintomas_cod_pk primary key (Cod_sint),  
  nombre_sint varchar2 (30) not null  
);
```

```
create table Provincia (  
  Cod_prov number,  
  constraint Provincia_Cod_prov_pk primary key (Cod_prov),  
  nombre_prov varchar2 (20) not null  
);
```

```
create table Distrito(  
  Cod_dist number,  
  constraint Distrito_Cod_distrito_pk primary key (Cod_dist),  
  nombre_dist varchar2 (20) not null,  
  Cod_prov number not null,  
  constraint Distrito_Cod_prov_fk foreign key (Cod_prov) references Provincia (Cod_prov)  
);
```

```
create table Corregimiento(  
  Cod_correg number,  
  constraint Corregimiento_Cod_correg_pk primary key (Cod_correg),  
  nombre_correg varchar2 (20) not null,
```

```
Cod_dist number not null,  
constraint Corregimiento_Cod_dist_fk foreign key (Cod_dist) references Distrito (Cod_dist)  
);
```

```
create table Domicilio (  
Num_domic number,  
constraint Domicilio_Num_domic_pk primary key (Num_domic),  
direccion_d varchar2 (40) not null,  
num_pers_viv number not null,  
Cod_correg number not null,  
constraint Domicilio_Cod_correg_fk foreign key (Cod_correg) references Corregimiento  
(Cod_correg)  
);
```

```
create table Sala(  
Cod_sala varchar2 (20),  
Cod_h varchar2 (15),  
constraint Sala_Cod_h_fk foreign key (Cod_h) references Hospital (Cod_h),  
constraint Sala_Hospital_cod primary key (Cod_sala, Cod_h),  
piso number not null,  
num_camras number not null  
);
```

```
create table Persona(  
id_persona number,  
Cedula_p varchar2 (20) not null,  
constraint Persona_id_p_pk primary key (id_persona),  
nombre_p varchar2 (20) not null,  
apellido_p varchar2 (20) not null,  
sexo char not null,  
fecha_nac date not null,  
edad number,  
Num_domic number not null,  
constraint Persona_Num_domic_fk foreign key (Num_domic) references Domicilio (Num_domic),  
Cod_sala varchar2 (20),  
Cod_h varchar2 (15),  
constraint infect_pers_fk foreign key (Cod_sala, Cod_h) references Sala (Cod_sala, Cod_h)  
);
```

```
create table Doctor_Hospital (  
id_doctor number,  
constraint Doctor_h_id_fk foreign key (id_doctor) references Doctor (id_doctor),  
Cod_h varchar2 (15),  
constraint Doctor_h_cod_fk foreign key (Cod_h) references Hospital (Cod_h),  
constraint Doctor_h_cod_id_pk primary key (id_doctor, Cod_h)  
);
```

```
create table Persona_Sintoma(  
Cod_sint varchar2 (15),  
constraint Sintomas_cod_fk foreign key (Cod_sint) references Sintomas (Cod_sint),  
id_persona number,  
constraint Persona_Sintoma_id_fk foreign key (id_persona) references Persona (id_persona),  
constraint Persona_sintoma_pk primary key (Cod_sint, id_persona)  
);
```

```
create table Persona_Enfermedades_Previas(  

```

```

Cod_enf varchar2 (15) not null,
constraint Persona_Enf_Previas_Cod_enf_fk foreign key (Cod_enf) references
Enfermedades_previas (Cod_enf),
id_persona number not null,
constraint Persona_Enf_Previas_fk foreign key (id_persona) references Persona (id_persona),
constraint Cod_enf_Cedula_p_pk primary key (Cod_enf, id_persona),
fecha_estudio date not null
);

```

```

create table Persona_Prueba(
id_persona number,
constraint Persona_prueba_fk foreign key (id_persona) references Persona (id_persona),
Cod_prueba varchar2 (15),
constraint Persona_Prueba_cod_fk foreign key (Cod_prueba) references Prueba (Cod_prueba),
constraint Persona_prueba_pk primary key (id_persona, Cod_prueba),
fec_d date not null
);

```

```

create table Telefono_h(
Cod_h varchar2(15),
constraint Telefono_Cod_h_fk foreign key (Cod_h)
references Hospital (Cod_h),
constraint cod_h_pk primary key (Cod_h),
num_tel_h varchar2 (20)
);

```

```

create table Tipo_Telefono_p (
ID_telefono_p number not null,
tipo_telefono_p varchar2 (20) not null,
constraint ID_telefono_pk primary key (ID_telefono_p)
);

```

```

create table Telefono_p(
ID_telefono_p number not null,
id_persona number,
constraint ID_tel_fk foreign key (ID_telefono_p)
references Tipo_Telefono_p (ID_telefono_p),
constraint Telefono_p_id_persona_fk foreign key (id_persona)
references Persona (id_persona),
constraint ID_telef_cd_pk primary key (ID_telefono_p, id_persona),
num_tel_p varchar2(25)
);

```

```

create table per_stat(
numstad number not null,
id_persona number not null,
cedula varchar2(20) not null,
nombre varchar2(20) not null,
apellido varchar2(20) not null,
edad number,
estado varchar2(35)
);

```

```

Create table Control_Infect(
N_inf number,
Id_pers number,

```

```
cedula_p varchar2 (20) not null,  
nombre_p varchar2 (20) not null,  
apellido_p varchar2 (20) not null,  
edad number,  
fec_d date not null,  
usuario varchar2(15)  
);
```

```
create table statistic(  
hombre number,  
mujer number,  
edada number,  
edadb number,  
edadc number  
);
```

```
Create table Control_stat(  
mujer number,  
hombre number,  
edada number,  
edadb number,  
edadc number,  
fecha date,  
usuario varchar2(15)  
);
```

SECUENCIAS

```
CREATE SEQUENCE seq_doc  
START WITH 1  
INCREMENT BY 1;
```

```
CREATE SEQUENCE seq_pers  
START WITH 1  
INCREMENT BY 1;
```

```
create sequence trig_per  
start with 1  
increment by 1;
```

```
create sequence trig_stad  
start with 1  
increment by 1;
```

```
create sequence stat  
start with 1  
increment by 1;
```

FUNCIONES

```
CREATE OR REPLACE FUNCTION calc_edad (fecha_nac PERSONA.fecha_nac%TYPE)
RETURN NUMBER AS
BEGIN
RETURN FLOOR (months_between (sysdate, fecha_nac)/12);
END calc_edad;
/
```

```
CREATE OR REPLACE FUNCTION stat_male
RETURN NUMBER AS
v_hom NUMBER;
v_total NUMBER;
v_resp NUMBER;
BEGIN
SELECT count(sexo) INTO v_hom FROM Persona
WHERE sexo='M';
SELECT count(sexo) INTO v_total FROM Persona;
v_resp:=(v_hom/v_total)*100;
RETURN v_resp;
END stat_male;
/
```

```
CREATE OR REPLACE FUNCTION stat_female
RETURN NUMBER AS
v_muj NUMBER;
v_total NUMBER;
v_resp NUMBER;
BEGIN
SELECT count(sexo) INTO v_muj FROM Persona
WHERE sexo='F';
SELECT count(sexo) INTO v_total FROM Persona;
v_resp:=(v_muj/v_total)*100;
RETURN v_resp;
END stat_female;
/
```

```
CREATE OR REPLACE FUNCTION population(
p_edad persona.edad%TYPE)
RETURN VARCHAR2 AS
BEGIN
IF p_edad >= 60 THEN
RETURN 'Riesgo Alto';
ELSIF p_edad >= 30 THEN
RETURN 'Riesgo Regular';
ELSE
RETURN 'Riesgo Bajo';
END IF;
END population;
/
```



```

CREATE OR REPLACE FUNCTION stat_edadb
RETURN NUMBER AS
v_edadc NUMBER;
v_total NUMBER;
v_resp NUMBER;
BEGIN
SELECT count(edad) INTO v_edadc FROM Persona
WHERE edad>=30 AND edad<60;
SELECT count(*) INTO v_total FROM Persona;
v_resp:=(v_edadc/v_total)*100;
RETURN v_resp;
END stat_edadb;
/

```

```

CREATE OR REPLACE FUNCTION stat_edadc
RETURN NUMBER AS
v_edadc NUMBER;
v_total NUMBER;
v_resp NUMBER;
BEGIN
SELECT count(edad) INTO v_edadc FROM Persona
WHERE edad<30;
SELECT count(id_persona) INTO v_total FROM Persona;
v_resp:=(v_edadc/v_total)*100;
RETURN v_resp;
END stat_edadc;
/

```

```

CREATE OR REPLACE FUNCTION stat_edada
RETURN NUMBER AS
v_edadc NUMBER;
v_total NUMBER;
v_resp NUMBER;
BEGIN
SELECT count(edad) INTO v_edadc FROM Persona
WHERE edad>=60;
SELECT count(id_persona) INTO v_total FROM Persona;
v_resp:=(v_edadc/v_total)*100;
RETURN v_resp;
END stat_edada;
/

```

TRIGGERS

```
CREATE OR REPLACE TRIGGER per_pac
BEFORE UPDATE OR DELETE OR INSERT ON Persona
FOR EACH ROW
DECLARE
    v_Control CHAR(1);
BEGIN
    IF UPDATING THEN
        v_Control := 'U';
        INSERT INTO Control_Infect (n_inf, id_pers, cedula_p, nombre_p, apellido_p, edad, fec_d, usuario)
        VALUES (trig_per.nextval, :NEW.id_persona, :NEW.cedula_p, :NEW.nombre_p, :new.apellido_p,
        :new.edad, sysdate, user);
    ELSIF INSERTING THEN
        v_Control := 'I';
        INSERT INTO Control_Infect (n_inf, id_pers, cedula_p, nombre_p, apellido_p, edad, fec_d, usuario)
        VALUES (trig_per.nextval, :NEW.id_persona, :NEW.cedula_p, :NEW.nombre_p, :new.apellido_p,
        :new.edad, sysdate, user);
    ELSIF DELETING THEN
        v_Control := 'D';
        INSERT INTO Control_Infect (n_inf, id_pers, cedula_p, nombre_p, apellido_p, edad, fec_d, usuario)
        VALUES (trig_per.nextval, :NEW.id_persona, :NEW.cedula_p, :NEW.nombre_p, :new.apellido_p,
        :new.edad, sysdate, user);
    END IF;
END per_pac;
/
```

```
CREATE OR REPLACE TRIGGER upd_static
BEFORE INSERT OR UPDATE ON Statistic
FOR EACH ROW
DECLARE
    v_Control CHAR(1);
BEGIN
    IF UPDATING THEN
        v_Control := 'U';
        INSERT INTO control_stat
        values (:NEW.mujer, :NEW.hombre, :NEW.edada, :NEW.edadb, :NEW.edadc, sysdate, user);
    ELSIF INSERTING THEN
        v_Control := 'I';
        INSERT INTO control_stat
        values (:NEW.mujer, :NEW.hombre, :NEW.edada, :NEW.edadb, :NEW.edadc, sysdate, user);
    END IF;
END upd_static;
/
```

PROCEDIMIENTOS

```
CREATE OR REPLACE PROCEDURE add_enfermedad (  
  p_code enfermedades_previas.cod_enf%TYPE,  
  p_nombrenf enfermedades_previas.nombre_enf%TYPE  
) AS  
BEGIN  
  INSERT INTO Enfermedades_previas (cod_enf, nombre_enf)  
  values (p_code, p_nombrenf);  
EXCEPTION  
  WHEN NO_DATA_FOUND THEN  
    DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
  WHEN DUP_VAL_ON_INDEX THEN  
    DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
  WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
COMMIT;  
END add_enfermedad;  
/
```

```
CREATE OR REPLACE PROCEDURE add_prueba (  
  p_codpb prueba.cod_prueba%TYPE,  
  p_tipop prueba.tipo_prueba%TYPE  
) AS  
BEGIN  
  INSERT INTO Prueba (cod_prueba, tipo_prueba)  
  values (p_codpb, p_tipop);  
EXCEPTION  
  WHEN NO_DATA_FOUND THEN  
    DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
  WHEN DUP_VAL_ON_INDEX THEN  
    DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
  WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
COMMIT;  
END add_prueba;  
/
```

```
CREATE OR REPLACE PROCEDURE add_hospital (  
  p_codh hospital.cod_h%TYPE,  
  p_nombreh hospital.nombre_h%TYPE,  
  p_direccionh hospital.direccion_h%TYPE  
) AS  
BEGIN  
  INSERT INTO Hospital (cod_h, nombre_h, direccion_h)  
  values (p_codh, p_nombreh, p_direccionh);  
EXCEPTION  
  WHEN NO_DATA_FOUND THEN  
    DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
  WHEN DUP_VAL_ON_INDEX THEN  
    DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
  WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
COMMIT;  
END add_hospital;  
/
```

```

CREATE OR REPLACE PROCEDURE add_doctor (
p_cedulad doctor.cedula_d%TYPE,
p_nombred doctor.nombre_d%TYPE,
p_apellidod doctor.apellido_d%TYPE
) AS
BEGIN
INSERT INTO Doctor (id_doctor, cedula_d, nombre_d, apellido_d)
values (seq_doc.nextval, p_cedulad, p_nombred, p_apellidod);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
COMMIT;
END add_doctor;
/

```

```

CREATE OR REPLACE PROCEDURE add_sintoma (
p_cods sintomas.cod_sint%TYPE,
p_nombres sintomas.nombre_sint%TYPE
) AS
BEGIN
INSERT INTO Sintomas (cod_sint, nombre_sint)
values (p_cods, p_nombres);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
COMMIT;
END add_sintoma;
/

```

```

CREATE OR REPLACE PROCEDURE add_provincia (
p_codp provincia.cod_prov%TYPE,
p_nombrep provincia.nombre_prov%TYPE
) AS
BEGIN
INSERT INTO Provincia (cod_prov, nombre_prov)
values (p_codp, p_nombrep);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
COMMIT;
END add_provincia;
/

```

```

CREATE OR REPLACE PROCEDURE add_distrito (
p_coddiss distrito.cod_dist%TYPE,
p_nombredis distrito.nombre_dist%TYPE,
p_codprov distrito.cod_prov%TYPE
) AS
BEGIN
INSERT INTO Distrito (cod_dist, nombre_dist, cod_prov)
values (p_coddiss, p_nombredis, p_codprov);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
COMMIT;
END add_distrito;
/

```

```

CREATE OR REPLACE PROCEDURE add_correg (
p_codcorr corregimiento.cod_correg%TYPE,
p_nombreco corregimiento.nombre_correg%TYPE,
p_codis corregimiento.cod_dist%TYPE
) AS
BEGIN
INSERT INTO Corregimiento (cod_correg, nombre_correg, cod_dist)
values (p_codcorr, p_nombreco, p_codis);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
COMMIT;
END add_correg;
/

```

```

CREATE OR REPLACE PROCEDURE add_domicilio (
p_numdom domicilio.num_domic%TYPE,
p_direcciond domicilio.direccion_d%TYPE,
p_numpers domicilio.num_pers_viv%TYPE,
p_codcor domicilio.cod_correg%TYPE
) AS
BEGIN
INSERT INTO Domicilio (num_domic, direccion_d, num_pers_viv, cod_correg)
values (p_numdom, p_direcciond, p_numpers, p_codcor);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
COMMIT;
END add_domicilio;

```

/

```
CREATE OR REPLACE PROCEDURE add_persona (  
  p_cedulap persona.cedula_p%TYPE,  
  p_nombrep persona.nombre_p%TYPE,  
  p_apellidop persona.apellido_p%TYPE,  
  p_sexo persona.sexo%TYPE,  
  p_fechan persona.fecha_nac%TYPE,  
  p_numdom persona.num_domic%TYPE,  
  p_codsala persona.cod_sala%TYPE,  
  p_codh persona.cod_h%TYPE  
) AS  
  p_edad persona.edad%TYPE;  
BEGIN  
  p_edad:=calc_edad(p_fechan);  
  INSERT INTO Persona (id_persona, cedula_p, nombre_p, apellido_p, sexo, fecha_nac, edad,  
  num_domic, cod_sala, cod_h)  
  values (seq_pers.nextval, p_cedulap, p_nombrep, p_apellidop, p_sexo, p_fechan, p_edad,  
  p_numdom, p_codsala, p_codh);  
EXCEPTION  
  WHEN NO_DATA_FOUND THEN  
    DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
  WHEN DUP_VAL_ON_INDEX THEN  
    DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
  WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
COMMIT;  
END add_persona;  
/
```

```
CREATE OR REPLACE PROCEDURE Add_doctor_hospital (  
  p_id_doctor doctor_hospital.id_doctor%TYPE,  
  p_cod_h doctor_hospital.cod_h%TYPE  
) AS  
BEGIN  
  INSERT INTO doctor_hospital (id_doctor, cod_h)  
  values (p_id_doctor, p_cod_h);  
EXCEPTION  
  WHEN NO_DATA_FOUND THEN  
    DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
  WHEN DUP_VAL_ON_INDEX THEN  
    DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
  WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
COMMIT;  
END Add_doctor_hospital;  
/
```

```
CREATE OR REPLACE PROCEDURE add_telefono_h (  
  p_cod_h telefono_h.cod_h%TYPE,  
  p_num_tel_h telefono_h.num_tel_h%TYPE  
) AS  
BEGIN  
  INSERT INTO telefono_h (cod_h, num_tel_h)  
  values (p_cod_h, p_num_tel_h);  
EXCEPTION
```

```

        WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
        WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
        WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
    COMMIT;
END add_telefono_h;
/

```

```

CREATE OR REPLACE PROCEDURE add_sala (
p_cod_sala sala.cod_sala%TYPE,
p_cod_h sala.cod_h%TYPE,
p_piso sala.piso%TYPE,
p_num_camas sala.num_camas%TYPE
) AS
BEGIN
INSERT INTO sala (cod_sala, cod_h, piso, num_camas)
values (p_cod_sala, p_cod_h, p_piso, p_num_camas);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
    WHEN DUP_VAL_ON_INDEX THEN
    DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
    WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
    COMMIT;
END add_sala;
/

```

```

CREATE OR REPLACE PROCEDURE add_tipo_telefono_p (
    p_id_telefono_p IN tipo_telefono_p.id_telefono_p%TYPE,
    p_tipo_telefono_p IN tipo_telefono_p.tipo_telefono_p%TYPE
) AS
BEGIN
    INSERT INTO Tipo_Telefono_p (id_telefono_p, tipo_telefono_p)
    VALUES (p_id_telefono_p, p_tipo_telefono_p);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
    COMMIT;
END add_tipo_telefono_p;
/

```

```

CREATE OR REPLACE PROCEDURE add_telefono_p (
    p_id_telefono_p IN telefono_p.id_telefono_p%TYPE,
    p_id_persona IN telefono_p.id_persona%TYPE,
    p_num_tel_p IN telefono_p.num_tel_p%TYPE
) AS
BEGIN
    INSERT INTO telefono_p(id_telefono_p, id_persona, num_tel_p)
    VALUES (p_id_telefono_p, p_id_persona, p_num_tel_p);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE ('No se encontró la tabla');
    COMMIT;

```

```
END add_telefono_p;  
/
```

```
CREATE OR REPLACE PROCEDURE add_persona_sintoma (  
p_cod_sint persona_sintoma.cod_sint%TYPE,  
p_id_persona persona_sintoma.id_persona%TYPE  
) AS  
BEGIN  
INSERT INTO Persona_sintoma (cod_sint, id_persona)  
Values (p_cod_sint, p_id_persona);  
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
    WHEN DUP_VAL_ON_INDEX THEN  
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
COMMIT;  
END add_persona_sintoma;  
/
```

```
CREATE OR REPLACE PROCEDURE add_persona_prueba (  
p_id_persona persona_prueba.id_persona%TYPE,  
p_cod_prueba persona_prueba.cod_prueba%TYPE,  
p_fec_d persona_prueba.fec_d%TYPE  
) AS  
BEGIN  
INSERT INTO Persona_prueba (id_persona, cod_prueba, fec_d)  
values (p_id_persona, p_cod_prueba, p_fec_d);  
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
    WHEN DUP_VAL_ON_INDEX THEN  
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
COMMIT;  
END add_persona_prueba;  
/
```

```
CREATE OR REPLACE PROCEDURE add_persona_ep (  
p_cod_enf persona_enfermedades_previas.cod_enf%TYPE,  
p_id_persona persona_enfermedades_previas.id_persona%TYPE,  
p_fecha_estudio persona_enfermedades_previas.fecha_estudio%TYPE  
) AS  
BEGIN  
INSERT INTO Persona_enfermedades_previas (cod_enf, id_persona, fecha_estudio)  
values (p_cod_enf, p_id_persona, p_fecha_estudio);  
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
    WHEN DUP_VAL_ON_INDEX THEN  
        DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
```



```

COMMIT;
END add_persona_ep;
/

```

```

CREATE OR REPLACE PROCEDURE act_age
AS
CURSOR c_edad IS
SELECT id_persona, cedula_p, nombre_p, apellido_p, edad
FROM Persona;
v_idp persona.id_persona%TYPE;
v_nombre persona.nombre_p%TYPE;
v_apellido persona.apellido_p%TYPE;
v_edad persona.edad%TYPE;
v_cedula persona.cedula_p%TYPE;
v_funcion per_stat.estado%TYPE;
BEGIN
OPEN c_edad;
LOOP
FETCH c_edad INTO v_idp, v_cedula, v_nombre, v_apellido, v_edad;
v_funcion:=population(v_edad);
INSERT INTO per_stat (numstad, id_persona, cedula, nombre, apellido, edad, estado)
values (stat.nextval,v_idp, v_cedula, v_nombre, v_apellido, v_edad, v_funcion);
EXIT WHEN c_edad%NOTFOUND;
END LOOP;
CLOSE c_edad;
EXCEPTION
WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE ('No se encontró los datos');
COMMIT;
END act_age;
/

```

```

create or replace PROCEDURE upd_stat
AS
p_hom statistic.hombre%TYPE;
p_muj statistic.mujer%TYPE;
p_edada statistic.edada%TYPE;
p_edadb statistic.edadb%TYPE;
p_edadc statistic.edadc%TYPE;
BEGIN
p_hom:=stat_male;
p_muj:=stat_female;
p_edada:=stat_edada;
p_edadb:=stat_edadb;
p_edadc:=stat_edadc;
INSERT INTO Statistic
values (p_hom, p_muj,p_edada,p_edadb,p_edadc);
EXCEPTION
WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE ('No se encontro los datos');
COMMIT;
END upd_stat;
/

```

INVOCACIONES

BEGIN

```
add_enfermedad ('HIP', 'Hipertensión');
add_enfermedad ('CAN', 'Cáncer');
add_enfermedad ('DIA', 'Diabetes');
add_enfermedad ('EPU', 'Enfermedades Pulmonares');
add_enfermedad ('ECA', 'Enfermedades Cardiacas');
END;
/
```

BEGIN

```
add_prueba ('S', 'Prueba Serológica');
add_prueba ('H', 'Prueba por Hisopado');
END;
/
```

BEGIN

```
add_hospital ('8-H1', 'Hospital Santo Tomás', 'Calle 37 Este');
add_hospital ('8-H2', 'Clínica Hospital San Fernando', 'Vía España');
add_hospital ('8-H3', 'Hospital Santa Fe', 'Ave. Simón Bolívar');
add_hospital ('8-H4', 'Hospital Modular', 'Albrook');
add_hospital ('8-H5', 'Complejo Dr. Arnulfo A. Madrid', 'Ave. Simón Bolívar');
add_hospital ('3-H1', 'Hospital Manuel A. Guerrero', 'Calle 11');
add_hospital ('3-H2', 'Hospital Colón 4 Altos', '4 Altos');
add_hospital ('4-H1', 'Hospital Chiriquí', 'Ave. 3ra y Calle Central');
add_hospital ('4-H2', 'Hospital Dr. Rafael Hernández', 'San Pablo Viejo');
add_hospital ('13-H1', 'Hospital Nicolás A. Solano', 'La Chorrera');
add_hospital ('13-H2', 'Clínica Hospital Panamericano', 'Calle Santa Rita');
add_hospital ('6-H1', 'Hospital Gustavo N. Collado', 'Ave. Rinaldo Camarano');
add_hospital ('7-H1', 'Hospital Anita Moreno', 'Ave. Dr. Belisario Porras');
add_hospital ('1-H1', 'Hospital Guillermo S. Borbón', 'Isla Colon');
add_hospital ('1-H2', 'Isla Colon Regional Hospital', 'Ave. H Norte');
add_hospital ('9-H1', 'Hospital Dr. Luis Fábrega', 'Santiago');
add_hospital ('9-H2', 'Policlínica Dr. Horacio Gómez', 'Santiago');
add_hospital ('5-H1', 'Hospital de Metetí', 'Metetí');
add_hospital ('2-H1', 'Hospital Aquilino Tejeira', 'Penonomé');
add_hospital ('2-H2', 'Hospital Dr. Rafael Estévez', 'Aguadulce');
add_hospital ('12-H1', 'Hospital de San Pedrito', 'Ngöbe-Buglé');
add_hospital ('10-H1', 'Hospital Marvel Iglesia', 'Guna Yala');
END;
/
```

BEGIN

```
add_doctor ('8-365-873', 'Tatiana', 'Ojo');
add_doctor ('7-546-654', 'Vivian', 'Aguilar');
add_doctor ('8-234-432', 'Denisse', 'Thorne');
add_doctor ('8-465-543', 'Reynaldo', 'Castillero');
add_doctor ('3-65-673', 'Giselle', 'Balmaceda');
add_doctor ('4-100-657', 'Moisés', 'De Sedas');
add_doctor ('1-432-532', 'Jeannette', 'Johnson');
add_doctor ('2-348-43', 'Tommy', 'Liao');
add_doctor ('9-543-67', 'Andrés', 'Soto');
add_doctor ('6-234-6754', 'José', 'Lorenzo');
add_doctor ('5-123-890', 'Gustavo', 'Santamaría');
add_doctor ('3-234-567', 'Sunshine', 'Del Cid');
```

```
add_doctor ('9-321-789', 'Raúl', 'Mendoza');
add_doctor ('8-332-654', 'Italys', 'Ledezma');
add_doctor ('7-654-2134', 'Daniel', 'Araúz');
END;
/
```

```
BEGIN
add_sintoma ('F', 'Fiebre');
add_sintoma ('TS', 'Tos seca');
add_sintoma ('C', 'Cansancio');
add_sintoma ('DG', 'Dolor de garganta');
add_sintoma ('DC', 'Dolor de cabeza');
add_sintoma ('D', 'Diarrea');
add_sintoma ('DM', 'Molestias y dolores');
add_sintoma ('CO', 'Conjuntivitis');
add_sintoma ('DR', 'Dificultad para respirar');
add_sintoma ('E', 'Erupciones cutáneas');
add_sintoma ('IH', 'Incápacidad para hablar');
add_sintoma ('P', 'Pérdida de olfato y gusto');
END;
/
```

```
BEGIN
add_provincia (1, 'Bocas del Toro');
add_provincia (2, 'Coclé');
add_provincia (3, 'Colón');
add_provincia (4, 'Chiriquí');
add_provincia (5, 'Darién');
add_provincia (6, 'Herrera');
add_provincia (7, 'Los Santos');
add_provincia (8, 'Panamá');
add_provincia (9, 'Veraguas');
add_provincia (10, 'Guna Yala');
add_provincia (11, 'Emberá Wounaan');
add_provincia (12, 'Ngäbe-buglé');
add_provincia (13, 'Panamá Oeste');
add_provincia (14, 'Madugandí');
add_provincia (15, 'Wargandí');
END;
/
```

```
BEGIN
add_distrito (0101, 'Almirante', 1);
add_distrito (0102, 'Bocas del Toro', 1);
add_distrito (0103, 'Changuinola', 1);
add_distrito (0201, 'Penonomé', 2);
add_distrito (0202, 'Aguadulce', 2);
add_distrito (0203, 'Antón', 2);
add_distrito (0301, 'Colón', 3);
add_distrito (0302, 'Chagres', 3);
add_distrito (0303, 'Portobelo', 3);
add_distrito (0401, 'David', 4);
add_distrito (0402, 'San Félix', 4);
add_distrito (0403, 'Barú', 4);
add_distrito (0501, 'Chepigana', 5);
add_distrito (0502, 'Pinogana', 5);
```

```
add_distrito (0503, 'Santa Fe', 5);
add_distrito (0601, 'Chitré', 6);
add_distrito (0602, 'Ocú', 6);
add_distrito (0603, 'Las Minas', 6);
add_distrito (0701, 'Las Tablas', 7);
add_distrito (0702, 'Tonosí', 7);
add_distrito (0703, 'Guararé', 7);
add_distrito (0801, 'San Miguelito', 8);
add_distrito (0802, 'Panamá', 8);
add_distrito (0803, 'Balboa', 8);
add_distrito (0901, 'Santiago', 9);
add_distrito (0902, 'Atalaya', 9);
add_distrito (0903, 'Calobre', 9);
add_distrito (1101, 'Cémaco', 11);
add_distrito (1102, 'Sambú', 11);
add_distrito (1201, 'Besikó', 12);
add_distrito (1202, 'Jirondai', 12);
add_distrito (1203, 'Kankintú', 12);
add_distrito (1301, 'Arraiján', 13);
add_distrito (1302, 'La Chorrera', 13);
add_distrito (1303, 'San Carlos', 13);
END;
/
```

BEGIN

```
add_correg (010101, 'Barrio Francés', 0101);
add_correg (010201, 'Bastimentos', 0102);
add_correg (010301, 'El Silencio', 0103);
add_correg (020101, 'Río Grande', 0201);
add_correg (020201, 'Pocrí', 0202);
add_correg (020301, 'El Valle', 0203);
add_correg (030101, 'Cristóbal', 0301);
add_correg (030201, 'Piña', 0302);
add_correg (030301, 'Cacique', 0303);
add_correg (040101, 'San Carlos', 0401);
add_correg (040201, 'Las Lajas', 0402);
add_correg (040301, 'Puerto Armuelles', 0403);
add_correg (050101, 'La Palma', 0501);
add_correg (050201, 'Yaviza', 0502);
add_correg (050301, 'Agua Fría', 0503);
add_correg (060101, 'Monagrillo', 0601);
add_correg (060201, 'Llano Grande', 0602);
add_correg (060301, 'Chumical', 0603);
add_correg (070101, 'Valle Rico', 0701);
add_correg (070201, 'Cambutal', 0702);
add_correg (070301, 'Llano Abajo', 0703);
add_correg (080101, 'Omar Torrijos', 0801);
add_correg (080201, 'Don Bosco', 0802);
add_correg (080301, 'San Miguel', 0803);
add_correg (090101, 'Urracá', 0901);
add_correg (090201, 'San Antonio', 0902);
add_correg (090301, 'La Yeguada', 0903);
add_correg (110101, 'Lajas Blancas', 1101);
add_correg (110201, 'Río Sabalo', 1102);
add_correg (120101, 'Cerro Banco', 1201);
add_correg (120201, 'Man Creek', 1202);
```

```
add_correg (120301, 'Piedra Roja', 1203);
add_correg (130101, 'Vista Alegre', 1301);
add_correg (130201, 'El Coco', 1302);
add_correg (130301, 'Las Uvas', 1303);
END;
/
```

BEGIN

```
add_domicilio (1803, 'Ave. Olmedo Solé', 7, 10101);
add_domicilio (1567, 'Calle Sea Wall', 4, 10201);
add_domicilio (1726, 'Sendero Caimán', 9, 10301);
add_domicilio (6489, 'Calle Verde', 3, 20101);
add_domicilio (8456, 'Calle Abajo', 3, 20201);
add_domicilio (8623, 'Ave. Castor', 6, 20301);
add_domicilio (560, 'Calle Fitzpatrick', 5, 30101);
add_domicilio (23, 'Calle Sandía', 3, 30201);
add_domicilio (789, 'Calle Indio', 5, 30301);
add_domicilio (85, 'Calle América', 4, 40101);
add_domicilio (236, 'Rincón Largo', 7, 40201);
add_domicilio (889, 'Ave. Barco', 2, 40301);
add_domicilio (1582, 'Calle Coco', 4, 50101);
add_domicilio (432, 'Ave. Candelario', 8, 50201);
add_domicilio (976, 'Sector Hielo', 3, 50301);
add_domicilio (25, 'Calle Mercado', 6, 60101);
add_domicilio (4, 'Calle Arriba', 1, 60201);
add_domicilio (57, 'Ave. Montañosa', 4, 60301);
add_domicilio (478, 'Calle Mala', 9, 70101);
add_domicilio (245, 'Calle Dulce', 3, 70201);
add_domicilio (267, 'Calle San Francisco', 2, 70301);
add_domicilio (332, 'San Isidro Sector B', 5, 80101);
add_domicilio (442, 'Calle F', 3, 80201);
add_domicilio (90, 'Ave. Roja', 5, 80301);
add_domicilio (1478, 'Calle Centavo', 3, 90101);
add_domicilio (898, 'Calle Santa', 2, 90201);
add_domicilio (76, 'Calle Y Norte', 6, 90301);
add_domicilio (2, 'Calle Negra', 5, 110101);
add_domicilio (37, 'Ave. S', 10, 110201);
add_domicilio (1009, 'Ave. Dinero', 12, 120101);
add_domicilio (4235, 'Calle Bilingüe', 9, 120201);
add_domicilio (5344, 'Calle Rubí', 3, 120301);
add_domicilio (468, 'Ave. Felicidad', 5, 130101);
add_domicilio (89, 'Calle Tropical', 1, 130201);
add_domicilio (743, 'Ave. Mora', 6, 130301);
END;
/
```

BEGIN

```
add_sala ('SBT-1H1', '1-H1', 3, 8);
add_sala ('SBT-1H2', '1-H2', 2, 6);
add_sala ('SCL-2H1', '2-H1', 4, 4);
add_sala ('SCL-2H2', '2-H2', 3, 4);
add_sala ('SCO-3H1', '3-H1', 4, 5);
add_sala ('SCO-3H2', '3-H2', 3, 6);
add_sala ('SCH-4H1', '4-H1', 4, 6);
add_sala ('SCH-4H2', '4-H2', 2, 4);
add_sala ('SDA-5H1', '5-H1', 2, 8);
```

```

add_sala ('SHE-6H1', '6-H1', 7, 6);
add_sala ('SLS-7H1', '7-H1', 4, 4);
add_sala ('SPA-8H1', '8-H1', 12, 6);
add_sala ('SPA-8H2', '8-H2', 10, 4);
add_sala ('SPA-8H3', '8-H3', 11, 4);
add_sala ('SPA-8H4', '8-H4', 1, 2);
add_sala ('SPA-8H5', '8-H5', 6, 6);
add_sala ('SVE-9H1', '9-H1', 4, 6);
add_sala ('SVE-9H2', '9-H2', 3, 6);
add_sala ('SPO-13H1', '13-H1', 4, 6);
add_sala ('SPO-13H2', '13-H2', 2, 4);
add_sala ('SNG-12H1', '12-H1', 2, 12);
add_sala ('SGY-10H1', '10-H1', 2, 10);
add_sala ('UCISPA-8H5', '8-H5', 2, 5);
add_sala ('UCISPA-8H1', '8-H1', 2, 2);
add_sala ('UCISPA-8H3', '8-H3', 3, 4);
END;
/

```

BEGIN

```

add_persona ('8-956-231', 'Lisa', 'Cortés', 'F', TO_DATE('29/12/1982','DD/MM/YYYY'), 90, null, null);
add_persona ('1-356-751', 'Melisa', 'Castillo', 'F', TO_DATE('13/10/1967', 'DD/MM/YYYY'), 1567,
'SBT-1H1', '1-H1');
add_persona ('2-437-769', 'Hector', 'González', 'M', TO_DATE('10/01/1993','DD/MM/YYYY'), 6489,
null, null);
add_persona ('3-379-451', 'José', 'Magdaleno', 'M', TO_DATE('27/05/1997','DD/MM/YYYY'), 560,
'SCO-3H1', '3-H1');
add_persona ('4-587-998', 'Victoria', 'Pérez', 'F', TO_DATE('24/03/1953','DD/MM/YYYY'), 889, null,
null);
add_persona ('5-187-7348', 'Ana', 'Ortiz', 'F', TO_DATE('29/12/1972','DD/MM/YYYY'), 976, 'SDA-
5H1', '5-H1');
add_persona ('6-130-643', 'Rodrigo', 'Rodríguez', 'M', TO_DATE('20/10/1964','DD/MM/YYYY'), 57,
null, null);
add_persona ('7-457-32', 'Martín', 'Márquez', 'M', TO_DATE('12/10/1970','DD/MM/YYYY'), 245, 'SLS-
7H1', '7-H1');
add_persona ('8-587-949', 'Marquela', 'Pinzón', 'F', TO_DATE('2/01/1991','DD/MM/YYYY'), 332, null,
null);
add_persona ('9-516-420', 'Paola', 'Jones', 'F', TO_DATE('23/10/1999','DD/MM/YYYY'), 76, 'SVE-
9H1', '9-H1');
add_persona ('8-950-949', 'Victor', 'Jimenez', 'M', TO_DATE('4/07/1995','DD/MM/YYYY'), 332, null,
null);
add_persona ('8-886-720', 'Román', 'Polanco', 'M', TO_DATE('7/03/1975','DD/MM/YYYY'), 90,
'UCISPA-8H3', '8-H3');
add_persona ('8-936-600', 'Sofía', 'Moreno', 'F', TO_DATE('15/12/1959','DD/MM/YYYY'), 468, 'SPO-
13H1', '13-H1');
add_persona ('3-714-830', 'Alice', 'Rojas', 'F', TO_DATE('6/12/1997','DD/MM/YYYY'), 789, 'SCO-
3H2', '3-H2');
add_persona ('12-450-92', 'Victoriano', 'Montezuma', 'M', TO_DATE('13/11/1984','DD/MM/YYYY'),
1009, null, null);
add_persona ('1-438-360', 'Casimiro', 'Bermudez', 'M', TO_DATE('10/11/1990','DD/MM/YYYY'),
1726, 'SBT-1H2', '1-H2');
add_persona ('2-571-457', 'Carmen', 'Alvarado', 'F', TO_DATE('24/06/1978','DD/MM/YYYY'), 8623,
'SCL-2H2', '2-H2');
add_persona ('4-614-730', 'Kathy', 'Arosemena', 'F', TO_DATE('27/02/1987','DD/MM/YYYY'), 889,
'SCH-4H2', '4-H2');

```

```
add_persona ('5-411-250', 'Daniela', 'López', 'F', TO_DATE('19/05/1992','DD/MM/YYYY'), 432, 'SDA-5H1', '5-H1');
add_persona ('6-248-581', 'Jonathan', 'De La Flor', 'M', TO_DATE('31/08/1996','DD/MM/YYYY'), 25, 'SHE-6H1', '6-H1');
END;
/
```

```
BEGIN
Add_doctor_hospital (1, '3-H1');
Add_doctor_hospital (2, '3-H2');
Add_doctor_hospital (3, '7-H1');
Add_doctor_hospital (4, '9-H1');
Add_doctor_hospital (5, '9-H2');
Add_doctor_hospital (6, '12-H1');
Add_doctor_hospital (7, '1-H1');
Add_doctor_hospital (8, '1-H2');
Add_doctor_hospital (9, '6-H1');
Add_doctor_hospital (10, '4-H1');
Add_doctor_hospital (11, '4-H2');
Add_doctor_hospital (12, '8-H1');
Add_doctor_hospital (13, '8-H2');
Add_doctor_hospital (14, '8-H3');
Add_doctor_hospital (15, '8-H4');
Add_doctor_hospital (16, '8-H5');
Add_doctor_hospital (17, '5-H1');
Add_doctor_hospital (18, '2-H1');
Add_doctor_hospital (19, '2-H2');
Add_doctor_hospital (20, '10-H1');
Add_doctor_hospital (21, '13-H1');
Add_doctor_hospital (22, '13-H2');
END;
/
```

```
BEGIN
add_persona_sintoma ('F', 1);
add_persona_sintoma ('DG', 1);
add_persona_sintoma ('DR', 2);
add_persona_sintoma ('F', 2);
add_persona_sintoma ('TS', 3);
add_persona_sintoma ('DC', 4);
add_persona_sintoma ('F', 5);
add_persona_sintoma ('P', 6);
add_persona_sintoma ('E', 7);
add_persona_sintoma ('DG', 7);
add_persona_sintoma ('DR', 8);
add_persona_sintoma ('CO', 9);
add_persona_sintoma ('DM', 10);
add_persona_sintoma ('D', 11);
add_persona_sintoma ('C', 12);
add_persona_sintoma ('E', 12);
add_persona_sintoma ('IH', 13);
add_persona_sintoma ('TS', 14);
add_persona_sintoma ('F', 15);
add_persona_sintoma ('CO', 15);
END;
/
```

```

BEGIN
add_persona_ep ('CAN', 1, to_date('10/3/2019','dd/mm/yyyy'));
add_persona_ep ('DIA', 2, to_date('12/5/2015','dd/mm/yyyy'));
add_persona_ep ('ECA', 3, to_date('13/4/2017','dd/mm/yyyy'));
add_persona_ep ('EPU', 4, to_date('30/1/2018','dd/mm/yyyy'));
add_persona_ep ('HIP', 5, to_date('10/8/2019','dd/mm/yyyy'));
add_persona_ep ('CAN', 6, to_date('31/10/2018','dd/mm/yyyy'));
add_persona_ep ('DIA', 7, to_date('12/3/2016','dd/mm/yyyy'));
add_persona_ep ('ECA', 8, to_date('10/5/2017','dd/mm/yyyy'));
add_persona_ep ('EPU', 9, to_date('24/6/2019','dd/mm/yyyy'));
add_persona_ep ('HIP', 10, to_date('16/9/2017','dd/mm/yyyy'));
END;
/

```

```

BEGIN
add_persona_prueba (1, 'H', to_date('23/4/2020','dd/mm/yyyy'));
add_persona_prueba (2, 'H', to_date('30/6/2020','dd/mm/yyyy'));
add_persona_prueba (3, 'S', to_date('12/5/2020','dd/mm/yyyy'));
add_persona_prueba (4, 'H', to_date('7/6/2020','dd/mm/yyyy'));
add_persona_prueba (5, 'S', to_date('13/6/2020','dd/mm/yyyy'));
add_persona_prueba (6, 'H', to_date('10/5/2020','dd/mm/yyyy'));
add_persona_prueba (7, 'S', to_date('8/5/2020','dd/mm/yyyy'));
add_persona_prueba (8, 'H', to_date('30/5/2020','dd/mm/yyyy'));
add_persona_prueba (9, 'S', to_date('12/4/2020','dd/mm/yyyy'));
add_persona_prueba (10, 'H', to_date('25/5/2020','dd/mm/yyyy'));
add_persona_prueba (11, 'S', to_date('9/6/2020','dd/mm/yyyy'));
add_persona_prueba (12, 'H', to_date('4/7/2020','dd/mm/yyyy'));
add_persona_prueba (13, 'H', to_date('23/3/2020','dd/mm/yyyy'));
add_persona_prueba (14, 'S', to_date('18/4/2020','dd/mm/yyyy'));
add_persona_prueba (15, 'H', to_date('29/6/2020','dd/mm/yyyy'));
END;
/

```

```

BEGIN
Add_telefono_h ('10-H1', '345-8910');
Add_telefono_h ('12-H1', '268-0943');
Add_telefono_h ('13-H1', '254-8926');
Add_telefono_h ('13-H2', '253-3447');
Add_telefono_h ('1-H1', '774-5689');
Add_telefono_h ('1-H2', '765-8910');
Add_telefono_h ('2-H1', '945-8543');
Add_telefono_h ('2-H2', '998-3219');
Add_telefono_h ('3-H1', '445-5432');
Add_telefono_h ('3-H2', '447-5617');
Add_telefono_h ('4-H1', '785-8532');
Add_telefono_h ('4-H2', '735-3210');
Add_telefono_h ('5-H1', '245-5670');
Add_telefono_h ('6-H1', '995-6789');
Add_telefono_h ('7-H1', '996-7480');
Add_telefono_h ('8-H1', '507-5830');
Add_telefono_h ('8-H2', '305-6300');
Add_telefono_h ('8-H3', '360-7400');
Add_telefono_h ('8-H4', '399-8082');
Add_telefono_h ('8-H5', '503-6600');
Add_telefono_h ('9-H1', '985-4287');
Add_telefono_h ('9-H2', '978-0045');
END;

```


/

```
BEGIN
add_tipo_telefono_p (1, 'Residencial');
add_tipo_telefono_p (2, 'Celular');
END;
/
```

Vistas

- **Prov_Dom**

```
create view Prov_Dom as select d.num_domic, c.nombre_correg, p.nombre_prov,p.Cod_prov
from Domicilio d join Corregimiento c
on d.Cod_correg=c.Cod_correg join Distrito dis
on dis.Cod_dist=c.Cod_dist join Provincia p
on p.Cod_prov=dis.Cod_prov;
```

Esta vista permite observar el nombre del domicilio junto con el distrito y a la provincia que pertenecen.

- **Per_Dom**

```
create view Per_Dom as select distinct Cedula_p, apellido_p, nombre_p, Num_domic
from Persona
order by apellido_p;
```

Esta consulta muestra la cédula, el nombre y el número de domicilio del paciente. Esta consulta está dirigida para los analistas de datos del MINSA ya que ellos requirieron estos datos para ir contabilizando la cantidad de personas que se van infectando cada día para realizar un reporte diario.

- **Sin**

```
create view Sin as select per.apellido_p, nombre_p as, sim.nombre_sint, per.Cedula_p
from Sintomas sim join Persona_Sintoma persim
on sim.Cod_sint = persim.Cod_sint join Persona per
on per.id_persona = persim.id_persona
where sim.nombre_sint in ('Fiebre');
```

Un epidemiólogo puede utilizar esta consulta para ver todos los pacientes que han ingresado con fiebre y hacer una estadística de cuál es la cantidad de la población afectada por este síntoma y determinar si hay algún cambio en futuras pruebas, ya que es posible estén utilizando tratamientos y observar cómo es el comportamiento de las personas en cuanto a la fiebre y poder indicar algún factor en cuanto al tratamiento en cuestión.

- **Prov_per_dom**

```
create view Prov_per_dom as select per.apellido_p, nombre_p
from Provincia pro join Distrito dis
on pro.Cod_prov = dis.Cod_prov join Corregimiento cor
on dis.Cod_dist = cor.Cod_dist join Domicilio dom
on cor.Cod_correg = dom.Cod_correg join Persona per
on per.Num_domic = dom.Num_domic join Persona_Sintoma pers
```

```
on per.id_persona = pers.id_persona join Persona_Enfermedades_Previas peref  
on pers.id_persona = peref.id_persona  
where pro.Cod_prov = 8 and pers.Cod_sint not in ('F', 'TS') or peref.Cod_enf = 'DIA';
```

Esta consulta pudiese ser útil para un Epidemiólogo, para poder conocer el comportamiento del virus si se seleccionan dos personas de una provincia determinada, en este caso, Panamá, que no tengan dos de los síntomas más comunes, los cuales son tos seca y fiebre, y además, que tengan o no alguna enfermedad previa, en este caso, diabetes, y así puede realizarles los estudios pertinentes.

- **Prov_per**

```
create view Prov_per as select per.apellido_p, nombre_p, pro.nombre_prov  
from Provincia pro join Distrito dis  
on pro.Cod_prov = dis.Cod_prov join Corregimiento cor  
on dis.Cod_dist = cor.Cod_dist join Domicilio dom  
on cor.Cod_correg = dom.Cod_correg join Persona per  
on per.Num_domic = dom.Num_domic  
where pro.Cod_prov between 1 and 5;
```

Esta consulta muestra el nombre del paciente y el nombre de la provincia en que se encuentra, con la condición de que estas provincias estén entre Bocas del Toro, Coclé, Colón, Chiriquí y Herrera. Esta consulta está dirigida principalmente al ministro de salud, ya que si este quiere tener un informe más claro de cómo está la situación en estas provincias en específico.

ANEXO

Creación de Tablas

```
SQL> create table Enfermedades_previas (  
2   Cod_enf varchar2 (15),  
3   constraint Enf_prev_pk primary key (Cod_enf),  
4   nombre_enf varchar2 (30) not null  
5   );  
  
Table created.  
  
SQL>  
SQL> create table Prueba(  
2   Cod_prueba varchar2 (15),  
3   constraint Prueba_Cod_prueba_pk primary key (Cod_prueba),  
4   tipo_prueba varchar2 (25) not null  
5   );  
  
Table created.  
  
SQL>  
SQL> create table Hospital(  
2   Cod_h varchar2 (15),  
3   constraint Hospital_Cod_h_pk primary key (Cod_h),  
4   nombre_h varchar2 (30) not null,  
5   direccion_h varchar2 (30) not null  
6   );  
  
Table created.  
  
SQL>  
SQL> create table Doctor(  
2   id_doctor number,  
3   Cedula_d varchar2 (15) not null,  
4   constraint Doctor_id_pk primary key (id_doctor),  
5   nombre_d varchar2 (25) not null,  
6   apellido_d varchar2 (25) not null  
7   );  
  
Table created.  
  
SQL>  
SQL> create table Sintomas (  
2   Cod_sint varchar2 (15),  
3   constraint Sintomas_cod_pk primary key (Cod_sint),  
4   nombre_sint varchar2 (30) not null  
5   );  
  
Table created.  
  
SQL>  
SQL> create table Provincia (  
2   Cod_prov number,  
3   constraint Provincia_Cod_prov_pk primary key (Cod_prov),  
4   nombre_prov varchar2 (20) not null  
5   );
```

```

SQL> create table Distrito(
2   Cod_dist number,
3   constraint Distrito_Cod_distrito_pk primary key (Cod_dist),
4   nombre_dist varchar2 (20) not null,
5   Cod_prov number not null,
6   constraint Distrito_Cod_prov_fk foreign key (Cod_prov) references Provincia (Cod_prov)
7 );

Table created.

SQL>
SQL> create table Corregimiento(
2   Cod_correg number,
3   constraint Corregimiento_Cod_correg_pk primary key (Cod_correg),
4   nombre_correg varchar2 (20) not null,
5   Cod_dist number not null,
6   constraint Corregimiento_Cod_dist_fk foreign key (Cod_dist) references Distrito (Cod_dist)
7 );

Table created.

SQL>
SQL> create table Domicilio (
2   Num_domic number,
3   constraint Domicilio_Num_domic_pk primary key (Num_domic),
4   direccion_d varchar2 (40) not null,
5   num_pers_viv number not null,
6   Cod_correg number not null,
7   constraint Domicilio_Cod_correg_fk foreign key (Cod_correg) references Corregimiento (Cod_correg)
8 );

Table created.

SQL>
SQL> create table Sala(
2   Cod_sala varchar2 (20),
3   Cod_h varchar2 (15),
4   constraint Sala_Cod_h_fk foreign key (Cod_h) references Hospital (Cod_h),
5   constraint Sala_Hospital_cod primary key (Cod_sala, Cod_h),
6   piso number not null,
7   num_camras number not null
8 );

Table created.

```

```

SQL> create table Persona(
2   id_persona number,
3   Cedula_p varchar2 (20) not null,
4   constraint Persona_id_p_pk primary key (id_persona),
5   nombre_p varchar2 (20) not null,
6   apellido_p varchar2 (20) not null,
7   sexo char not null,
8   fecha_nac date not null,
9   edad number,
10  Num_domic number not null,
11  constraint Persona_Num_domic_fk foreign key (Num_domic) references Domicilio (Num_domic),
12  Cod_sala varchar2 (20),
13  Cod_h varchar2 (15),
14  constraint infect_pers_fk foreign key (Cod_sala, Cod_h) references Sala (Cod_sala, Cod_h)
15 );

Table created.

SQL>
SQL> create table Doctor_Hospital (
2   id_doctor number,
3   constraint Doctor_h_id_fk foreign key (id_doctor) references Doctor (id_doctor),
4   Cod_h varchar2 (15),
5   constraint Doctor_h_cod_fk foreign key (Cod_h) references Hospital (Cod_h),
6   constraint Doctor_h_cod_id_pk primary key (id_doctor, Cod_h)
7 );

Table created.

SQL>
SQL> create table Persona_Sintoma(
2   Cod_sint varchar2 (15),
3   constraint Sintomas_cod_fk foreign key (Cod_sint) references Sintomas (Cod_sint),
4   id_persona number,
5   constraint Persona_Sintoma_id_fk foreign key (id_persona) references Persona (id_persona),
6   constraint Persona_sintoma_pk primary key (Cod_sint, id_persona)
7 );

Table created.

SQL>
SQL> create table Persona_Enfermedades_Previas(
2   Cod_enf varchar2 (15) not null,
3   constraint Persona_Enf_Previas_Cod_enf_fk foreign key (Cod_enf) references Enfermedades_previas (Cod_enf),
4   id_persona number not null,
5   constraint Persona_Enf_Previas_fk foreign key (id_persona) references Persona (id_persona),
6   constraint Cod_enf_Cedula_p_pk primary key (Cod_enf, id_persona),
7   fecha_estudio date not null
8 );

Table created.

```

```

SQL> create table Persona_Prueba(
  2 id_persona number,
  3 constraint Persona_prueba_fk foreign key (id_persona) references Persona (id_persona),
  4 Cod_prueba varchar2 (15),
  5 constraint Persona_Prueba_cod_fk foreign key (Cod_prueba) references Prueba (Cod_prueba),
  6 constraint Persona_prueba_pk primary key (id_persona, Cod_prueba),
  7 fec_d date not null
  8 );

```

Table created.

```

SQL>
SQL> create table Telefono_h(
  2 Cod_h varchar2(15),
  3 constraint Telefono_Cod_h_fk foreign key (Cod_h)
  4 references Hospital (Cod_h),
  5 constraint cod_h_pk primary key (Cod_h),
  6 num_tel_h varchar2 (20)
  7 );

```

Table created.

```

SQL>
SQL> create table Tipo_Telefono_p (
  2 ID_telefono_p number not null,
  3 tipo_telefono_p varchar2 (20) not null,
  4 constraint ID_telefono_pk primary key (ID_telefono_p)
  5 );

```

Table created.

```

SQL>
SQL> create table Telefono_p(
  2 ID_telefono_p number not null,
  3 id_persona number,
  4 constraint ID_tel_fk foreign key (ID_telefono_p)
  5 references Tipo_Telefono_p (ID_telefono_p),
  6 constraint Telefono_p_id_persona_fk foreign key (id_persona)
  7 references Persona (id_persona),
  8 constraint ID_telef_cd_pk primary key (ID_telefono_p, id_persona),
  9 num_tel_p varchar2(25)
  10 );

```

Table created.

```

SQL>
SQL> create table per_stat(
  2 numstad number not null,
  3 id_persona number not null,
  4 cedula varchar2(20) not null,
  5 nombre varchar2(20) not null,
  6 apellido varchar2(20) not null,
  7 edad number,
  8 estado varchar2(35)
  9 );

```

```

SQL> Create table Control_Infect(
  2 N_inf number,
  3 Id_pers number,
  4 cedula_p varchar2 (20) not null,
  5 nombre_p varchar2 (20) not null,
  6 apellido_p varchar2 (20) not null,
  7 edad number,
  8 fec_d date not null,
  9 usuario varchar2(15)
  10 );

```

Table created.

```

SQL>
SQL> create table statistic(
  2 hombre number,
  3 mujer number,
  4 edada number,
  5 edadb number,
  6 edadc number
  7 );

```

Table created.

```

SQL>
SQL> Create table Control_stat(
  2 mujer number,
  3 hombre number,
  4 edada number,
  5 edadb number,
  6 edadc number,
  7 fecha date,
  8 usuario varchar2(15)
  9 );

```

Table created.

Creación de Secuencias

```
SQL> CREATE SEQUENCE seq_doc
  2  START WITH 1
  3  INCREMENT BY 1;

Sequence created.

SQL>
SQL> CREATE SEQUENCE seq_pers
  2  START WITH 1
  3  INCREMENT BY 1;

Sequence created.

SQL>
SQL> create sequence trig_per
  2  start with 1
  3  increment by 1;

Sequence created.

SQL>
SQL> create sequence trig_stad
  2  start with 1
  3  increment by 1;

Sequence created.

SQL>
SQL> create sequence stat
  2  start with 1
  3  increment by 1;

Sequence created.
```

Creación de Funciones

```
SQL> CREATE OR REPLACE FUNCTION calc_edad (fecha_nac PERSONA.fecha_nac%TYPE)
2 RETURN NUMBER AS
3 BEGIN
4 RETURN FLOOR (months_between (sysdate, fecha_nac)/12);
5 END calc_edad;
6 /
```

Function created.

```
SQL>
SQL> CREATE OR REPLACE FUNCTION stat_male
2 RETURN NUMBER AS
3 v_hom NUMBER;
4 v_total NUMBER;
5 v_resp NUMBER;
6 BEGIN
7 SELECT count(sexo) INTO v_hom FROM Persona
8 WHERE sexo='M';
9 SELECT count(sexo) INTO v_total FROM Persona;
10 v_resp:=(v_hom/v_total)*100;
11 RETURN v_resp;
12 END stat_male;
13 /
```

Function created.

```
SQL>
SQL> CREATE OR REPLACE FUNCTION stat_female
2 RETURN NUMBER AS
3 v_muj NUMBER;
4 v_total NUMBER;
5 v_resp NUMBER;
6 BEGIN
7 SELECT count(sexo) INTO v_muj FROM Persona
8 WHERE sexo='F';
9 SELECT count(sexo) INTO v_total FROM Persona;
10 v_resp:=(v_muj/v_total)*100;
11 RETURN v_resp;
12 END stat_female;
13 /
```

Function created.

```
SQL> CREATE OR REPLACE FUNCTION population(
2 p_edad persona.edad%TYPE)
3 RETURN VARCHAR2 AS
4 BEGIN
5 IF p_edad >= 60 THEN
6 RETURN 'Riesgo Alto';
7 ELSIF p_edad >= 30 THEN
8 RETURN 'Riesgo Regular';
9 ELSE
10 RETURN 'Riesgo Bajo';
11 END IF;
12 END population;
13 /
```

Function created.

```

SQL> CREATE OR REPLACE FUNCTION stat_edadb
  2 RETURN NUMBER AS
  3 v_edadc NUMBER;
  4 v_total NUMBER;
  5 v_resp NUMBER;
  6 BEGIN
  7 SELECT count(edad) INTO v_edadc FROM Persona
  8 WHERE edad>=30 AND edad<60;
  9 SELECT count(*) INTO v_total FROM Persona;
 10 v_resp:=(v_edadc/v_total)*100;
 11 RETURN v_resp;
 12 END stat_edadb;
 13 /

```

Function created.

```

SQL>
SQL> CREATE OR REPLACE FUNCTION stat_edadc
  2 RETURN NUMBER AS
  3 v_edadc NUMBER;
  4 v_total NUMBER;
  5 v_resp NUMBER;
  6 BEGIN
  7 SELECT count(edad) INTO v_edadc FROM Persona
  8 WHERE edad<30;
  9 SELECT count(id_persona) INTO v_total FROM Persona;
 10 v_resp:=(v_edadc/v_total)*100;
 11 RETURN v_resp;
 12 END stat_edadc;
 13 /

```

Function created.

```

SQL>
SQL> CREATE OR REPLACE FUNCTION stat_edada
  2 RETURN NUMBER AS
  3 v_edadc NUMBER;
  4 v_total NUMBER;
  5 v_resp NUMBER;
  6 BEGIN
  7 SELECT count(edad) INTO v_edadc FROM Persona
  8 WHERE edad>=60;
  9 SELECT count(id_persona) INTO v_total FROM Persona;
 10 v_resp:=(v_edadc/v_total)*100;
 11 RETURN v_resp;
 12 END stat_edada;
 13 /

```

Function created.

Creación de Triggers

```
SQL> CREATE OR REPLACE TRIGGER per_pac
2  BEFORE UPDATE OR DELETE OR INSERT ON Persona
3  FOR EACH ROW
4  DECLARE
5      v_Control CHAR(1);
6  BEGIN
7      IF UPDATING THEN
8          v_Control := 'U';
9          INSERT INTO Control_Infect (n_inf, id_pers, cedula_p, nombre_p, apellido_p, edad, fec_d, usuario)
10         VALUES (trig_per.nextval, :NEW.id_persona, :NEW.cedula_p, :NEW.nombre_p, :new.apellido_p, :new.edad, sysdate, user);
11      ELSIF INSERTING THEN
12          v_Control := 'I';
13          INSERT INTO Control_Infect (n_inf, id_pers, cedula_p, nombre_p, apellido_p, edad, fec_d, usuario)
14         VALUES (trig_per.nextval, :NEW.id_persona, :NEW.cedula_p, :NEW.nombre_p, :new.apellido_p, :new.edad, sysdate, user);
15      ELSIF DELETING THEN
16          v_Control := 'D';
17          INSERT INTO Control_Infect (n_inf, id_pers, cedula_p, nombre_p, apellido_p, edad, fec_d, usuario)
18         VALUES (trig_per.nextval, :NEW.id_persona, :NEW.cedula_p, :NEW.nombre_p, :new.apellido_p, :new.edad, sysdate, user);
19      END IF;
20  END per_pac;
21  /

Trigger created.

SQL>
SQL> CREATE OR REPLACE TRIGGER upd_static
2  BEFORE INSERT OR UPDATE ON Statistic
3  FOR EACH ROW
4  DECLARE
5      v_Control CHAR(1);
6  BEGIN
7      IF UPDATING THEN
8          v_Control := 'U';
9          INSERT INTO control_stat
10         values (:NEW.mujer, :NEW.hombre, :NEW.edada, :NEW.edadb, :NEW.edadc, sysdate, user);
11      ELSIF INSERTING THEN
12          v_Control := 'I';
13          INSERT INTO control_stat
14         values (:NEW.mujer, :NEW.hombre, :NEW.edada, :NEW.edadb, :NEW.edadc, sysdate, user);
15      END IF;
16  END upd_static;
17  /

Trigger created.
```

Creación de Procedimientos

```
SQL> CREATE OR REPLACE PROCEDURE add_enfermedad (  
2  p_code enfermedades_previas.cod_enf%TYPE,  
3  p_nombrenf enfermedades_previas.nombre_enf%TYPE  
4  ) AS  
5  BEGIN  
6  INSERT INTO Enfermedades_previas (cod_enf, nombre_enf)  
7  values (p_code, p_nombrenf);  
8  EXCEPTION  
9  WHEN NO_DATA_FOUND THEN  
10 DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
11 WHEN DUP_VAL_ON_INDEX THEN  
12 DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
13 WHEN OTHERS THEN  
14 DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
15 COMMIT;  
16 END add_enfermedad;  
17 /
```

Procedure created.

```
SQL>  
SQL> CREATE OR REPLACE PROCEDURE add_prueba (  
2  p_codpb prueba.cod_prueba%TYPE,  
3  p_tipop prueba.tipo_prueba%TYPE  
4  ) AS  
5  BEGIN  
6  INSERT INTO Prueba (cod_prueba, tipo_prueba)  
7  values (p_codpb, p_tipop);  
8  EXCEPTION  
9  WHEN NO_DATA_FOUND THEN  
10 DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
11 WHEN DUP_VAL_ON_INDEX THEN  
12 DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
13 WHEN OTHERS THEN  
14 DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
15 COMMIT;  
16 END add_prueba;  
17 /
```

Procedure created.

```
SQL> CREATE OR REPLACE PROCEDURE add_hospital (  
2  p_codh hospital.cod_h%TYPE,  
3  p_nombreh hospital.nombre_h%TYPE,  
4  p_direccionh hospital.direccion_h%TYPE  
5  ) AS  
6  BEGIN  
7  INSERT INTO Hospital (cod_h, nombre_h, direccion_h)  
8  values (p_codh, p_nombreh, p_direccionh);  
9  EXCEPTION  
10 WHEN NO_DATA_FOUND THEN  
11 DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
12 WHEN DUP_VAL_ON_INDEX THEN  
13 DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
14 WHEN OTHERS THEN  
15 DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
16 COMMIT;  
17 END add_hospital;  
18 /
```

Procedure created.

```
SQL>  
SQL> CREATE OR REPLACE PROCEDURE add_doctor (  
2  p_cedulad doctor.cedula_d%TYPE,  
3  p_nombred doctor.nombre_d%TYPE,  
4  p_apellido doctor.apellido_d%TYPE  
5  ) AS  
6  BEGIN  
7  INSERT INTO Doctor (id_doctor, cedula_d, nombre_d, apellido_d)  
8  values (seq_doc.nextval, p_cedulad, p_nombred, p_apellido_d);  
9  EXCEPTION  
10 WHEN NO_DATA_FOUND THEN  
11 DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');  
12 WHEN DUP_VAL_ON_INDEX THEN  
13 DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');  
14 WHEN OTHERS THEN  
15 DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');  
16 COMMIT;  
17 END add_doctor;  
18 /
```

Procedure created.

```

SQL> CREATE OR REPLACE PROCEDURE add_sintoma (
  2  p_cods sintomas.cod_sint%TYPE,
  3  p_nombres sintomas.nombre_sint%TYPE
  4  ) AS
  5  BEGIN
  6  INSERT INTO Sintomas (cod_sint, nombre_sint)
  7  values (p_cods, p_nombres);
  8  EXCEPTION
  9  WHEN NO_DATA_FOUND THEN
 10  DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
 11  WHEN DUP_VAL_ON_INDEX THEN
 12  DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
 13  WHEN OTHERS THEN
 14  DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
 15  COMMIT;
 16  END add_sintoma;
 17  /

```

Procedure created.

```

SQL>
SQL> CREATE OR REPLACE PROCEDURE add_provincia (
  2  p_codp provincia.cod_prov%TYPE,
  3  p_nombrep provincia.nombre_prov%TYPE
  4  ) AS
  5  BEGIN
  6  INSERT INTO Provincia (cod_prov, nombre_prov)
  7  values (p_codp, p_nombrep);
  8  EXCEPTION
  9  WHEN NO_DATA_FOUND THEN
 10  DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
 11  WHEN DUP_VAL_ON_INDEX THEN
 12  DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
 13  WHEN OTHERS THEN
 14  DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
 15  COMMIT;
 16  END add_provincia;
 17  /

```

Procedure created.

```

SQL> CREATE OR REPLACE PROCEDURE add_distrito (
  2  p_coddiss distrito.cod_dist%TYPE,
  3  p_nombredis distrito.nombre_dist%TYPE,
  4  p_codprov distrito.cod_prov%TYPE
  5  ) AS
  6  BEGIN
  7  INSERT INTO Distrito (cod_dist, nombre_dist, cod_prov)
  8  values (p_coddiss, p_nombredis, p_codprov);
  9  EXCEPTION
 10  WHEN NO_DATA_FOUND THEN
 11  DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
 12  WHEN DUP_VAL_ON_INDEX THEN
 13  DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
 14  WHEN OTHERS THEN
 15  DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
 16  COMMIT;
 17  END add_distrito;
 18  /

```

Procedure created.

```

SQL>
SQL> CREATE OR REPLACE PROCEDURE add_correg (
  2  p_codcorr corregimiento.cod_correg%TYPE,
  3  p_nombreco corregimiento.nombre_correg%TYPE,
  4  p_codis corregimiento.cod_dist%TYPE
  5  ) AS
  6  BEGIN
  7  INSERT INTO Corregimiento (cod_correg, nombre_correg, cod_dist)
  8  values (p_codcorr, p_nombreco, p_codis);
  9  EXCEPTION
 10  WHEN NO_DATA_FOUND THEN
 11  DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
 12  WHEN DUP_VAL_ON_INDEX THEN
 13  DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
 14  WHEN OTHERS THEN
 15  DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
 16  COMMIT;
 17  END add_correg;
 18  /

```

Procedure created.

```

SQL> CREATE OR REPLACE PROCEDURE add_domicilio (
2  p_numdom domicilio.num_domic%TYPE,
3  p_direcciond domicilio.direccion_d%TYPE,
4  p_numpers domicilio.num_pers_viv%TYPE,
5  p_codcor domicilio.cod_correg%TYPE
6  ) AS
7  BEGIN
8  INSERT INTO Domicilio (num_domic, direccion_d, num_pers_viv, cod_correg)
9  values (p_numdom, p_direcciond, p_numpers, p_codcor);
10 EXCEPTION
11 WHEN NO_DATA_FOUND THEN
12 DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
13 WHEN DUP_VAL_ON_INDEX THEN
14 DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
15 WHEN OTHERS THEN
16 DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
17 COMMIT;
18 END add_domicilio;
19 /

```

Procedure created.

```

SQL>
SQL> CREATE OR REPLACE PROCEDURE add_persona (
2  p_cedulap persona.cedula_p%TYPE,
3  p_nombrep persona.nombre_p%TYPE,
4  p_apellidop persona.apellido_p%TYPE,
5  p_sexo persona.sexo%TYPE,
6  p_fechan persona.fecha_nac%TYPE,
7  p_numdom persona.num_domic%TYPE,
8  p_codsala persona.cod_sala%TYPE,
9  p_codh persona.cod_h%TYPE
10 ) AS
11 p_edad persona.edad%TYPE;
12 BEGIN
13 p_edad:=calc_edad(p_fechan);
14 INSERT INTO Persona (id_persona, cedula_p, nombre_p, apellido_p, sexo, fecha_nac, edad, num_domic, cod_sala, cod_h)
15 values (seq_pers.nextval, p_cedulap, p_nombrep, p_apellidop, p_sexo, p_fechan, p_edad, p_numdom, p_codsala, p_codh);
16 EXCEPTION
17 WHEN NO_DATA_FOUND THEN
18 DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
19 WHEN DUP_VAL_ON_INDEX THEN
20 DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
21 WHEN OTHERS THEN
22 DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
23 COMMIT;
24 END add_persona;
25 /

```

Procedure created.

```

SQL> CREATE OR REPLACE PROCEDURE Add_doctor_hospital (
2  p_id_doctor doctor_hospital.id_doctor%TYPE,
3  p_cod_h doctor_hospital.cod_h%TYPE
4  ) AS
5  BEGIN
6  INSERT INTO doctor_hospital (id_doctor, cod_h)
7  values (p_id_doctor, p_cod_h);
8  EXCEPTION
9  WHEN NO_DATA_FOUND THEN
10 DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
11 WHEN DUP_VAL_ON_INDEX THEN
12 DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
13 WHEN OTHERS THEN
14 DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
15 COMMIT;
16 END Add_doctor_hospital;
17 /

```

Procedure created.

```

SQL>
SQL> CREATE OR REPLACE PROCEDURE add_telefono_h (
2  p_cod_h telefono_h.cod_h%TYPE,
3  p_num_tel_h telefono_h.num_tel_h%TYPE
4  ) AS
5  BEGIN
6  INSERT INTO telefono_h (cod_h, num_tel_h)
7  values (p_cod_h, p_num_tel_h);
8  EXCEPTION
9  WHEN NO_DATA_FOUND THEN
10 DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
11 WHEN DUP_VAL_ON_INDEX THEN
12 DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
13 WHEN OTHERS THEN
14 DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
15 COMMIT;
16 END add_telefono_h;
17 /

```

Procedure created.

```

SQL> CREATE OR REPLACE PROCEDURE add_sala (
2   p_cod_sala sala.cod_sala%TYPE,
3   p_cod_h sala.cod_h%TYPE,
4   p_piso sala.piso%TYPE,
5   p_num_camas sala.num_camas%TYPE
6 ) AS
7 BEGIN
8   INSERT INTO sala (cod_sala, cod_h, piso, num_camas)
9   values (p_cod_sala, p_cod_h, p_piso, p_num_camas);
10 EXCEPTION
11 WHEN NO_DATA_FOUND THEN
12   DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
13 WHEN DUP_VAL_ON_INDEX THEN
14   DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
15 WHEN OTHERS THEN
16   DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
17 COMMIT;
18 END add_sala;
19 /

```

Procedure created.

```

SQL>
SQL> CREATE OR REPLACE PROCEDURE add_tipo_telefono_p (
2   p_id_telefono_p IN tipo_telefono_p.id_telefono_p%TYPE,
3   p_tipo_telefono_p IN tipo_telefono_p.tipo_telefono_p%TYPE
4 ) AS
5 BEGIN
6   INSERT INTO Tipo_Telefono_p (id_telefono_p, tipo_telefono_p)
7   VALUES (p_id_telefono_p, p_tipo_telefono_p);
8 EXCEPTION
9 WHEN NO_DATA_FOUND THEN
10  DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
11 COMMIT;
12 END add_tipo_telefono_p;
13 /

```

Procedure created.

```

SQL>
SQL> CREATE OR REPLACE PROCEDURE add_telefono_p (
2   p_id_telefono_p IN telefono_p.id_telefono_p%TYPE,
3   p_id_persona IN telefono_p.id_persona%TYPE,
4   p_num_tel_p IN telefono_p.num_tel_p%TYPE
5 ) AS
6 BEGIN
7   INSERT INTO telefono_p(id_telefono_p, id_persona, num_tel_p)
8   VALUES (p_id_telefono_p, p_id_persona, p_num_tel_p);
9 EXCEPTION
10  WHEN NO_DATA_FOUND THEN
11    DBMS_OUTPUT.PUT_LINE ('No se encontró la tabla');
12 COMMIT;
13 END add_telefono_p;
14 /

```

```

SQL> CREATE OR REPLACE PROCEDURE add_persona_sintoma (
  2  p_cod_sint persona_sintoma.cod_sint%TYPE,
  3  p_id_persona persona_sintoma.id_persona%TYPE
  4  ) AS
  5  BEGIN
  6  INSERT INTO Persona_sintoma (cod_sint, id_persona)
  7  Values (p_cod_sint, p_id_persona);
  8  EXCEPTION
  9  WHEN NO_DATA_FOUND THEN
 10  DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
 11  WHEN DUP_VAL_ON_INDEX THEN
 12  DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
 13  WHEN OTHERS THEN
 14  DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
 15  COMMIT;
 16  END add_persona_sintoma;
 17  /

```

Procedure created.

```

SQL>
SQL>
SQL> CREATE OR REPLACE PROCEDURE add_persona_prueba (
  2  p_id_persona persona_prueba.id_persona%TYPE,
  3  p_cod_prueba persona_prueba.cod_prueba%TYPE,
  4  p_fec_d persona_prueba.fec_d%TYPE
  5  ) AS
  6  BEGIN
  7  INSERT INTO Persona_prueba (id_persona, cod_prueba, fec_d)
  8  values (p_id_persona, p_cod_prueba, p_fec_d);
  9  EXCEPTION
 10  WHEN NO_DATA_FOUND THEN
 11  DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
 12  WHEN DUP_VAL_ON_INDEX THEN
 13  DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
 14  WHEN OTHERS THEN
 15  DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
 16  COMMIT;
 17  END add_persona_prueba;
 18  /

```

Procedure created.

```

SQL> CREATE OR REPLACE PROCEDURE add_persona_ep (
  2  p_cod_enf persona_enfermedades_previas.cod_enf%TYPE,
  3  p_id_persona persona_enfermedades_previas.id_persona%TYPE,
  4  p_fecha_estudio persona_enfermedades_previas.fecha_estudio%TYPE
  5  ) AS
  6  BEGIN
  7  INSERT INTO Persona_enfermedades_previas (cod_enf, id_persona, fecha_estudio)
  8  values (p_cod_enf, p_id_persona, p_fecha_estudio);
  9  EXCEPTION
 10  WHEN NO_DATA_FOUND THEN
 11  DBMS_OUTPUT.PUT_LINE ('No se encontro la tabla');
 12  WHEN DUP_VAL_ON_INDEX THEN
 13  DBMS_OUTPUT.PUT_LINE ('Ya existe este registro');
 14  WHEN OTHERS THEN
 15  DBMS_OUTPUT.PUT_LINE ('No se insertaron los datos');
 16  COMMIT;
 17  END add_persona_ep;
 18  /

```

Procedure created.

```

SQL> create or replace PROCEDURE upd_stat
2 AS
3 p_hom statistic.hombre%TYPE;
4 p_muj statistic.mujer%TYPE;
5 p_edada statistic.edada%TYPE;
6 p_edadb statistic.edadb%TYPE;
7 p_edadc statistic.edadc%TYPE;
8 BEGIN
9 p_hom:=stat_male;
10 p_muj:=stat_female;
11 p_edada:=stat_edada;
12 p_edadb:=stat_edadb;
13 p_edadc:=stat_edadc;
14 INSERT INTO Statistic
15 values (p_hom, p_muj,p_edada,p_edadb,p_edadc);
16 EXCEPTION
17 WHEN NO_DATA_FOUND THEN
18 DBMS_OUTPUT.PUT_LINE ('No se encontro los datos');
19 COMMIT;
20 END upd_stat;
21 /

```

Procedure created.

```

SQL> CREATE OR REPLACE PROCEDURE act_age
2 AS
3 CURSOR c_edad IS
4 SELECT id_persona, cedula_p, nombre_p, apellido_p, edad
5 FROM Persona;
6 v_idp persona.id_persona%TYPE;
7 v_nombre persona.nombre_p%TYPE;
8 v_apellido persona.apellido_p%TYPE;
9 v_edad persona.edad%TYPE;
10 v_cedula persona.cedula_p%TYPE;
11 v_funcion per_stat.estado%TYPE;
12 BEGIN
13 OPEN c_edad;
14 LOOP
15 FETCH c_edad INTO v_idp, v_cedula, v_nombre, v_apellido, v_edad;
16 v_funcion:=population(v_edad);
17 INSERT INTO per_stat (numstad, id_persona, cedula, nombre, apellido, edad, estado)
18 values (stat.nextval,v_idp, v_cedula, v_nombre, v_apellido, v_edad, v_funcion);
19 EXIT WHEN c_edad%NOTFOUND;
20 END LOOP;
21 CLOSE c_edad;
22 EXCEPTION
23 WHEN NO_DATA_FOUND THEN
24 DBMS_OUTPUT.PUT_LINE ('No se encontró los datos');
25 COMMIT;
26 END act_age;
27 /

```

Procedure created.

Invocaciones

```
SQL> BEGIN
2  add_enfermedad ('HIP', 'Hipertensión');
3  add_enfermedad ('CAN', 'Cáncer');
4  add_enfermedad ('DIA', 'Diabetes');
5  add_enfermedad ('EPU', 'Enfermedades Pulmonares');
6  add_enfermedad ('ECA', 'Enfermedades Cardíacas');
7  END;
8  /

PL/SQL procedure successfully completed.

SQL>
SQL> BEGIN
2  add_prueba ('S', 'Prueba Serológica');
3  add_prueba ('H', 'Prueba por Hisopado');
4  END;
5  /

PL/SQL procedure successfully completed.

SQL>
SQL> BEGIN
2  add_hospital ('8-H1', 'Hospital Santo Tomás', 'Calle 37 Este');
3  add_hospital ('8-H2', 'Clínica Hospital San Fernando', 'Vía España');
4  add_hospital ('8-H3', 'Hospital Santa Fe', 'Ave. Simón Bolívar');
5  add_hospital ('8-H4', 'Hospital Modular', 'Albrook');
6  add_hospital ('8-H5', 'Complejo Dr. Arnulfo A. Madrid', 'Ave. Simón Bolívar');
7  add_hospital ('3-H1', 'Hospital Manuel A. Guerrero', 'Calle 11');
8  add_hospital ('3-H2', 'Hospital Colón 4 Altos', '4 Altos');
9  add_hospital ('4-H1', 'Hospital Chiriquí', 'Ave. 3ra y Calle Central');
10 add_hospital ('4-H2', 'Hospital Dr. Rafael Hernández', 'San Pablo Viejo');
11 add_hospital ('13-H1', 'Hospital Nicolás A. Solano', 'La Chorrera');
12 add_hospital ('13-H2', 'Clínica Hospital Panamericano', 'Calle Santa Rita');
13 add_hospital ('6-H1', 'Hospital Gustavo N. Collado', 'Ave. Rinaldo Camarano');
14 add_hospital ('7-H1', 'Hospital Anita Moreno', 'Ave. Dr. Belisario Porras');
15 add_hospital ('1-H1', 'Hospital Guillermo S. Borbón', 'Isla Colón');
16 add_hospital ('1-H2', 'Isla Colón Regional Hospital', 'Ave. H Norte');
17 add_hospital ('9-H1', 'Hospital Dr. Luis Fábrega', 'Santiago');
18 add_hospital ('9-H2', 'Policlínica Dr. Horacio Gómez', 'Santiago');
19 add_hospital ('5-H1', 'Hospital de Metetí', 'Metetí');
20 add_hospital ('2-H1', 'Hospital Aquilino Tejera', 'Panonomé');
21 add_hospital ('2-H2', 'Hospital Dr. Rafael Estévez', 'Aguadulce');
22 add_hospital ('12-H1', 'Hospital de San Pedrito', 'Ngöbe-Buglé');
23 add_hospital ('10-H1', 'Hospital Marvel Iglesia', 'Guna Yala');
24 END;
25 /

PL/SQL procedure successfully completed.
```

```
SQL> BEGIN
2  add_doctor ('8-365-873', 'Tatiana', 'Ojo');
3  add_doctor ('7-546-654', 'Vivian', 'Aguilar');
4  add_doctor ('8-234-432', 'Denisse', 'Thorne');
5  add_doctor ('8-465-543', 'Reynaldo', 'Castillero');
6  add_doctor ('3-65-673', 'Giselle', 'Balmaceda');
7  add_doctor ('4-100-657', 'Moisés', 'De Sedas');
8  add_doctor ('1-432-532', 'Jeannette', 'Johnson');
9  add_doctor ('2-348-43', 'Tommy', 'Liao');
10 add_doctor ('9-543-67', 'Andrés', 'Soto');
11 add_doctor ('6-234-6754', 'José', 'Lorenzo');
12 add_doctor ('5-123-890', 'Gustavo', 'Santamaría');
13 add_doctor ('3-234-567', 'Sunshine', 'Del Cid');
14 add_doctor ('9-321-789', 'Raúl', 'Mendoza');
15 add_doctor ('8-332-654', 'Italys', 'Ledezma');
16 add_doctor ('7-654-2134', 'Daniel', 'Araúz');
17 END;
18 /

PL/SQL procedure successfully completed.
```

```
SQL>
SQL> BEGIN
2  add_sintoma ('F', 'Fiebre');
3  add_sintoma ('TS', 'Tos seca');
4  add_sintoma ('C', 'Cansancio');
5  add_sintoma ('DG', 'Dolor de garganta');
6  add_sintoma ('DC', 'Dolor de cabeza');
7  add_sintoma ('D', 'Diarrea');
8  add_sintoma ('DM', 'Molestias y dolores');
9  add_sintoma ('CO', 'Conjuntivitis');
10 add_sintoma ('DR', 'Dificultad para respirar');
11 add_sintoma ('E', 'Erupciones cutáneas');
12 add_sintoma ('IH', 'Incapacidad para hablar');
13 add_sintoma ('P', 'Pérdida de olfato y gusto');
14 END;
15 /

PL/SQL procedure successfully completed.
```



```

SQL> BEGIN
  2  add_provincia (1, 'Bocas del Toro');
  3  add_provincia (2, 'Coclé');
  4  add_provincia (3, 'Colón');
  5  add_provincia (4, 'Chiriquí');
  6  add_provincia (5, 'Darién');
  7  add_provincia (6, 'Herrera');
  8  add_provincia (7, 'Los Santos');
  9  add_provincia (8, 'Panamá');
 10  add_provincia (9, 'Veraguas');
 11  add_provincia (10, 'Guna Yala');
 12  add_provincia (11, 'Emberá Wounaan');
 13  add_provincia (12, 'Ngäbe-buglé');
 14  add_provincia (13, 'Panamá Oeste');
 15  add_provincia(14, 'Madugandí');
 16  add_provincia (15, 'Wargandí');
 17  END;
 18  /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
  2  add_distrito (0101, 'Almirante', 1);
  3  add_distrito (0102, 'Bocas del Toro', 1);
  4  add_distrito (0103, 'Changuinola', 1);
  5  add_distrito (0201, 'Penonomé', 2);
  6  add_distrito (0202, 'Aguadulce', 2);
  7  add_distrito (0203, 'Antón', 2);
  8  add_distrito (0301, 'Colón', 3);
  9  add_distrito (0302, 'Chagres', 3);
 10  add_distrito (0303, 'Portobelo', 3);
 11  add_distrito (0401, 'David', 4);
 12  add_distrito (0402, 'San Félix', 4);
 13  add_distrito (0403, 'Barú', 4);
 14  add_distrito (0501, 'Chepigana', 5);
 15  add_distrito (0502, 'Pinogana', 5);
 16  add_distrito (0503, 'Santa Fe', 5);
 17  add_distrito (0601, 'Chitré', 6);
 18  add_distrito (0602, 'Ocú', 6);
 19  add_distrito (0603, 'Las Minas', 6);
 20  add_distrito (0701, 'Las Tablas', 7);
 21  add_distrito (0702, 'Tonosí', 7);
 22  add_distrito (0703, 'Guararé', 7);
 23  add_distrito (0801, 'San Miguelito', 8);
 24  add_distrito (0802, 'Panamá', 8);
 25  add_distrito (0803, 'Balboa', 8);
 26  add_distrito (0901, 'Santiago', 9);
 27  add_distrito (0902, 'Atalaya', 9);
 28  add_distrito (0903, 'Calobre', 9);
 29  add_distrito (1101, 'Cémaco', 11);
 30  add_distrito (1102, 'Sambú', 11);
 31  add_distrito (1201, 'Besikó', 12);
 32  add_distrito (1202, 'Jirondai', 12);
 33  add_distrito (1203, 'Kankintú', 12);
 34  add_distrito (1301, 'Arraiján', 13);
 35  add_distrito (1302, 'La Chorrera', 13);
 36  add_distrito (1303, 'San Carlos', 13);
 37  END;
 38  /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
2  add_correg (010101, 'Barrio Francés', 0101);
3  add_correg (010201, 'Bastimentos', 0102);
4  add_correg (010301, 'El Silencio', 0103);
5  add_correg (020101, 'Río Grande', 0201);
6  add_correg (020201, 'Pocri', 0202);
7  add_correg (020301, 'El Valle', 0203);
8  add_correg (030101, 'Cristóbal', 0301);
9  add_correg (030201, 'Piña', 0302);
10 add_correg (030301, 'Cacique', 0303);
11 add_correg (040101, 'San Carlos', 0401);
12 add_correg (040201, 'Las Lajas', 0402);
13 add_correg (040301, 'Puerto Armuelles', 0403);
14 add_correg (050101, 'La Palma', 0501);
15 add_correg (050201, 'Yaviza', 0502);
16 add_correg (050301, 'Agua Fria', 0503);
17 add_correg (060101, 'Monagrillo', 0601);
18 add_correg (060201, 'Llano Grande', 0602);
19 add_correg (060301, 'Chumical', 0603);
20 add_correg (070101, 'Valle Rico', 0701);
21 add_correg (070201, 'Cambutal', 0702);
22 add_correg (070301, 'Llano Abajo', 0703);
23 add_correg (080101, 'Omar Torrijos', 0801);
24 add_correg (080201, 'Don Bosco', 0802);
25 add_correg (080301, 'San Miguel', 0803);
26 add_correg (090101, 'Urracá', 0901);
27 add_correg (090201, 'San Antonio', 0902);
28 add_correg (090301, 'La Yeguada', 0903);
29 add_correg (110101, 'Lajas Blancas', 1101);
30 add_correg (110201, 'Río Sabalo', 1102);
31 add_correg (120101, 'Cerro Banco', 1201);
32 add_correg (120201, 'Man Creek', 1202);
33 add_correg (120301, 'Piedra Roja', 1203);
34 add_correg (130101, 'Vista Alegre', 1301);
35 add_correg (130201, 'El Coco', 1302);
36 add_correg (130301, 'Las Uvas', 1303);
37 END;
38 /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
2  add_domicilio (1803, 'Ave. Olmedo Solé', 7, 10101);
3  add_domicilio (1567, 'Calle Sea Wall', 4, 10201);
4  add_domicilio (1726, 'Sendero Caimán', 9, 10301);
5  add_domicilio (6489, 'Calle Verde', 3, 20101);
6  add_domicilio (8456, 'Calle Abajo', 3, 20201);
7  add_domicilio (8623, 'Ave. Castor', 6, 20301);
8  add_domicilio (560, 'Calle Fitzpatrick', 5, 30101);
9  add_domicilio (23, 'Calle Sandía', 3, 30201);
10 add_domicilio (789, 'Calle Indio', 5, 30301);
11 add_domicilio (85, 'Calle América', 4, 40101);
12 add_domicilio (236, 'Rincón Largo', 7, 40201);
13 add_domicilio (889, 'Ave. Barco', 2, 40301);
14 add_domicilio (1582, 'Calle Coco', 4, 50101);
15 add_domicilio (432, 'Ave. Candelario', 8, 50201);
16 add_domicilio (976, 'Sector Hielo', 3, 50301);
17 add_domicilio (25, 'Calle Mercado', 6, 60101);
18 add_domicilio (4, 'Calle Arriba', 1, 60201);
19 add_domicilio (57, 'Ave. Montañosa', 4, 60301);
20 add_domicilio (478, 'Calle Mala', 9, 70101);
21 add_domicilio (245, 'Calle Dulce', 3, 70201);
22 add_domicilio (267, 'Calle San Francisco', 2, 70301);
23 add_domicilio (332, 'San Isidro Sector B', 5, 80101);
24 add_domicilio (442, 'Calle F', 3, 80201);
25 add_domicilio (90, 'Ave. Roja', 5, 80301);
26 add_domicilio (1478, 'Calle Centavo', 3, 90101);
27 add_domicilio (898, 'Calle Santa', 2, 90201);
28 add_domicilio (76, 'Calle Y Norte', 6, 90301);
29 add_domicilio (2, 'Calle Negra', 5, 110101);
30 add_domicilio (37, 'Ave. S', 10, 110201);
31 add_domicilio (1009, 'Ave. Dinero', 12, 120101);
32 add_domicilio (4235, 'Calle Bilingüe', 9, 120201);
33 add_domicilio (5344, 'Calle Rubí', 3, 120301);
34 add_domicilio (468, 'Ave. Felicidad', 5, 130101);
35 add_domicilio (89, 'Calle Tropical', 1, 130201);
36 add_domicilio (743, 'Ave. Mora', 6, 130301);
37 END;
38 /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
2 add_sala ('SBT-1H1', '1-H1', 3, 8);
3 add_sala ('SBT-1H2', '1-H2', 2, 6);
4 add_sala ('SCL-2H1', '2-H1', 4, 4);
5 add_sala ('SCL-2H2', '2-H2', 3, 4);
6 add_sala ('SCO-3H1', '3-H1', 4, 5);
7 add_sala ('SCO-3H2', '3-H2', 3, 6);
8 add_sala ('SCH-4H1', '4-H1', 4, 6);
9 add_sala ('SCH-4H2', '4-H2', 2, 4);
10 add_sala ('SDA-5H1', '5-H1', 2, 8);
11 add_sala ('SHE-6H1', '6-H1', 7, 6);
12 add_sala ('SLS-7H1', '7-H1', 4, 4);
13 add_sala ('SPA-8H1', '8-H1', 12, 6);
14 add_sala ('SPA-8H2', '8-H2', 10, 4);
15 add_sala ('SPA-8H3', '8-H3', 11, 4);
16 add_sala ('SPA-8H4', '8-H4', 1, 2);
17 add_sala ('SPA-8H5', '8-H5', 6, 6);
18 add_sala ('SVE-9H1', '9-H1', 4, 6);
19 add_sala ('SVE-9H2', '9-H2', 3, 6);
20 add_sala ('SPO-13H1', '13-H1', 4, 6);
21 add_sala ('SPO-13H2', '13-H2', 2, 4);
22 add_sala ('SNG-12H1', '12-H1', 2, 12);
23 add_sala ('SGY-10H1', '10-H1', 2, 10);
24 add_sala ('UCISPA-8H5', '8-H5', 2, 5);
25 add_sala ('UCISPA-8H1', '8-H1', 2, 2);
26 add_sala ('UCISPA-8H3', '8-H3', 3, 4);
27 END;
28 /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
2 add_persona ('8-956-231', 'Lisa', 'Cortés', 'F', TO_DATE('29/12/1982', 'DD/MM/YYYY'), 90, null, null);
3 add_persona ('1-356-751', 'Melisa', 'Castillo', 'F', TO_DATE('13/10/1987', 'DD/MM/YYYY'), 1567, 'SBT-1H1', '1-H1');
4 add_persona ('2-437-769', 'Hector', 'González', 'M', TO_DATE('10/01/1993', 'DD/MM/YYYY'), 6489, null, null);
5 add_persona ('3-379-451', 'José', 'Magdaleno', 'M', TO_DATE('27/05/1997', 'DD/MM/YYYY'), 560, 'SCO-3H1', '3-H1');
6 add_persona ('4-587-998', 'Victoria', 'Pérez', 'F', TO_DATE('24/03/1953', 'DD/MM/YYYY'), 889, null, null);
7 add_persona ('5-187-7348', 'Ana', 'Ortiz', 'F', TO_DATE('29/12/1972', 'DD/MM/YYYY'), 976, 'SDA-5H1', '5-H1');
8 add_persona ('6-130-643', 'Rodrigo', 'Rodríguez', 'M', TO_DATE('20/10/1964', 'DD/MM/YYYY'), 57, null, null);
9 add_persona ('7-457-32', 'Martín', 'Márquez', 'M', TO_DATE('12/10/1970', 'DD/MM/YYYY'), 245, 'SLS-7H1', '7-H1');
10 add_persona ('8-587-949', 'Marqueia', 'Pinzón', 'F', TO_DATE('2/01/1991', 'DD/MM/YYYY'), 332, null, null);
11 add_persona ('9-516-420', 'Paola', 'Jones', 'F', TO_DATE('23/10/1999', 'DD/MM/YYYY'), 76, 'SVE-9H1', '9-H1');
12 add_persona ('8-950-949', 'Victor', 'Jimenez', 'M', TO_DATE('4/07/1995', 'DD/MM/YYYY'), 332, null, null);
13 add_persona ('8-886-720', 'Román', 'Polanco', 'M', TO_DATE('7/03/1975', 'DD/MM/YYYY'), 90, 'UCISPA-8H3', '8-H3');
14 add_persona ('8-936-600', 'Sofia', 'Moreno', 'F', TO_DATE('15/12/1959', 'DD/MM/YYYY'), 468, 'SPO-13H1', '13-H1');
15 add_persona ('3-714-830', 'Alice', 'Rojas', 'F', TO_DATE('6/12/1997', 'DD/MM/YYYY'), 789, 'SCO-3H2', '3-H2');
16 add_persona ('12-450-92', 'Victoriano', 'Montezuma', 'M', TO_DATE('13/11/1984', 'DD/MM/YYYY'), 1009, null, null);
17 add_persona ('1-438-360', 'Casimiro', 'Bermudez', 'M', TO_DATE('10/11/1990', 'DD/MM/YYYY'), 1726, 'SBT-1H2', '1-H2');
18 add_persona ('2-571-457', 'Carmen', 'Alvarado', 'F', TO_DATE('24/06/1978', 'DD/MM/YYYY'), 8623, 'SCL-2H2', '2-H2');
19 add_persona ('4-614-730', 'Kathy', 'Arosemena', 'F', TO_DATE('27/02/1987', 'DD/MM/YYYY'), 889, 'SCH-4H2', '4-H2');
20 add_persona ('5-411-250', 'Daniela', 'López', 'F', TO_DATE('19/05/1992', 'DD/MM/YYYY'), 432, 'SDA-5H1', '5-H1');
21 add_persona ('6-248-581', 'Jonathan', 'De La Flor', 'M', TO_DATE('31/08/1996', 'DD/MM/YYYY'), 25, 'SHE-6H1', '6-H1');
22 END;
23 /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
2 Add_doctor_hospital (1, '3-H1');
3 Add_doctor_hospital (2, '3-H2');
4 Add_doctor_hospital (3, '7-H1');
5 Add_doctor_hospital (4, '9-H1');
6 Add_doctor_hospital (5, '9-H2');
7 Add_doctor_hospital (6, '12-H1');
8 Add_doctor_hospital (7, '1-H1');
9 Add_doctor_hospital (8, '1-H2');
10 Add_doctor_hospital (9, '6-H1');
11 Add_doctor_hospital (10, '4-H1');
12 Add_doctor_hospital (11, '4-H2');
13 Add_doctor_hospital (12, '8-H1');
14 Add_doctor_hospital (13, '8-H2');
15 Add_doctor_hospital (14, '8-H3');
16 Add_doctor_hospital (15, '8-H4');
17 Add_doctor_hospital (16, '8-H5');
18 Add_doctor_hospital (17, '5-H1');
19 Add_doctor_hospital (18, '2-H1');
20 Add_doctor_hospital (19, '2-H2');
21 Add_doctor_hospital (20, '10-H1');
22 Add_doctor_hospital (21, '13-H1');
23 Add_doctor_hospital (22, '13-H2');
24 END;
25 /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
  2 add_persona_sintoma ('F', 1);
  3 add_persona_sintoma ('DG', 1);
  4 add_persona_sintoma ('DR', 2);
  5 add_persona_sintoma ('F', 2);
  6 add_persona_sintoma ('TS', 3);
  7 add_persona_sintoma ('DC', 4);
  8 add_persona_sintoma ('F', 5);
  9 add_persona_sintoma ('P', 6);
 10 add_persona_sintoma ('E', 7);
 11 add_persona_sintoma ('DG', 7);
 12 add_persona_sintoma ('DR', 8);
 13 add_persona_sintoma ('CO', 9);
 14 add_persona_sintoma ('DM', 10);
 15 add_persona_sintoma ('D', 11);
 16 add_persona_sintoma ('C', 12);
 17 add_persona_sintoma ('E', 12);
 18 add_persona_sintoma ('IH', 13);
 19 add_persona_sintoma ('TS', 14);
 20 add_persona_sintoma ('F', 15);
 21 add_persona_sintoma ('CO', 15);
 22 END;
 23 /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
  2 add_persona_ep ('CAN', 1, to_date('10/3/2019', 'dd/mm/yyyy'));
  3 add_persona_ep ('DIA', 2, to_date('12/5/2015', 'dd/mm/yyyy'));
  4 add_persona_ep ('ECA', 3, to_date('13/4/2017', 'dd/mm/yyyy'));
  5 add_persona_ep ('EPU', 4, to_date('30/1/2018', 'dd/mm/yyyy'));
  6 add_persona_ep ('HIP', 5, to_date('10/8/2019', 'dd/mm/yyyy'));
  7 add_persona_ep ('CAN', 6, to_date('31/10/2018', 'dd/mm/yyyy'));
  8 add_persona_ep ('DIA', 7, to_date('12/3/2016', 'dd/mm/yyyy'));
  9 add_persona_ep ('ECA', 8, to_date('10/5/2017', 'dd/mm/yyyy'));
 10 add_persona_ep ('EPU', 9, to_date('24/6/2019', 'dd/mm/yyyy'));
 11 add_persona_ep ('HIP', 10, to_date('16/9/2017', 'dd/mm/yyyy'));
 12 END;
 13 /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
  2 add_persona_prueba (1, 'H', to_date('23/4/2020', 'dd/mm/yyyy'));
  3 add_persona_prueba (2, 'H', to_date('30/6/2020', 'dd/mm/yyyy'));
  4 add_persona_prueba (3, 'S', to_date('12/5/2020', 'dd/mm/yyyy'));
  5 add_persona_prueba (4, 'H', to_date('7/6/2020', 'dd/mm/yyyy'));
  6 add_persona_prueba (5, 'S', to_date('13/6/2020', 'dd/mm/yyyy'));
  7 add_persona_prueba (6, 'H', to_date('10/5/2020', 'dd/mm/yyyy'));
  8 add_persona_prueba (7, 'S', to_date('8/5/2020', 'dd/mm/yyyy'));
  9 add_persona_prueba (8, 'H', to_date('30/5/2020', 'dd/mm/yyyy'));
 10 add_persona_prueba (9, 'S', to_date('12/4/2020', 'dd/mm/yyyy'));
 11 add_persona_prueba (10, 'H', to_date('25/5/2020', 'dd/mm/yyyy'));
 12 add_persona_prueba (11, 'S', to_date('9/6/2020', 'dd/mm/yyyy'));
 13 add_persona_prueba (12, 'H', to_date('4/7/2020', 'dd/mm/yyyy'));
 14 add_persona_prueba (13, 'H', to_date('23/3/2020', 'dd/mm/yyyy'));
 15 add_persona_prueba (14, 'S', to_date('18/4/2020', 'dd/mm/yyyy'));
 16 add_persona_prueba (15, 'H', to_date('29/6/2020', 'dd/mm/yyyy'));
 17 END;
 18 /

```

PL/SQL procedure successfully completed.


```

SQL> BEGIN
  2 Add_telefono_h ('10-H1', '345-8910');
  3 Add_telefono_h ('12-H1', '268-0943');
  4 Add_telefono_h ('13-H1', '254-8926');
  5 Add_telefono_h ('13-H2', '253-3447');
  6 Add_telefono_h ('1-H1', '774-5689');
  7 Add_telefono_h ('1-H2', '765-8910');
  8 Add_telefono_h ('2-H1', '945-8543');
  9 Add_telefono_h ('2-H2', '998-3219');
 10 Add_telefono_h ('3-H1', '445-5432');
 11 Add_telefono_h ('3-H2', '447-5617');
 12 Add_telefono_h ('4-H1', '785-8532');
 13 Add_telefono_h ('4-H2', '735-3210');
 14 Add_telefono_h ('5-H1', '245-5670');
 15 Add_telefono_h ('6-H1', '995-6789');
 16 Add_telefono_h ('7-H1', '996-7480');
 17 Add_telefono_h ('8-H1', '507-5830');
 18 Add_telefono_h ('8-H2', '305-6300');
 19 Add_telefono_h ('8-H3', '360-7400');
 20 Add_telefono_h ('8-H4', '399-8082');
 21 Add_telefono_h ('8-H5', '503-6600');
 22 Add_telefono_h ('9-H1', '985-4287');
 23 Add_telefono_h ('9-H2', '978-0045');
 24 END;
 25 /

```

PL/SQL procedure successfully completed.

```

SQL> BEGIN
  2 add_tipo_telefono_p (1, 'Residencial');
  3 add_tipo_telefono_p (2, 'Celular');
  4 END;
  5 /

```

PL/SQL procedure successfully completed.

```

SQL> begin
  2 upd_stat;
  3 end;
  4 /

```

PL/SQL procedure successfully completed.

```

SQL> begin
  2 act_age;
  3 end;
  4 /

```

PL/SQL procedure successfully completed.

Creación de Vistas

```
SQL> create view Prov_Dom as select d.num_domic, c.nombre_correg, p.nombre_prov, p.Cod_prov
  2  from Domicilio d join Corregimiento c
  3  on d.Cod_correg=c.Cod_correg join Distrito dis
  4  on dis.Cod_dist=c.Cod_dist join Provincia p
  5  on p.Cod_prov=dis.Cod_prov;
```

View created.

```
SQL> create view Per_Dom as select distinct Cedula_p, apellido_p, nombre_p, Num_domic
  2  from Persona
  3  order by apellido_p;
```

View created.

```
SQL>
SQL> create view Sin as select per.apellido_p, nombre_p as, sim.nombre_sint, per.Cedula_p
  2  from Sintomas sim join Persona_Sintoma persim
  3  on sim.Cod_sint = persim.Cod_sint join Persona per
  4  on per.id_persona = persim.id_persona
  5  where sim.nombre_sint in ('Fiebre');
```

View created.

```
SQL>
SQL> create view Prov_per_dom as select per.apellido_p, nombre_p
  2  from Provincia pro join Distrito dis
  3  on pro.Cod_prov = dis.Cod_prov join Corregimiento cor
  4  on dis.Cod_dist = cor.Cod_dist join Domicilio dom
  5  on cor.Cod_correg = dom.Cod_correg join Persona per
  6  on per.Num_domic = dom.Num_domic join Persona_Sintoma pers
  7  on per.id_persona = pers.id_persona join Persona_Enfermedades_Previas peref
  8  on pers.id_persona = peref.id_persona
  9  where pro.Cod_prov = 8 and pers.Cod_sint not in ('F', 'TS') or peref.Cod_enf = 'DIA';
```

View created.

```
SQL>
SQL> create view Prov_per as select per.apellido_p, nombre_p, pro.nombre_prov
  2  from Provincia pro join Distrito dis
  3  on pro.Cod_prov = dis.Cod_prov join Corregimiento cor
  4  on dis.Cod_dist = cor.Cod_dist join Domicilio dom
  5  on cor.Cod_correg = dom.Cod_correg join Persona per
  6  on per.Num_domic = dom.Num_domic
  7  where pro.Cod_prov between 1 and 5;
```

View created.

Vistas

```
SQL> select * from prov_dom;
```

NUM_DOMIC	NOMBRE_CORREG	NOMBRE_PROV	COD_PROV

1803	Barrio Francés	Bocas del Toro	1
1567	Bastimentos	Bocas del Toro	1
1726	El Silencio	Bocas del Toro	1
6489	Río Grande	Coclé	2
8456	Pocrí	Coclé	2
8623	El Valle	Coclé	2
560	Cristóbal	Colón	3
23	Piña	Colón	3
789	Cacique	Colón	3
85	San Carlos	Chiriquí	4
236	Las Lajas	Chiriquí	4

NUM_DOMIC	NOMBRE_CORREG	NOMBRE_PROV	COD_PROV

889	Puerto Armuelles	Chiriquí	4
1582	La Palma	Darién	5
432	Yaviza	Darién	5
976	Agua Fría	Darién	5
25	Monagrillo	Herrera	6
4	Llano Grande	Herrera	6
57	Chumical	Herrera	6
478	Valle Rico	Los Santos	7
245	Cambutal	Los Santos	7
267	Llano Abajo	Los Santos	7
332	Omar Torrijos	Panamá	8

NUM_DOMIC	NOMBRE_CORREG	NOMBRE_PROV	COD_PROV

442	Don Bosco	Panamá	8
90	San Miguel	Panamá	8
1478	Urracá	Veraguas	9
898	San Antonio	Veraguas	9
76	La Yeguada	Veraguas	9
2	Lajas Blancas	Emberá Wounaan	11
37	Río Sabalo	Emberá Wounaan	11
1009	Cerro Banco	Ngäbe-buglé	12
4235	Man Creek	Ngäbe-buglé	12
5344	Piedra Roja	Ngäbe-buglé	12
468	Vista Alegre	Panamá Oeste	13

NUM_DOMIC	NOMBRE_CORREG	NOMBRE_PROV	COD_PROV

89	El Coco	Panamá Oeste	13
743	Las Uvas	Panamá Oeste	13

```
35 rows selected.
```

```
SQL> select * from per_dom;
```

CEDULA_P	APELLIDO_P	NOMBRE_P	NUM_DOMIC
2-571-457	Alvarado	Carmen	8623
4-614-730	Arosemena	Kathy	889
1-438-360	Bermudez	Casimiro	1726
1-356-751	Castillo	Melisa	1567
8-956-231	Cortés	Lisa	90
6-248-581	De La Flor	Jonathan	25
2-437-769	González	Hector	6489
8-950-949	Jimenez	Victor	332
9-516-420	Jones	Paola	76
5-411-250	López	Daniela	432
3-379-451	Magdaleno	José	560

CEDULA_P	APELLIDO_P	NOMBRE_P	NUM_DOMIC
12-450-92	Montezuma	Victoriano	1009
8-936-600	Moreno	Sofía	468
7-457-32	Márquez	Martín	245
5-187-7348	Ortiz	Ana	976
8-587-949	Pinzón	Marquela	332
8-886-720	Polanco	Román	90
4-587-998	Pérez	Victoria	889
6-130-643	Rodríguez	Rodrigo	57
3-714-830	Rojas	Alice	789

20 rows selected.

```
SQL> select * from sin;
```

APELLIDO_P	NOMBRE_P	NOMBRE_SINT
CEDULA_P		
Cortés	Lisa	Fiebre
8-956-231		
Castillo	Melisa	Fiebre
1-356-751		
Pérez	Victoria	Fiebre
4-587-998		
APELLIDO_P	NOMBRE_P	NOMBRE_SINT
CEDULA_P		
Montezuma	Victoriano	Fiebre
12-450-92		


```
SQL> select * from prov_per_dom;
```

APELLIDO_P	NOMBRE_P
Castillo	Melisa
Castillo	Melisa
Rodríguez	Rodrigo
Rodríguez	Rodrigo
Pinzón	Marquela
Cortés	Lisa

6 rows selected.

```
SQL> select * from prov_per;
```

APELLIDO_P	NOMBRE_P	NOMBRE_PROV
Castillo	Melisa	Bocas del Toro
González	Hector	Coclé
Magdaleno	José	Colón
Pérez	Victoria	Chiriquí
Ortiz	Ana	Darién
Rojas	Alice	Colón
Bermudez	Casimiro	Bocas del Toro
Alvarado	Carmen	Coclé
Arosemena	Kathy	Chiriquí
López	Daniela	Darién

10 rows selected.

Exportación

```
(2)U(sers), or (3)T(ables): (2)U > 2
```

```
Export grants (sφ/no): sφ > sφ
```

```
Export table data (sφ/no): sφ > sφ
```

```
Compress extents (sφ/no): sφ > sφ
```

```
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set  
server uses AL32UTF8 character set (possible charset conversion)
```

```
. exporting pre-schema procedural objects and actions  
. exporting foreign function library names for user EQUIP01  
. exporting PUBLIC type synonyms  
. exporting private type synonyms  
. exporting object type definitions for user EQUIP01
```

```
About to export EQUIP01's objects ...
```

```
. exporting database links  
. exporting sequence numbers  
. exporting cluster definitions
```

```
. about to export EQUIP01's tables via Conventional Path ...
```

. . exporting table	CONTROL_INFECT	20 rows exported
. . exporting table	CONTROL_STAT	1 rows exported
. . exporting table	CORREGIMIENTO	35 rows exported
. . exporting table	DISTRITO	35 rows exported

. . exporting table	DOMICILIO	35 rows exported
. . exporting table	ENFERMEDADES_PREVIAS	5 rows exported
. . exporting table	HOSPITAL	21 rows exported
. . exporting table	PERSONA	20 rows exported
. . exporting table	PERSONA_ENFERMEDADES_PREVIAS	10 rows exported
. . exporting table	PERSONA_PRUEBA	15 rows exported
. . exporting table	PERSONA_SINTOMA	20 rows exported
. . exporting table	PER_STAT	21 rows exported
. . exporting table	PROVINCIA	15 rows exported
. . exporting table	PRUEBA	2 rows exported
. . exporting table	SALA	24 rows exported
. . exporting table	SINTOMAS	12 rows exported
. . exporting table	STATISTIC	1 rows exported
. . exporting table	TELEFONO_H	21 rows exported
. . exporting table	TELEFONO_P	0 rows exported
. . exporting table	TIPO_TELEFONO_P	2 rows exported

```
. exporting synonyms
```

```
. exporting views
```

```
. exporting stored procedures
```

```
. exporting operators
```

```
. exporting referential integrity constraints
```

```
. exporting triggers
```

```
. exporting indextypes
```

```
. exporting bitmap, functional and extensible indexes
```

```
. exporting posttables actions
```

CONCLUSIÓN

La crisis del coronavirus es el mayor desafío que nos ha tocado enfrentar por primera vez después de mucho tiempo, actualmente los países se encuentran en una lucha constante para contrarrestar la propagación de la enfermedad mediante la realización de pruebas, tratamientos, recomendaciones de salud, poniendo en cuarentena a la población entre muchas otras cosas más.

Las enfermedades tienen un gran impacto en la demografía de la población humana, pues las enfermedades son las causantes de muchas muertes en todo el planeta, por ello la modelación se ha vuelto una herramienta muy importante en el estudio de la propagación de las infecciones.

Además de problemas en la salud mundial esta pandemia también ha ocasionado muchos problemas en la industria, economía y muchos otros sectores ya que con la medida de la cuarentena las personas no pueden salir a trabajar y ni mucho menos pagar sus impuestos. Cada país necesita actuar de inmediato para prepararse, responder y recuperarse. Por ello se necesitan de bases de datos para almacenar todo tipo de datos que se necesiten en el presente o que quizás en un futuro se puedan necesitar.

Con la realización de este trabajo nos hemos percatado que tener un buen control de datos, no específicamente esta situación de pandemia actual sino también para cualquier otra problemática, es de mucha ayuda porque nos permite organizarnos mejor en situaciones de estrés y como principal fin de las bases de datos y los sistemas de información, nos ayuda a tomar las mejores decisiones ya sea dentro de una organización o a nivel poblacional de un país o región.