



**UNIVERSIDAD TECNOLÓGICA DE
PANAMÁ**



**FACULTAD DE INGENIERÍA DE SISTEMAS
COMPUTACIONALES**

LICENCIATURA EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN

SISTEMAS DE BASES DE DATOS II

PROYECTO FINAL

INTEGRANTES:

ANDREINA GÓMEZ 8-939-1682

STEFANIE AROSEMENA 8-885-1747

EMANOL GONZALEZ 3-745-1637

JOSÉ QUINTERO 8-952-698

DOCENTE: ING. HENRY J. LEZCANO P.

GRUPO: 1IF121

FECHA: 30 NOVIEMBRE.

AÑO 2020.

1. Resumen

Este documento corresponde al Proyecto Final de Sistemas de Bases de Datos II, de la Licenciatura en Ingeniería de Sistemas de Información. Donde mostraremos el diseño e implementación de un Sistema de Base de Datos, que da solución a la problemática que estamos viviendo en la actualidad que es la Pandemia COVID-19 en Panamá enfocándonos en las medidas de control de pacientes positivos COVID-19 en Hoteles.

El proyecto ha consistido en desarrollar una Base de Datos para el Ministerio de Salud de Panamá (MINSA), cuyas necesidades es obtener un registro de toda la información necesaria y así estos pueden tomar decisiones con respecto a esta problemática basados a datos organizados.

Este documento describe las diferentes fases de desarrollo del proyecto como lo es la planificación, análisis, diseño e implementación. Además, describe la misión, objetivos de la base de datos. Por último, incluye código fuente desarrollado (en forma de sentencias SQL) para la creación de las tablas, los procedimientos, los triggers y funciones junto a capturas de pantalla de código y de testeos que permiten comprobar el correcto funcionamiento.

2. Índice de contenido y de figuras.

1 Resumen	2
2 Índice de contenido	3
3 Introducción	4
4 Primera Parte. Diseño de la Base de Datos	5
4.1 Nombre del Proyecto de Base de Datos	5
4.2 Misión de la Base de Datos	5
4.3 Objetivo de la Misión de la Base de Datos	5
4.4 Definición del Proyecto	6
4.4.1 Ámbito	6
4.4.2 Alcance	6
4.4.3 Limites	6
4.5 Análisis de requerimientos	7
4.6 Modelado de la Base de Datos	10
4.6.1 Modelo Conceptual E/R	10
4.6.2 Modelo Lógico Relación E/R Normalizado	10
4.6.3 Modelo Físico según sistema de gestión (Oracle)	12
5 Implementación de Base de Datos	19
6 Pruebas	39
7 Conclusiones	82
8 Referencias Bibliográficas	84

3. Introducción.

Ante el vertiginoso crecimiento de nuevos casos de COVID-19 en el país de Panamá, el Ministerio de Salud (MINSAL) han realizado esfuerzos para tratar de contrarrestar los contagios, utilizando protocolos como el aislamiento en hoteles. Para esto, han buscado diferentes hoteles en los cuales se albergarán pacientes positivos COVID-19. La estrategia principal de mantener habilitada las instalaciones de hoteles, permite que el personal médico atienda a los afectados de manera expedita, y al mismo tiempo se evita que propaguen el virus a sus familiares.

Es importante llevar el control detallado de todas las personas afectadas que ingresan a estas instalaciones, así como también observar la evolución de estos, llevar el registro de las pruebas COVID-19 y la fechas en que fueron realizadas.

Cada vez que una persona es detectada como positiva, la misma es entrevistada y debe proporcionar al MINSAL, los datos de las personas que han estado en contacto con ella para de esta forma darle seguimiento y poner estos contactos bajo cuarentena previsorio. Se le da seguimiento.

A cada hotel se le tiene asignado un personal de salud (médicos y enfermeras) quienes tienen bajo su responsabilidad la atención y evolución de todos los afectados.

4. Primera Parte. Diseño de la Base de Datos.

4.1. Nombre del Proyecto de Base de Datos

El nombre de este proyecto es Base de Datos Control de Pacientes COVID-19 positivos en Hoteles, esto provee medidas de control para enfrentar esta pandemia al Ministerio de Salud de Panamá (MINSA) este es un ministerio de la República de Panamá que forma parte del Órgano Ejecutivo. Esta institución se encarga de la salud y el bienestar de la población panameña y de la situación sanitaria en general del país. El ministerio de Salud se creó el 15 de enero de 1969 por medio del Decreto de Gabinete N. °1.

4.2. Misión de la Base de Datos

Misión:

Modernizar la red de atención de salud a nivel nacional, que instruye el establecimiento e implementación de planes y proyectos para el desarrollo de nuevas estrategias de Salud para contrarrestar el contagio del COVID-19, mediante la recolección de datos para obtener una vista general de contagios y estado de la situación.

4.3. Objetivo de la Misión de la Base de Datos

El sistema de base de datos propuesto brinda a esta institución más que facilitar el acceso a las informaciones del paciente, la tecnología de la información que trabaja para orientar las decisiones inherentes al cuidado de la salud. Además de ello, promueve mejorías en la calidad del cuidado de la salud, en la productividad de los profesionales de la salud, facilita el acceso a los servicios disponibles y reduce costos administrativos asociados a la prestación de servicios, permite almacenar gran número de información de una forma organizada, realizar búsquedas de forma rápida y simple desde un ordenador, haciendo más eficaz el manejo de información bajando la tensión que estamos pasando por la pandemia.

4.4. Definición del Proyecto

4.4.1.Ámbito

Este sistema de base de datos registrará a los pacientes positivos por el nuevo COVID-19, reunirá información importante de este individuo para así llevar control de entrada de pacientes a los diferentes hoteles del país que están siendo utilizados para albergar personas afectadas. Además, este sistema también tendrá una lista de información de personas que han estado en contacto con el paciente afectado para así controlar y evitar la propagación del virus.

A la vez se almacena información de los hoteles que están siendo utilizado, junto a esto la información del personal de salud asignado a estas instalaciones.

Todo esto, facilitará al MINSA realizar búsquedas de forma rápida y simple de toda la información correspondiente.

4.4.2.Alcance

Vigilancia Epidemiológica, incluyendo el Control de los Contactos de los Casos Confirmados, el Control personas que estuvieron contactos con el paciente.

Capacidad de respuesta integral del sistema de salud para la atención de los pacientes como lo monitores.

Capacidad de respuesta del diagnóstico de laboratorio.

4.4.3.Limites

Limitantes en este proyecto podemos mencionar:

- La falta de equipos tecnológicos en el Ministerio de Salud.
- Mala organización institucional.
- Falta de información por parte del paciente positivo.
- Necesidad de contar con mano de obra cada vez más calificada y actualizada en forma permanente.

- La complejidad y la infraestructura necesaria implica que se necesitará una mayor mano de obra.
- Se debe trabajar en la seguridad de la infraestructura, así como cada uno de los sistemas.

4.5. Análisis de requerimientos

Técnicas utilizadas:

- Observación de la operación de la institución, Ministerio de Salud.
- Investigación.
- Cuestionarios.

Identificación de datos necesarios para el diseño de la base de datos.	
Pacientes positivos COVID19	<ul style="list-style-type: none"> • Cédula. • Nombre. • Apellido. • Edad. • Teléfono. • Dirección. • Provincia. • Corregimiento. • Fecha del ingreso al hotel.
Historial médico del paciente positivo COVID-19	<ul style="list-style-type: none"> • Fecha de la prueba COVID-19. • Resultado prueba COVID-19. • Conocer si presenta alguna enfermedad.
Personas que han estado en contacto con el paciente positivo COVID-19	<ul style="list-style-type: none"> • Nombre. • Teléfono. • Dirección.
Hotel	<ul style="list-style-type: none"> • Nombre del Hotel. • Corregimiento.

	<ul style="list-style-type: none"> • Provincia. • Cantidad de trabajadores del hotel (Para determinar cuántas personas pueden llegar a estar expuestas y mantener orden de la capacidad de trabajadores).
Personal de la Salud	<ul style="list-style-type: none"> • Registro. • Nombre. • Cédula. • Ocupación. • Teléfono.

Esta base de datos es diseñada para llevar el control de los registros de pacientes COVID-19 positivo que se encuentran hospedados en hoteles, por la cantidad grande de información que se maneja requiere la información precisa y de forma sencilla y rápida de obtener, verificar, agregar.

El MINSA en la gestión ha ordenado hospedar pacientes positivos COVID-19 a diferentes hoteles en Panamá. Y necesita llevar un control de pacientes, hoteles y personal de salud asignados a estos hoteles.

Al registrar un paciente positivo por COVID-19 se procede a reunir la siguiente información (cédula, nombre, apellido, edad, teléfonos, dirección, provincia, corregimiento, fecha de ingreso a dicho hospital). También se necesita conocer el historial médico de estos pacientes y si sufren de otra enfermedad. En la figura 1 se muestra un ejemplo de formulario de registro de casos COVID-19 detallando las posibles enfermedades. Además, es necesario preguntar a los pacientes afectados los contactos que ha tenido contacto y así obtener una lista de control y evitar la propagación del virus.

OTRAS ENFERMEDADES (presentes antes del ingreso) (S. d. = se desconoce)			
Cardiopatía crónica (no hipertensión)	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Hipertensión	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Neumopatía crónica	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Asma	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Insuficiencia renal crónica	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Hepatopatía crónica	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Trastorno neurológico crónico	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Diabetes	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Fumador (actualmente)	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Tuberculosis	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Asplenia	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Neoplasia maligna	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
Otros	<input type="checkbox"/> Sí	<input type="checkbox"/> No	<input type="checkbox"/> S. d.
En caso afirmativo, especifique: _____			
Infección por el VIH	<input type="checkbox"/> Sí (toma TAR)	<input type="checkbox"/> Sí (no toma TAR)	<input type="checkbox"/> No <input type="checkbox"/> S. d.

Figura 1. Formulario de registro de casos de COVID-19. Versión rápida. 8 de abril de 2020.

Se conoce que cadenas hoteleras de diferentes partes del país se han puesto a disposición del Ministerio de Salud (MINSA), para la atención de pacientes afectados con el COVID-19.

En la figura 2 se muestra el ejemplo de formulario para el uso de hoteles durante la pandemia de la COVID-19.

Información general del hotel

Nombre del hotel:
Dirección:
Ciudad:
País:
Número de empleados del hotel que continuarán prestando servicios:

Para determinar cuántas personas pueden llegar a estar expuestas.

Figura 2. Consideraciones para el uso de hoteles durante la pandemia de la COVID-19 por la Organización Panamericana de la Salud (OPS).

El personal de salud asignado a estas instalaciones lleva el control y atienden a los pacientes afectados. Hay que destacar que el personal de salud solo es enviado a un hotel determinado.

Y estos atienden a todos los afectados que se encuentran en los hoteles y son los encargados de atender los pacientes y realizar visitas y anotar la fecha y los detalles de las visitas, para observar la evolución del afectado.

4.6. Modelado de la Base de Datos

4.6.1. Modelo Conceptual E/R



4.6.2. Modelo Lógico Relación E/R Normalizado

Tipo_profesion		
PK	cod_profesion	N
	profesion	N

enfermedades		
PK	cod_enfer	N
	nombre_enfermedad	N

provincia		
PK	cod_provincia	N
	nombre_provincia	N

historialmedico		
PK	cod_hm	N
	fecha_pruebacovid	N
	resultado	N

hotel		
PK	cod_hotel	N
	nombre_hotel	N
	corregimiento	N
	provincia	N
FK	provin_fk	N
	cod_provincia	N
	cant_trabajador	N

historial_enfermedades		
PK	cod_hm	N
FK	codigo_historial_fk	N
	cod_hm	N
	cod_enfer	N
FK	cod_enfer_fk	N
	cod_enfer	N

contacto_de_afectado		
PK	Cod_contact	N
	Nombre	N
	telefono	N
	direccion	N

paciente_contacto		
PK	codig_paciente	N
FK	codig_fk	N
	Cod_paciente	N
	cod_contact	N
FK	cod_contacto_fk	N
	cod_contact	N

personal_de_salud		
PK	registro	N
	nombre	N
	Cedula	N
	telefono	N
	ocupacion	N
FK	ocup_fk	N
	cod_profesion	N
	cod_hotel	N
FK	personal_cod_hotel_fk	N
	cod_hotel	N

paciente		
PK	Cod_paciente	N
	Cedula	N
	nombre	N
	apellido	N
	edad	N
	fecha_nac	N
	sexo	N
	telefono	N
	direccion	N
	corregimiento	N
	fecha_ingreso	N
	provincia	N
FK	prov_fk	N
	cod_provincia	N
	cod_hm	N
FK	cod_hm_fk	N
	cod_hm	N
	cod_hotel	N
	cod_hotel_fk	N
	cod_hotel	N

4.6.3. Modelo Físico según sistema de gestión (Oracle)

A continuación, nuestro modelo físico de base de datos, consta de once (11) tablas.

Tabla Tipo de Profesión. (Contiene las diferentes profesiones del personal de la salud).	Create table Tipo_profesion (cod_profesion varchar2(5) primary key not null, profesion varchar2(100));
	<pre>SQL> Create table Tipo_profesion (2 cod_profesion varchar2(5) primary key not null, 3 profesion varchar2(100)); Table created.</pre>

Tabla enfermedades. (Contiene las referencias de enfermedades).	Create table enfermedades(cod_enfer varchar2(10) primary key, nombre_enfermedad varchar2(70));
	<pre>SQL> Create table enfermedades(2 cod_enfer varchar2(10) primary key, 3 nombre_enfermedad varchar2(70)); Table created.</pre>

Tabla Provincias. (Contiene las provincias de Panamá).	Create table provincia(cod_provincia varchar2(5) primary key not null, nombre_provincia varchar2(30));
	<pre>SQL> Create table provincia(2 cod_provincia varchar2(5) primary key not null, 3 nombre_provincia varchar2(30)); Table created.</pre>

<p>Tabla Historial Médico.</p> <p>(Contiene el historial médico de los pacientes).</p>	<p>Create table historialmedico(cod_hm varchar2(12) primary key not null, fecha_pruebacovid date, resultado varchar2(10));</p>
	<pre>SQL> Create table historialmedico(2 cod_hm varchar2(12) primary key not null, 3 fecha_pruebacovid date, 4 resultado varchar2(10)); Table created.</pre>

<p>Tabla Hotel.</p> <p>(Contiene los hoteles que están albergando pacientes positivos COVID-19).</p>	<p>Create table hotel(cod_hotel number primary key not null, nombre_hotel varchar2(40), corregimiento varchar2 (40), provincia varchar2(20), constraint provin_fk foreign key (provincia) references provincia (cod_provincia), cant_trabajador number);</p>
	<pre>SQL> Create table hotel(2 cod_hotel number primary key not null, 3 nombre_hotel varchar2(40), 4 corregimiento varchar2 (40), 5 provincia varchar2(20), 6 constraint provin_fk foreign key (provincia) references provincia (cod_provincia), 7 cant_trabajador number); Table created.</pre>

<p>Tabla Paciente. (Contiene la información del paciente COVID-19).</p>	<pre>Create table paciente(Cod_paciente varchar2(20) primary key not null, Cedula varchar2(20), nombre varchar2(40), apellido varchar2(40), edad number, fecha_nac date, sexo varchar2(10), telefono varchar2(12), direccion varchar2(60), corregimiento varchar2(40), fecha_ingresohotel date, provincia varchar2(20), constraint prov_fk foreign key (provincia) references provincia (cod_provincia), cod_hm varchar2(12), constraint cod_hm_fk foreign key (cod_hm) references historialmedico(cod_hm), cod_hotel number, constraint cod_hotel_fk foreign key (cod_hotel) references hotel(cod_hotel));</pre>
--	--

	<pre> SQL> Create table paciente(2 Cod_paciente varchar2(20) primary key not null, 3 Cedula varchar2(20), 4 nombre varchar2(40), 5 apellido varchar2(40), 6 edad number, 7 fecha_nac date, 8 sexo varchar2(10), 9 telefono varchar2(12), 10 direccion varchar2(60), 11 corregimiento varchar2(40), 12 fecha_ingresohotel date, 13 provincia varchar2(20), 14 constraint prov_fk foreign key (provincia) references provincia (cod_provincia), 15 cod_hm varchar2(12), 16 constraint cod_hm_fk foreign key (cod_hm) r ferences historialmedico(cod_hm), 17 cod_hotel number, 18 constraint cod_hotel_fk foreign key (cod_ho tel) references hotel(cod_hotel)); Table created. </pre>
--	--

<p>Tabla Personal de Salud.</p> <p>(Contiene información sobre el personal de la salud que es asignado a los hoteles).</p>	<pre> Create table personal_de_salud(registro varchar2(10) primary key not null, nombre varchar2(50), cedula varchar2(30), telefono varchar2(12), ocupacion varchar2(60), constraint ocup_fk foreign key (ocupacion) references Tipo_profesion(cod_profesion), cod_hotel number, constraint personal_cod_hotel_fk foreign key (cod_hotel) references hotel(cod_hotel)); </pre>
---	---

	<pre>SQL> Create table personal_de_salud(2 registro varchar2(10) primary key not null, 3 nombre varchar2(50), 4 cedula varchar2(30), 5 telefono varchar2(12), 6 ocupacion varchar2(60), 7 constraint ocup_fk foreign key (ocupacion) references Tipo_profesion(cod_profesion), 8 cod_hotel number, 9 constraint personal_cod_hotel_fk foreign key (cod_hotel) references hotel(cod_hotel)); Table created.</pre>
--	--

<p>Tabla Historial Personal Salud. (Contiene información que introduce el personal de la salud cuando realiza visitas a los pacientes).</p>	<pre>Create table historial_personalsalud(cod_hm varchar2(12), constraint historial_medlsaludfk foreign key (cod_hm) references historialmedico(cod_hm), regist varchar2(10), constraint historial_personalsaludfk foreign key (regist) references personal_de_salud(registro), fecha date, detalle varchar2(100));</pre>
	<pre>SQL> Create table historial_personalsalud(2 cod_hm varchar2(12), 3 constraint historial_medlsaludfk foreign key (cod_hm) references historialmedico(cod_hm), 4 regist varchar2(10), 5 constraint historial_personalsaludfk foreign key (regist) references personal_de_salud(registro), 6 fecha date, 7 detalle varchar2(100)); Table created.</pre>

<p>Tabla Historial enfermedades.</p> <p>(Contiene la relacion de información del código de historial médico de un paciente y el código de enfermedad).</p>	<p>Create table historial_enfermedades (cod_hm varchar2(12), constraint codigo_historial_fk foreign key (cod_hm) references historialmedico(cod_hm), cod_enfer varchar2(10), constraint cod_enfer_fk foreign key (cod_enfer) references enfermedades(cod_enfer));</p> <pre>SQL> Create table historial_enfermedades (2 cod_hm varchar2(12), 3 constraint codigo_historial_fk foreign key (cod_hm) references historialmedico(cod_hm), 4 cod_enfer varchar2(10), 5 constraint cod_enfer_fk foreign key (cod_enfer) references enfermedades(cod_enfer)); Table created.</pre>
---	---

<p>Tabla contacto de Afectado.</p> <p>(Contiene información del contacto del paciente positivo COVID-19).</p>	<p>Create table contacto_de_afectado(Cod_contact varchar2(12) primary key not null, Nombre varchar2(40), telefono varchar2(12), direccion varchar2(60));</p> <pre>SQL> Create table contacto_de_afectado(2 Cod_contact varchar2(12) primary key not null, 3 Nombre varchar2(40), 4 telefono varchar2(12), 5 direccion varchar2(60)); Table created.</pre>
--	--

<p>Tabla Paciente Contacto. (Contiene la relación del paciente positivo COVID-19 y contactos de este).</p>	<pre>Create table paciente_contacto(codig_paciente varchar2(20), constraint codig_fk foreign key (codig_paciente) references paciente(Cod_paciente), cod_contact varchar2(12), constraint cod_contacto_fk foreign key (cod_contact) references contacto_de_afectado(cod_contact));</pre> <pre>SQL> Create table paciente_contacto(2 codig_paciente varchar2(20), 3 constraint codig_fk foreign key (codig_paciente) references paciente(Cod_paciente), 4 cod_contact varchar2(12), 5 constraint cod_contacto_fk foreign key (cod_cont act) references contacto_de_afectado(cod_contact));</pre> <p>Table created.</p>
---	--

5. Implementación de Base de Datos

Procedimientos

1 Procedimiento. Para registrar historiales médicos de los pacientes que serán ingresados en Hoteles por ser positivos en Covid-19.

```
CREATE SEQUENCE N_historialmed
START with 101
INCREMENT BY 1;
CREATE OR REPLACE PROCEDURE REG_HISTMED(
P_FECHA_P    historialmedico.fecha_pruebacovid%type,
P_RESULT      historialmedico.resultado%type,
P_INFO out varchar2)IS
BEGIN

INSERT
INTO historialmedico VALUES('a'||TO_CHAR(N_historialmed.nextval),P_FECHA
_P,P_RESULT);
    P_INFO := 'HISTORIAL REGISTRADO';

EXCEPTION
    WHEN OTHERS THEN
        P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';
COMMIT;
END;
/
```

```

SQL> CREATE SEQUENCE N_historialmed
  2  START  with 101
  3  INCREMENT BY 1;

Sequence created.

SQL> CREATE OR REPLACE PROCEDURE REG_HISTMED(
  2  P_FECHA_P      historialmedico.fecha_pruebacovid%type,
  3  P_RESULT      historialmedico.resultado%type,
  4  P_INFO      out  varchar2)IS
  5  BEGIN
  6
  7  INSERT INTO historialmedico VALUES('a' || TO_CHAR(N_historialmed.nextval)
,P_FECHA_P,P_RESULT);
  8      P_INFO := 'HISTORIAL REGISTRADO';
  9
 10  EXCEPTION
 11      WHEN OTHERS THEN
 12          P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';
 13  COMMIT;
 14  END;
 15  /

Procedure created.

```

2 Procedimiento. Para realizar el registro de Hoteles.

```

CREATE SEQUENCE N_hotel
START  with 1001
INCREMENT BY 1;

CREATE OR REPLACE PROCEDURE REG_HOTEL (
P_NAME_HOTEL  hotel.nombre_hotel%type,
P_CORREG      hotel.corregimiento%type,
P_PROVIN      hotel.provincia%type,
P_CANT_WORKER hotel.cant_trabajador%type,
P_INFO  OUT  VARCHAR2)

```

```

IS
BEGIN

INSERT INTO hotel
VALUES(N_hotel.nextval,P_NAME_HOTEL,P_CORREG,P_PROVIN,P_CANT_
WORKER);
P_INFO := '!!!REGISTRO EXITOSO!!!';

EXCEPTION
    WHEN OTHERS THEN
        P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';

END REG_HOTEL;
/

```

```

SQL> CREATE SEQUENCE N_hotel
  2  START  with 1001
  3  INCREMENT BY 1;

Sequence created.

SQL> CREATE OR REPLACE PROCEDURE REG_HOTEL (
  2  P_NAME_HOTEL  hotel.nombre_hotel%type,
  3  P_CORREG      hotel.corregimiento%type,
  4  P_PROVIN      hotel.provincia%type,
  5  P_CANT_WORKER hotel.cant_trabajador%type,
  6  P_INFO  OUT  VARCHAR2)
  7  IS
  8  BEGIN
  9
 10  INSERT INTO hotel VALUES(N_hotel.nextval,P_NAME_HOTEL,P_CORREG,P_PROVIN
,P_CANT_WORKER);
 11  P_INFO := '!!!REGISTRO EXITOSO!!!';
 12
 13  EXCEPTION
 14      WHEN OTHERS THEN
 15          P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';
 16
 17  END REG_HOTEL;
 18  /

Procedure created.

```

Función. Para calcular la edad del paciente.

```
CREATE OR REPLACE FUNCTION sacarEdad (p_fecfin IN DATE)
RETURN NUMBER
IS
v_anos NUMBER := 0;
BEGIN
v_anos := FLOOR(MONTHS_BETWEEN( TO_DATE(sysdate) ,p_fecfin ) / 12);
RETURN v_anos;
EXCEPTION
WHEN OTHERS THEN
RETURN 0;
END sacarEdad;
/
```

```
SQL> CREATE OR REPLACE FUNCTION sacarEdad (p_fecfin IN DATE)
2
3 RETURN NUMBER
4
5 IS
6
7 v_anos NUMBER := 0;
8
9 BEGIN
10
11 v_anos := FLOOR(MONTHS_BETWEEN( TO_DATE(sysdate) ,p_fecfin ) / 12);
12
13 RETURN v_anos;
14
15 EXCEPTION
16
17 WHEN OTHERS THEN
18
19 RETURN 0;
20
21 END sacarEdad;
22
23 /

Function created.
```

3 Procedimiento. Para realizar el registro de pacientes.

```
CREATE SEQUENCE N_paciente
INCREMENT BY 1;
CREATE OR REPLACE PROCEDURE REG_PACIENTE(
P_CEDULA      paciente.Cedula%TYPE,
P_NAME        paciente.nombre%TYPE,
P_LASTNAME    paciente.apellido%TYPE,
P_FECHNAC     paciente.fecha_nac%TYPE,
P_SEXO        paciente.sexo%TYPE,
P_TELEFONO    paciente.telefono%TYPE,
P_DIRECCION   paciente.direccion%TYPE,
P_CORREG      paciente.corregimiento%TYPE,
P_PROVIN      paciente.provincia%TYPE,
P_HISTMEDI    paciente.cod_hm%TYPE,
P_HOTEL       paciente.cod_hotel%TYPE,
P_INFO out varchar2)AS

v_edad number:=0;
ex_encontrado EXCEPTION;
v_ORDEN number := 1;

cursor c_paciente is
  SELECT *
  FROM paciente
  WHERE Cedula = P_CEDULA;

BEGIN
v_edad := sacarEdad(P_FECHNAC);

FOR hotel in c_paciente LOOP
```

```

        IF c_paciente%ROWCOUNT > 0 THEN
            v_ORDEN := 0;
            RAISE ex_encontrado;
        END IF;
    END LOOP;

    IF v_ORDEN = 1 THEN
        INSERT
        INTO paciente VALUES('PA'||TO_CHAR(N_paciente.nextval),P_CEDULA
        ,P_NAME,P_LASTNAME,v_edad,P_FECHNAC,P_SEXO,P_TELEFONO,
        P_DIRECCION,P_CORREG,SYSDATE,P_PROVIN,P_HISTMEDI,P_HOT
        EL);

        P_INFO := 'PACIENTE REGISTRADO: '||P_NAME ||', '||P_LASTNAME||',
        '||P_CEDULA;
    END IF;

    EXCEPTION
        WHEN ex_encontrado THEN
            P_INFO := '!!!REGISTRO EXISTENTE!!!';
        WHEN OTHERS THEN
            P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';
    COMMIT;
    END;
/

```



```

SQL> CREATE SEQUENCE N_paciente
  2 INCREMENT BY 1;

Sequence created.

SQL> CREATE OR REPLACE PROCEDURE REG_PACIENTE(
  2 P_CEDULA      paciente.Cedula%TYPE,
  3 P_NAME        paciente.nombre%TYPE,
  4 P_LASTNAME    paciente.apellido%TYPE,
  5 P_FECHNAC     paciente.fecha_nac%TYPE,
  6 P_SEXO        paciente.sexo%TYPE,
  7 P_TELEFONO    paciente.telefono%TYPE,
  8 P_DIRECCION   paciente.direccion%TYPE,
  9 P_CORREG      paciente.corregimiento%TYPE,
10 P_PROVIN      paciente.provincia%TYPE,
11 P_HISTMEDI     paciente.cod_hm%TYPE,
12 P_HOTEL       paciente.cod_hotel%TYPE,
13 P_INFO out varchar2)AS
14
15 v_edad number:=0;
16 ex_encontrado EXCEPTION;
17 v_ORDEN number := 1;
18
19     cursor c_paciente is
20     SELECT *
21     FROM paciente
22     WHERE Cedula = P_CEDULA;
23
24 BEGIN
25 v_edad := sacarEdad(P_FECHNAC);
26
27 FOR hotel in c_paciente LOOP
28 IF c_paciente%ROWCOUNT > 0 THEN
29 v_ORDEN := 0;
30 RAISE ex_encontrado;
31 END IF;
32 END LOOP;
33
34 IF v_ORDEN = 1 THEN
35 INSERT INTO paciente VALUES('PA'||TO_CHAR(N_paciente.nextval),P_CEDULA,
P_NAME,P_LASTNAME,v_edad,P_FECHNAC,P_SEXO,P_TELEFONO,P_DIRECCION,P_CORREG,SY
SDATE,P_PROVIN,P_HISTMEDI,P_HOTEL);
36
37 P_INFO := 'PACIENTE REGISTRADO: '||P_NAME ||', '||P_LASTNAME||', '||P_C
EDULA;
38 END IF;
39
40 EXCEPTION
41     WHEN ex_encontrado THEN
42         P_INFO := '!!!REGISTRO EXISTENTE!!!';
43     WHEN OTHERS THEN
44         P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';
45 COMMIT;
46 END;
47 /

Procedure created.

```

4 Procedimiento. Para crear registro de las diferentes ocupaciones del personal de salud encargado de atender a los pacientes dentro de los hoteles.

```
CREATE OR REPLACE PROCEDURE CREAR_OCUP (  
  P_COD_PROFESION  Tipo_profesion.cod_profesion%TYPE,  
  P_DESCRIPCION    Tipo_profesion.profesion%TYPE,  
  P_INFO out varchar2)  
AS  
  ex_encontrado EXCEPTION;  
  v_ORDEN number := 1;  
  cursor c_profesion is  
    SELECT *  
    FROM Tipo_profesion  
    WHERE cod_profesion = P_COD_PROFESION;  
BEGIN  
  FOR hotel in c_profesion LOOP  
    IF c_profesion %ROWCOUNT > 0 THEN  
      v_ORDEN := 0;  
      RAISE ex_encontrado;  
    END IF;  
  END LOOP;  
  IF v_ORDEN = 1 THEN  
    INSERT INTO Tipo_profesion VALUES(P_COD_PROFESION,  
    P_DESCRIPCION);  
    P_INFO := 'PROFECION REGISTRADA';  
  END IF;  
EXCEPTION  
  WHEN ex_encontrado THEN  
    P_INFO := '!!!REGISTRO EXISTENTE!!!';  
  WHEN OTHERS THEN  
    P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';
```

```
COMMIT;  
END CREAR_OCUP;  
/  

```

```
SQL> CREATE OR REPLACE PROCEDURE CREAR_OCUP (  
2  P_COD_PROFESION      Tipo_profesion.cod_profesion%TYPE,  
3  P_DESCRIPCION        Tipo_profesion.profesion%TYPE,  
4  P_INFO out varchar2)  
5  AS  
6  ex_encontrado EXCEPTION;  
7  v_ORDEN number := 1;  
8  cursor c_profesion is  
9  SELECT *  
10 FROM Tipo_profesion  
11 WHERE cod_profesion = P_COD_PROFESION;  
12 BEGIN  
13 FOR hotel in c_profesion LOOP  
14 IF c_profesion %ROWCOUNT > 0 THEN  
15 v_ORDEN := 0;  
16 RAISE ex_encontrado;  
17 END IF;  
18 END LOOP;  
19 IF v_ORDEN = 1 THEN  
20 INSERT INTO Tipo_profesion VALUES(P_COD_PROFESION, P_DESCRIPCION);  
21 P_INFO := 'PROFECION REGISTRADA';  
22 END IF;  
23 EXCEPTION  
24 WHEN ex_encontrado THEN  
25 P_INFO := '!!!REGISTRO EXISTENTE!!!';  
26 WHEN OTHERS THEN  
27 P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';  
28 COMMIT;  
29 END CREAR_OCUP;  
30 /
```

Procedure created.

5 Procedimiento. Para crear registro de enfermedades.

```
CREATE OR REPLACE PROCEDURE CREAR_ENFERMEDAD (  
  P_COD_ENFERMEDAD  enfermedades.cod_enfer%TYPE,  
  P_ENFERMEDAD      enfermedades.nombre_enfermedad%TYPE,  
  P_INFO out varchar2)  
AS  
ex_encontrado EXCEPTION;  
v_ORDEN number := 1;  
  cursor c_enferme is  
    SELECT *  
    FROM enfermedades  
    WHERE cod_enfer = P_COD_ENFERMEDAD;  
BEGIN  
  FOR hotel in c_enferme LOOP  
    IF c_enferme  
      %ROWCOUNT > 0 THEN  
      v_ORDEN := 0;  
      RAISE ex_encontrado;  
    END IF;  
  END LOOP;  
  IF v_ORDEN = 1 THEN  
    INSERT  
    INTO enfermedades VALUES(P_COD_ENFERMEDAD,P_ENFERMEDAD);  
    P_INFO := 'ENFERMEDAD REGISTRADA';  
  END IF;  
EXCEPTION  
  WHEN ex_encontrado THEN  
    P_INFO := '!!!REGISTRO EXISTENTE!!!';  
  WHEN OTHERS THEN  
    P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';
```

```
COMMIT;  
END CREAR_ENFERMEDAD;  
/
```

```
SQL> CREATE OR REPLACE PROCEDURE CREAR_ENFERMEDAD (  
2  P_COD_ENFERMEDAD    enfermedades.cod_enfer%TYPE,  
3  P_ENFERMEDAD        enfermedades.nombre_enfermedad%TYPE,  
4  P_INFO out varchar2)  
5  AS  
6  ex_encontrado EXCEPTION;  
7  v_ORDEN number := 1;  
8  cursor c_enferme is  
9      SELECT *  
10     FROM enfermedades  
11     WHERE cod_enfer = P_COD_ENFERMEDAD;  
12 BEGIN  
13 FOR hotel in c_enferme LOOP  
14 IF c_enferme  
15   %ROWCOUNT > 0 THEN  
16 v_ORDEN := 0;  
17 RAISE ex_encontrado;  
18 END IF;  
19 END LOOP;  
20 IF v_ORDEN = 1 THEN  
21 INSERT INTO enfermedades VALUES(P_COD_ENFERMEDAD,P_ENFERMEDAD);  
22   P_INFO := 'ENFERMEDAD REGISTRADA';  
23 END IF;  
24 EXCEPTION  
25   WHEN ex_encontrado THEN  
26     P_INFO := '!!!REGISTRO EXISTENTE!!!';  
27   WHEN OTHERS THEN  
28     P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';  
29 COMMIT;  
30 END CREAR_ENFERMEDAD;  
31 /
```

Procedure created.

6 Procedimiento. Para registrar el enlace si alguno de los pacientes presentan alguna enfermedad quede registrada en el historial médico de estos.

```
CREATE OR REPLACE PROCEDURE EN_HISTORIAL (  
P_HISTMED      historial_enfermedades.cod_hm%type,  
P_ENFERMEDAD   historial_enfermedades.cod_enfer%type,  
P_INFO out varchar2)  
AS  
BEGIN  
  INSERT  
  INTO historial_enfermedades VALUES(P_HISTMED,P_ENFERMEDAD);  
  P_INFO := 'ENLACE MEDICO REGISTRADO';  
EXCEPTION  
  WHEN DUP_VAL_ON_INDEX THEN  
    P_INFO:= '!!!ENLACE YA EXISTENTE!!!';  
  WHEN OTHERS THEN  
    P_INFO:= 'ERROR... ENLACE NO REGISTRADA';  
COMMIT;  
END EN_HISTORIAL;  
/
```

```
SQL> CREATE OR REPLACE PROCEDURE EN_HISTORIAL (  
2  P_HISTMED      historial_enfermedades.cod_hm%type,  
3  P_ENFERMEDAD   historial_enfermedades.cod_enfer%type,  
4  P_INFO out  varchar2)  
5  AS  
6  BEGIN  
7    INSERT INTO historial_enfermedades VALUES(P_HISTMED,P_ENFERMEDAD);  
8    P_INFO := 'ENLACE MEDICO REGISTRADO';  
9  EXCEPTION  
10 WHEN DUP_VAL_ON_INDEX THEN  
11 P_INFO:= '!!!ENLACE YA EXISTENTE!!!';  
12 WHEN OTHERS THEN  
13 P_INFO:= 'ERROR... ENLACE NO REGISTRADA';  
14 COMMIT;  
15 END EN_HISTORIAL;  
16 /  
  
Procedure created.
```

7 Procedimiento. Para registrar personas que han estado en contacto del paciente positivo COVID-19. Es una medida preventiva.

```
CREATE SEQUENCE N_perconta

INCREMENT BY 1;

CREATE OR REPLACE PROCEDURE REG_CONTACINFECT (

P_NAME          contacto_de_afectado.Nombre%TYPE,

P_TELEFONO       contacto_de_afectado.telefono%TYPE,

P_DIRECCION      contacto_de_afectado.direccion%TYPE,

P_INFO out varchar2)

AS

BEGIN

INSERT INTO contacto_de_afectado

VALUES('PI'||TO_CHAR(N_perconta.nextval),P_NAME,P_TELEFONO,P_DIRECCION);

P_INFO := 'ENLACE MEDICO REGISTRADO';

EXCEPTION

    WHEN DUP_VAL_ON_INDEX THEN

        P_INFO:= '!!!CONTACTOI_INFECTADO YA EXISTENTE!!!';

    WHEN OTHERS THEN

        P_INFO:= 'ERROR... ENLACE NO REGISTRADA';
```

COMMIT;

END REG_CONTACINFECT;

/

```
SQL> CREATE OR REPLACE PROCEDURE REG_CONTACINFECT (  
  2  P_NAME                contacto_de_afectado.Nombre%TYPE,  
  3  P_TELEFONO            contacto_de_afectado.telefono%TYPE,  
  4  P_DIRECCION            contacto_de_afectado.direccion%TYPE,  
  5  P_INFO out varchar2)  
  6  AS  
  7  BEGIN  
  8    INSERT INTO contacto_de_afectado VALUES('PI' || TO_CHAR(N_perconta.n  
extval),P_NAME,P_TELEFONO,P_DIRECCION);  
  9    P_INFO := 'ENLACE MEDICO REGISTRADO';  
 10  EXCEPTION  
 11  WHEN DUP_VAL_ON_INDEX THEN  
 12    P_INFO:= '!!!CONTACTOI_INFECTADO YA EXISTENTE!!!';  
 13  WHEN OTHERS THEN  
 14    P_INFO:= 'ERROR... ENLACE NO REGISTRADA';  
 15  COMMIT;  
 16  END REG_CONTACINFECT;  
 17  /
```

Procedure created.

8 Procedimiento. Para registrar el enlace del paciente positivo y contactos de estos, para obtener vista información precisa, desde sus llaves principales.

```
CREATE OR REPLACE PROCEDURE PACIEN_CONTACINFECT (  
P_COD_PACIENTE  paciente_contacto.codig_paciente%TYPE,  
P_COD_CONTACT  paciente_contacto.cod_contact%TYPE,  
P_INFO out varchar2)  
AS  
BEGIN  
  INSERT  
  INTO paciente_contacto VALUES(P_COD_PACIENTE, P_COD_CONTACT);  
  P_INFO := 'ENLACE REGISTRADO';  
EXCEPTION  
  WHEN DUP_VAL_ON_INDEX THEN  
    P_INFO:= '!!!REGISTRO YA EXISTENTE!!!';  
  WHEN OTHERS THEN  
    P_INFO:= 'ERROR... ENLACE NO REGISTRADA';  
COMMIT;  
END PACIEN_CONTACINFECT;  
/
```

```
SQL> CREATE OR REPLACE PROCEDURE PACIEN_CONTACINFECT (  
2  P_COD_PACIENTE  paciente_contacto.codig_paciente%TYPE,  
3  P_COD_CONTACT  paciente_contacto.cod_contact%TYPE,  
4  P_INFO out varchar2)  
5  AS  
6  BEGIN  
7    INSERT INTO paciente_contacto VALUES(P_COD_PACIENTE, P_COD_CONTACT);  
8    P_INFO := 'ENLACE REGISTRADO';  
9  EXCEPTION  
10 WHEN DUP_VAL_ON_INDEX THEN  
11 P_INFO:= '!!!REGISTRO YA EXISTENTE!!!';  
12 WHEN OTHERS THEN  
13 P_INFO:= 'ERROR... ENLACE NO REGISTRADA';  
14 COMMIT;  
15 END PACIEN_CONTACINFECT;  
16 /
```

Procedure created.

9 Procedimiento. Para registrar personal médico.

```
CREATE OR REPLACE PROCEDURE PERSONAL_MED (  
  P_COD_REGISTRO personal_de_salud.registro%TYPE,  
  P_NOMBRE personal_de_salud.nombre%TYPE,  
  P_CEDULA personal_de_salud.cedula%TYPE,  
  P_TELEFONO personal_de_salud.telefono%TYPE,  
  P_COD_OCUPACION personal_de_salud.ocupacion%TYPE,  
  P_COD_HOTEL personal_de_salud.cod_hotel%TYPE,  
  P_INFO out varchar2)  
AS  
ex_encontrado EXCEPTION;  
v_ORDEN number := 1;  
  cursor c_medico is  
    SELECT *  
    FROM personal_de_salud  
    WHERE cedula = P_CEDULA;  
BEGIN  
  FOR hotel in c_medico LOOP  
    IF c_medico%ROWCOUNT > 0 THEN  
      v_ORDEN := 0;  
      RAISE ex_encontrado;  
    END IF;  
  END LOOP;  
  IF v_ORDEN = 1 THEN  
    INSERT  
    INTO personal_de_salud VALUES(P_COD_REGISTRO,P_NOMBRE,P_CEDULA,  
    P_TELEFONO,P_COD_OCUPACION,P_COD_HOTEL);  
    P_INFO := 'MEDICO REGISTRADO';  
  END IF;  
  
EXCEPTION
```

```

    WHEN ex_encontrado THEN
        P_INFO := '!!!REGISTRO EXISTENTE!!!';
    WHEN OTHERS THEN
        P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';
COMMIT;
END PERSONAL_MED;
/

```

```

SQL> CREATE OR REPLACE PROCEDURE PERSONAL_MED (
  2  P_COD_REGISTRO personal_de_salud.registro%TYPE,
  3  P_NOMBRE personal_de_salud.nombre%TYPE,
  4  P_CEDULA personal_de_salud.cedula%TYPE,
  5  P_TELEFONO personal_de_salud.telefono%TYPE,
  6  P_COD_OCUPACION personal_de_salud.ocupacion%TYPE,
  7  P_COD_HOTEL personal_de_salud.cod_hotel%TYPE,
  8  P_INFO out varchar2)
  9  AS
10  ex_encontrado EXCEPTION;
11  v_ORDEN number := 1;
12  cursor c_medico is
13  SELECT *
14  FROM personal_de_salud
15  WHERE cedula = P_CEDULA;
16  BEGIN
17  FOR hotel in c_medico LOOP
18  IF c_medico%ROWCOUNT > 0 THEN
19  v_ORDEN := 0;
20  RAISE ex_encontrado;
21  END IF;
22  END LOOP;
23  IF v_ORDEN = 1 THEN
24  INSERT INTO personal_de_salud VALUES(P_COD_REGISTRO,P_NOMBRE,P_CEDULA,
25  _TELEFONO,P_COD_OCUPACION,P_COD_HOTEL);
26  P_INFO := 'MEDICO REGISTRADO';
27  END IF;
28  EXCEPTION
29  WHEN ex_encontrado THEN
30  P_INFO := '!!!REGISTRO EXISTENTE!!!';
31  WHEN OTHERS THEN
32  P_INFO:= 'ERROR... ALGO OCURRIO EN LA ACCION';
33  COMMIT;
34  END PERSONAL_MED;
35  /

Procedure created.

```

10 Procedimiento. Para registrar las diez (10) provincias de Panamá.

```
CREATE OR REPLACE PROCEDURE CREAR_PROVINCIA (  
P_COD_PROVIN    provincia.cod_provincia%TYPE,  
P_NAME          provincia.nombre_provincia%TYPE,  
P_INFO out varchar2)  
AS  
BEGIN  
INSERT INTO provincia VALUES(P_COD_PROVIN, P_NAME);  
P_INFO := 'PROVINCIA REGISTRADA';  
EXCEPTION  
    WHEN DUP_VAL_ON_INDEX THEN  
        P_INFO:= '!!!PROVINCIA EXISTENTE!!!';  
    WHEN OTHERS THEN  
        P_INFO:= 'ERROR... PROVINCIA NO CREADA';  
COMMIT;  
END CREAR_PROVINCIA;  
/
```

```
SQL> CREATE OR REPLACE PROCEDURE CREAR_PROVINCIA (  
2  P_COD_PROVIN    provincia.cod_provincia%TYPE,  
3  P_NAME          provincia.nombre_provincia%TYPE,  
4  P_INFO out varchar2)  
5  AS  
6  BEGIN  
7  INSERT INTO provincia VALUES(P_COD_PROVIN, P_NAME);  
8  P_INFO := 'PROVINCIA REGISTRADA';  
9  EXCEPTION  
10 WHEN DUP_VAL_ON_INDEX THEN  
11 P_INFO:= '!!!PROVINCIA EXISTENTE!!!';  
12 WHEN OTHERS THEN  
13 P_INFO:= 'ERROR... PROVINCIA NO CREADA';  
14 COMMIT;  
15 END CREAR_PROVINCIA;  
16 /  
  
Procedure created.
```

11 Procedimiento. Para crear un enlace entre el historial del paciente positivo covid-19 y el personal de salud.

```
CREATE OR REPLACE PROCEDURE HIST_PERSALUD (  
  P_COD_HISTMED    historial_personalsalud.cod_hm%TYPE,  
  P_REGIST         historial_personalsalud.regist%TYPE,  
  P_DETALLE        historial_personalsalud.detalle%TYPE,  
  P_INFO out varchar2)  
AS  
BEGIN  
  INSERT  
  INTO historial_personalsalud VALUES(P_COD_HISTMED,P_REGIST,SYSDAT  
  E,P_DETALLE);  
  P_INFO := 'REGISTRO CREADO';  
EXCEPTION  
  WHEN DUP_VAL_ON_INDEX THEN  
    P_INFO:= '!!!REGISTRO EXISTENTE!!!';  
  WHEN OTHERS THEN  
    P_INFO:= 'ERROR... NO CREADO';  
COMMIT;  
END HIST_PERSALUD;  
/
```

```

SQL> CREATE OR REPLACE PROCEDURE HIST_PERSALUD (
  2   P_COD_HISTMED      historial_personalsalud.cod_hm%TYPE,
  3   P_REGIST           historial_personalsalud.regist%TYPE,
  4   P_DETALLE          historial_personalsalud.detalle%TYPE,
  5   P_INFO out varchar2)
  6   AS
  7   BEGIN
  8     INSERT INTO historial_personalsalud VALUES(P_COD_HISTMED,P_REGIST,SYSD
ATE,P_DETALLE);
  9     P_INFO := 'REGISTRO CREADO';
 10   EXCEPTION
 11     WHEN DUP_VAL_ON_INDEX THEN
 12       P_INFO:= '!!!REGISTRO EXISTENTE!!!';
 13     WHEN OTHERS THEN
 14       P_INFO:= 'ERROR... NO CREADO';
 15   COMMIT;
 16   END HIST_PERSALUD;
 17   /

```

Procedure created.

6. Pruebas

Ingresando provincias en el procedimiento CREAR_PROVINCIA.

```
set serveroutput on;
DECLARE
    v_error varchar2(100);
BEGIN
    CREAR_PROVINCIA('PRO1','BOCAS DEL TORO',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    CREAR_PROVINCIA('PRO2','COCLE',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    CREAR_PROVINCIA('PRO3','COLON',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    CREAR_PROVINCIA('PRO4','CHIRIQUI',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    CREAR_PROVINCIA('PRO5','DARIEN',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    CREAR_PROVINCIA('PRO6','HERRERA',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    CREAR_PROVINCIA('PRO7','LOS SANTOS',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    CREAR_PROVINCIA('PRO8','PANAMA',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    CREAR_PROVINCIA('PRO9','VERAGUAS',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    CREAR_PROVINCIA('PRO10','PANAMA OESTE',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
END;
/
```

```

SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4      CREAM_PROVINCIA('PRO1','BOCAS DEL TORO',v_error);
  5      DBMS_OUTPUT.PUT_LINE(v_error);
  6      CREAM_PROVINCIA('PRO2','COCLE',v_error);
  7      DBMS_OUTPUT.PUT_LINE(v_error);
  8      CREAM_PROVINCIA('PRO3','COLON',v_error);
  9      DBMS_OUTPUT.PUT_LINE(v_error);
10      CREAM_PROVINCIA('PRO4','CHIRIQUI',v_error);
11      DBMS_OUTPUT.PUT_LINE(v_error);
12      CREAM_PROVINCIA('PRO5','DARIEN',v_error);
13      DBMS_OUTPUT.PUT_LINE(v_error);
14      CREAM_PROVINCIA('PRO6','HERRERA',v_error);
15      DBMS_OUTPUT.PUT_LINE(v_error);
16      CREAM_PROVINCIA('PRO7','LOS SANTOS',v_error);
17      DBMS_OUTPUT.PUT_LINE(v_error);
18      CREAM_PROVINCIA('PRO8','PANAMA',v_error);
19      DBMS_OUTPUT.PUT_LINE(v_error);
20      CREAM_PROVINCIA('PRO9','VERAGUAS',v_error);
21      DBMS_OUTPUT.PUT_LINE(v_error);
22      CREAM_PROVINCIA('PRO10','PANAMA OESTE',v_error);
23      DBMS_OUTPUT.PUT_LINE(v_error);
24  END;
25  /
PROVINCIA REGISTRADA
PROVINCIA REGISTRADA
PROVINCIA REGISTRADA
PROVINCIA REGISTRADA
PROVINCIA REGISTRADA
PROVINCIA REGISTRADA
PROVINCIA REGISTRADA
PROVINCIA REGISTRADA
PROVINCIA REGISTRADA
PROVINCIA REGISTRADA

PL/SQL procedure successfully completed.

```


Ingresando nombres de las enfermedades en el procedimiento

CREAR_ENFERMEDAD.

```
set serveroutput on;
```

```
DECLARE
```

```
    v_error varchar2(100);
```

```
BEGIN
```

```
    CREAR_ENFERMEDAD('ENF1','CARDIOPATIA CRONICA',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF2','HIPERTENSION',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF3','NEUMOPATIA CRONICA',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF4','ASMA',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF5','INSUFICIENCIA RENAL CRONICA',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF6','HEPATOPATIA CRONICA',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF7',' TRASTORNO NEUROLOGICO  
CRONICO',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF8','INFECCIÓN POR EL VIH',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF9','DIABETES',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF10','TUBERCULOSIS',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF11','ASPLENIA',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_ENFERMEDAD('ENF12','NEOPLASIA MALIGNA',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

END;

/

```
SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4      CREAR_ENFERMEDAD('ENF1','CARDIOPATIA CRONICA',v_error);
  5      DBMS_OUTPUT.PUT_LINE(v_error);
  6      CREAR_ENFERMEDAD('ENF2','HIPERTENSION',v_error);
  7      DBMS_OUTPUT.PUT_LINE(v_error);
  8      CREAR_ENFERMEDAD('ENF3','NEUMOPATIA CRONICA',v_error);
  9      DBMS_OUTPUT.PUT_LINE(v_error);
 10      CREAR_ENFERMEDAD('ENF4','ASMA',v_error);
 11      DBMS_OUTPUT.PUT_LINE(v_error);
 12      CREAR_ENFERMEDAD('ENF5','INSUFICIENCIA RENAL CRONICA',v_error);
 13      DBMS_OUTPUT.PUT_LINE(v_error);
 14      CREAR_ENFERMEDAD('ENF6','HEPATOPATIA CRONICA',v_error);
 15      DBMS_OUTPUT.PUT_LINE(v_error);
 16      CREAR_ENFERMEDAD('ENF7','TRASTORNO NEUROLOGICO CRONICO',v_error);
 17      DBMS_OUTPUT.PUT_LINE(v_error);
 18      CREAR_ENFERMEDAD('ENF8','INFECCIÓN POR EL VIH',v_error);
 19      DBMS_OUTPUT.PUT_LINE(v_error);
 20      CREAR_ENFERMEDAD('ENF9','DIABETES',v_error);
 21      DBMS_OUTPUT.PUT_LINE(v_error);
 22      CREAR_ENFERMEDAD('ENF10','TUBERCULOSIS',v_error);
 23      DBMS_OUTPUT.PUT_LINE(v_error);
 24      CREAR_ENFERMEDAD('ENF11','ASPLENIA',v_error);
 25      DBMS_OUTPUT.PUT_LINE(v_error);
 26      CREAR_ENFERMEDAD('ENF12','NEOPLASIA MALIGNA',v_error);
 27      DBMS_OUTPUT.PUT_LINE(v_error);
 28  END;
 29  /

ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA
ENFERMEDAD REGISTRADA

PL/SQL procedure successfully completed.
```

Ingresando las ocupaciones del personal de la salud.

```
set serveroutput on;
```

```
DECLARE
```

```
    v_error varchar2(100);
```

```
BEGIN
```

```
    CREAR_OCUP('OCU1','MEDICO',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
    CREAR_OCUP('OCU2','ENFERMERA',v_error);
```

```
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```
END;
```

```
/
```

```
SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4      CREAR_OCUP('OCU1','MEDICO',v_error);
  5      DBMS_OUTPUT.PUT_LINE(v_error);
  6      CREAR_OCUP('OCU2','ENFERMERA',v_error);
  7      DBMS_OUTPUT.PUT_LINE(v_error);
  8  END;
  9  /
PROFECION REGISTRADA
PROFECION REGISTRADA

PL/SQL procedure successfully completed.
```

Ingresando los Hoteles que albergan personas positivas Covid-19 en Panamá.

```
set serveroutput on;
DECLARE
v_error varchar2(100);
BEGIN
REG_HOTEL ('Royal Mansion', 'Chilibre', 'PRO8', 20,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Plaza', 'Santiago', 'PRO9', 15,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Mykonos', 'Santiago', 'PRO9', 12,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Lisboa', 'Calidonia', 'PRO8', 10,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Latino', 'Calidonia', 'PRO8', 29,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Marbella', 'Bella Vista', 'PRO8', 20,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Terranova', 'Bella Vista', 'PRO8', 15,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Residencial y Hotel Palazzo', 'Calidonia', 'PRO8', 16,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Aramo', 'Alanje', 'PRO9', 12,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Costa Inn', 'Calidonia', 'PRO8', 12,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Montreal', 'Calidonia', 'PRO8', 25,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Santa', 'Santiago', 'PRO9', 15,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Mira Azul', 'Santiago', 'PRO9', 24,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
```

```

REG_HOTEL ('Hotel Cabu', 'David', 'PRO9', 16,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HOTEL ('Hotel Mariano', 'David', 'PRO9', 10,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
END;
/

```

```

SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4  REG_HOTEL ('Royal Mansion', 'Chilibre', 'PRO8', 20,v_error);
  5  DBMS_OUTPUT.PUT_LINE(v_error);
  6  REG_HOTEL ('Hotel Plaza', 'Santiago', 'PRO9', 15,v_error);
  7  DBMS_OUTPUT.PUT_LINE(v_error);
  8  REG_HOTEL ('Hotel Mykonos', 'Santiago', 'PRO9', 12,v_error);
  9  DBMS_OUTPUT.PUT_LINE(v_error);
 10  REG_HOTEL ('Hotel Lisboa', 'Calidonia', 'PRO8', 10,v_error);
 11  DBMS_OUTPUT.PUT_LINE(v_error);
 12  REG_HOTEL ('Hotel Latino', 'Calidonia', 'PRO8', 29,v_error);
 13  DBMS_OUTPUT.PUT_LINE(v_error);
 14  REG_HOTEL ('Hotel Marbella', 'Bella Vista', 'PRO8', 20,v_error);
 15  DBMS_OUTPUT.PUT_LINE(v_error);
 16  REG_HOTEL ('Hotel Terranova', 'Bella Vista', 'PRO8', 15,v_error);
 17  DBMS_OUTPUT.PUT_LINE(v_error);
 18  REG_HOTEL ('Residencial y Hotel Palazzo', 'Calidonia', 'PRO8', 16,v_err
or);
 19  DBMS_OUTPUT.PUT_LINE(v_error);
 20  REG_HOTEL ('Hotel Aramo', 'Alanje', 'PRO9', 12,v_error);
 21  DBMS_OUTPUT.PUT_LINE(v_error);
 22  REG_HOTEL ('Hotel Costa Inn', 'Calidonia', 'PRO8', 12,v_error);
 23  DBMS_OUTPUT.PUT_LINE(v_error);
 24  REG_HOTEL ('Hotel Montreal', 'Calidonia', 'PRO8', 25,v_error);
 25  DBMS_OUTPUT.PUT_LINE(v_error);
 26  REG_HOTEL ('Hotel Santa', 'Santiago', 'PRO9', 15,v_error);
 27  DBMS_OUTPUT.PUT_LINE(v_error);
 28  REG_HOTEL ('Hotel Mira Azul', 'Santiago', 'PRO9', 24,v_error);
 29  DBMS_OUTPUT.PUT_LINE(v_error);
 30  REG_HOTEL ('Hotel Cabu', 'David', 'PRO9', 16,v_error);
 31  DBMS_OUTPUT.PUT_LINE(v_error);
 32  REG_HOTEL ('Hotel Mariano', 'David', 'PRO9', 10,v_error);
 33  DBMS_OUTPUT.PUT_LINE(v_error);
 34  END;
 35  /

```

```
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!  
!!!REGISTRO EXITOSO!!!
```

```
PL/SQL procedure successfully completed.
```

Ingresando personal de salud que se encargara de atender a los pacientes.

```
set serveroutput on;
DECLARE
    v_error varchar2(100);
BEGIN
PERSONAL_MED('34568', 'Juan Mendoza','8-43-4324','6245-
6479','OCU1',1001,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('98754', 'Jose Cabrera','8-430-890','6285-
3545','OCU1',1002,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('12768', 'Manuel Chanapi','8-160-565','6462-
7932','OCU1',1003,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('97450', 'Maria Escobar','8-7382-43','6245-
6479','OCU2',1004,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('45387', 'Franciso Palacio','8-244-324','6743-
8023','OCU1',1005,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('98433', 'Gianys Asprila','4-546-234','6576-
0391','OCU2',1006,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('47765', 'Miriam Zarco','4-541-3499','6845-
1232','OCU2',1007,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('45667', 'Juana Ayalá','4-7543-3107','6832-
4342','OCU1',1008,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('45777', 'Antonio Uribe','4-157-1023','6333-
2136','OCU1',1009,v_error);
```

```
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('47725', 'Melanie Abrego','4-469-1735','6943-
2388','OCU2',1010,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('35449', 'Miguel Garabato','1-32-457','6432-
7645','OCU1',1011,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('88823', 'Pedro Sandoval','1-354-6688','6232-
5455','OCU1',1012,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('47715', 'Helen Cardenas','1-390-607','6832-
1235','OCU2',1013,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('95122', 'Martin Vergara','6-948-3838','6312-
3490','OCU1',1014,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PERSONAL_MED ('54311', 'Margarita Arjona','6-894-607','6423-
7943','OCU2',1015,v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
END;
/
```



```

SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4  PERSONAL_MED('34568', 'Juan Mendoza','8-43-4324','6245-6479','OCU1'
,1001,v_error);
  5  DBMS_OUTPUT.PUT_LINE(v_error);
  6  PERSONAL_MED ('98754', 'Jose Cabrera','8-430-890','6285-3545','OCU1'
',1002,v_error);
  7  DBMS_OUTPUT.PUT_LINE(v_error);
  8  PERSONAL_MED ('12768', 'Manuel Chanapi','8-160-565','6462-7932','OC
U1',1003,v_error);
  9  DBMS_OUTPUT.PUT_LINE(v_error);
 10  PERSONAL_MED ('97450', 'Maria Escobar','8-7382-43','6245-6479','OCU
2',1004,v_error);
 11  DBMS_OUTPUT.PUT_LINE(v_error);
 12  PERSONAL_MED ('45387', 'Franciso Palacio','8-244-324','6743-8023','
OCU1',1005,v_error);
 13  DBMS_OUTPUT.PUT_LINE(v_error);
 14  PERSONAL_MED ('98433', 'Gianys Asprila','4-546-234','6576-0391','OC
U2',1006,v_error);
 15  DBMS_OUTPUT.PUT_LINE(v_error);
 16  PERSONAL_MED ('47765', 'Miriam Zarco','4-541-3499','6845-1232','OCU
2',1007,v_error);
 17  DBMS_OUTPUT.PUT_LINE(v_error);
 18  PERSONAL_MED ('45667', 'Juana Ayalá','4-7543-3107','6832-4342','OCU
1',1008,v_error);
 19  DBMS_OUTPUT.PUT_LINE(v_error);
 20  PERSONAL_MED ('45777', 'Antonio Uribe','4-157-1023','6333-2136','OC
U1',1009,v_error);
 21  DBMS_OUTPUT.PUT_LINE(v_error);
 22  PERSONAL_MED ('47725', 'Melanie Abrego','4-469-1735','6943-2388','O
CU2',1010,v_error);
 23  DBMS_OUTPUT.PUT_LINE(v_error);
 24  PERSONAL_MED ('35449', 'Miguel Garabato','1-32-457','6432-7645','OC
U1',1011,v_error);
 25  DBMS_OUTPUT.PUT_LINE(v_error);

```


Ingresando datos al historial medico de los pacientes.

```
set serveroutput on;
DECLARE
v_error varchar2(100);
BEGIN
REG_HISTMED(TO_DATE('2020-05-10','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-06-01','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-06-15','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-05-29','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-05-17','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-05-10','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-06-12','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-05-26','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-06-20','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-06-28','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-06-03','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-06-04','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-05-10','yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
```

```

REG_HISTMED(TO_DATE('2020-06-12', 'yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_HISTMED(TO_DATE('2020-06-23', 'yyyy-mm-dd'),'positivo',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
END;
/

```

```

SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4  REG_HISTMED(TO_DATE('2020-05-10', 'yyyy-mm-dd'),'positivo',v_error);
  DBMS_OUTPUT.PUT_LINE(v_error);
  5  REG_HISTMED(TO_DATE('2020-06-01', 'yyyy-mm-dd'),'positivo',v_error);

  6  DBMS_OUTPUT.PUT_LINE(v_error);
  7  REG_HISTMED(TO_DATE('2020-06-15', 'yyyy-mm-dd'),'positivo',v_error)
  ;
  8  DBMS_OUTPUT.PUT_LINE(v_error);
  9  REG_HISTMED(TO_DATE('2020-05-29', 'yyyy-mm-dd'),'positivo',v_error)
  ;
 10  DBMS_OUTPUT.PUT_LINE(v_error);
 11  REG_HISTMED(TO_DATE('2020-05-17', 'yyyy-mm-dd'),'positivo',v_error)
  ;
 12  DBMS_OUTPUT.PUT_LINE(v_error);
 13  REG_HISTMED(TO_DATE('2020-05-10', 'yyyy-mm-dd'),'positivo',v_error)
  ;
 14  DBMS_OUTPUT.PUT_LINE(v_error);
 15  REG_HISTMED(TO_DATE('2020-06-12', 'yyyy-mm-dd'),'positivo',v_error)
  ;
 16  DBMS_OUTPUT.PUT_LINE(v_error);
 17  REG_HISTMED(TO_DATE('2020-05-26', 'yyyy-mm-dd'),'positivo',v_error)
  ;
 18  DBMS_OUTPUT.PUT_LINE(v_error);
 19  REG_HISTMED(TO_DATE('2020-06-20', 'yyyy-mm-dd'),'positivo',v_error)
  ;
 20  DBMS_OUTPUT.PUT_LINE(v_error);
 21  REG_HISTMED(TO_DATE('2020-06-28', 'yyyy-mm-dd'),'positivo',v_error)
  ;
 22  DBMS_OUTPUT.PUT_LINE(v_error);
 23  REG_HISTMED(TO_DATE('2020-06-03', 'yyyy-mm-dd'),'positivo',v_error)
  ;
 24  DBMS_OUTPUT.PUT_LINE(v_error);
 25  REG_HISTMED(TO_DATE('2020-06-04', 'yyyy-mm-dd'),'positivo',v_error)
  );
 26  DBMS_OUTPUT.PUT_LINE(v_error);

```

[illegible]

Ingresando pacientes.

```
set serveroutput on;
DECLARE
    v_error varchar2(100);
BEGIN
    REG_PACIENTE('8-423-1988','Alexander','Betancur',TO_DATE('1993-05-16','yyyy-mm-dd'),'masculino','275-3688','El
naranja','Cañita','PRO8','a101',1001,v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_PACIENTE('2-399-456','Maria','Pineda',TO_DATE('1963-01-03','yyyy-mm-dd'),'femenino','298-1234','Rana de oro','Pedregal','PRO8','a102',1002,v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_PACIENTE('5-786-8009','Egidio','Bustamante',TO_DATE('1993-03-05','yyyy-mm-dd'),'masculino','390-7891','Genesis','Mañanitas','PRO8','a103',1003,v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_PACIENTE('8-987-999','Haygnes','Taborda',TO_DATE('1977-06-19','yyyy-mm-dd'),'masculino','291-5467','El llano','Pacora','PRO8','a104',1004,v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_PACIENTE('1-809-1004','Alexandra','Cardenas',TO_DATE('1968-03-23','yyyy-mm-dd'),'femenino','298-1567','Chepo','Las
Margaritas','PRO8','a105',1005,v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_PACIENTE('9-786-892','Elena','Chaverra',TO_DATE('1988-12-12','yyyy-mm-dd'),'femenino','390-5612','sector 2','Las Garzas','PRO8','a106',1006,v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_PACIENTE('9-877-8768','Jairo','Cardona',TO_DATE('1952-10-04','yyyy-mm-dd'),'masculino','567-1267','La cana','Torti','PRO8','a107',1007,v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
```

```

REG_PACIENTE('8-903-1189', 'Albeiro', 'Loaiza', TO_DATE('1973-09-15', 'yyyy-
mm-dd'), 'masculino', '291-7865', 'Belen', 'Tocumen', 'PRO8',
'a108', 1008, v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_PACIENTE('9-545-234', 'Reinaldo', 'Arango', TO_DATE('1963-07-01',
'yyyy-mm-dd'), 'masculino', '298-6555', 'Nuevo Tocumen', '24 de Diciembre',
'PRO8', 'a109', 1009, v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_PACIENTE('1-434-540', 'Alexia', 'Posada', TO_DATE('1979-08-09', 'yyyy-
mm-dd'), 'femenino', '299-7877', 'El arbol', 'Alcalde Díaz', 'PRO8',
'a110', 1010, v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_PACIENTE('3-435-4323', 'Murat', 'Martinez', TO_DATE('1982-01-12', 'yyyy-
mm-dd'), 'masculino', '234-6789', 'Casa real', 'Ancón', 'PRO8', 'a111', 1011, v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_PACIENTE('4-545-1234', 'Javier', 'Zea', TO_DATE('1950-03-06',
'yyyy/mm/dd'), 'masculino', '333-1232', 'enfrente del
parque andres bello', 'Bellavista', 'PRO8', 'a112', 1012, v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_PACIENTE('3-903-2133', 'Roxana', 'Mesa', TO_DATE('1963-09-16',
'yyyy/mm/dd'), 'femenino', '321-3213', 'Santa
Maria', 'Betania', 'PRO8', 'a113', 1013, v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_PACIENTE('5-435-12', 'Dara', 'Taborda', TO_DATE('1961-05-17', 'yyyy-mm-
dd'), 'femenino', '323-3232', 'Guarumal', 'Caimitillo', 'PRO8', 'a114', 1014, v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_PACIENTE('8-34-2343', 'Sarah', 'Henao', TO_DATE('1975-02-27', 'yyyy-mm-
dd'), 'femenino', '255-5555', 'enfrente del parque los aburridos', 'Calidonia',
'PRO8', 'a115', 1015, v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
End;

```

/

```
SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4  REG_PACIENTE('8-423-1988','Alexander','Betancur',TO_DATE('1993-05-1
6','yyyy-mm-dd'),'masculino','275-3688','El naranjal','Cañita','PRO8','a
101',1001,v_error);
  5  DBMS_OUTPUT.PUT_LINE(v_error);
  6  REG_PACIENTE('2-399-456','Maria','Pineda',TO_DATE('1963-01-03','yyy
y-mm-dd'),'femenino','298-1234','Rana de oro','Pedregal','PRO8','a102',1
002,v_error);
  7  DBMS_OUTPUT.PUT_LINE(v_error);
  8  REG_PACIENTE('5-786-8009','Egidio','Bustamante',TO_DATE('1993-03-05
','yyyy-mm-dd'),'masculino','390-7891','Genesis','Mañanitas','PRO8',
'a103',1003,v_error);
  9  DBMS_OUTPUT.PUT_LINE(v_error);
 10  REG_PACIENTE('8-987-999','Haygnes','Taborda',TO_DATE('1977-06-19
','yyyy-mm-dd'),'masculino','291-5467','El llano','Pacora','PRO8','a
104',1004,v_error);
 11  DBMS_OUTPUT.PUT_LINE(v_error);
 12  REG_PACIENTE('1-809-1004','Alexandra','Cardenas',TO_DATE('1968-03
-23','yyyy-mm-dd'),'femenino','298-1567','Chepo','Las Margaritas',
'PRO8','a105',1005,v_error);
 13  DBMS_OUTPUT.PUT_LINE(v_error);
 14  REG_PACIENTE('9-786-892','Elena','Chaverra',TO_DATE('1988-12-12'
,'yyyy-mm-dd'),'femenino','390-5612','sector 2','Las Garzas','PRO8'
,'a106',1006,v_error);
 15  DBMS_OUTPUT.PUT_LINE(v_error);
 16  REG_PACIENTE('9-877-8768','Jairo','Cardona',TO_DATE('1952-10-04'
,'yyyy-mm-dd'),'masculino','567-1267','La cana','Torti','PRO8','a1
07',1007,v_error);
 17  DBMS_OUTPUT.PUT_LINE(v_error);
 18  REG_PACIENTE('8-903-1189','Albeiro','Loaiza',TO_DATE('1973-09-15
','yyyy-mm-dd'),'masculino','291-7865','Belen','Tocumen','PRO8','a
108',1008,v_error);
 19  DBMS_OUTPUT.PUT_LINE(v_error);
 20  REG_PACIENTE('9-545-234','Reinaldo','Arango',TO_DATE('1963-07-01
','yyyy-mm-dd'),'masculino','298-6555','Nuevo Tocumen','24 de Diciem
bre','PRO8','a109',1009,v_error);
 21  DBMS_OUTPUT.PUT_LINE(v_error);
```



```

22 REG_PACIENTE('1-434-540', 'Alexia', 'Posada', TO_DATE('1979-08-09',
'yyyy-mm-dd'), 'femenino', '299-7877', 'El arbol', 'Alcalde Díaz', 'PRO8'
, 'a110', 1010, v_error);
23 DBMS_OUTPUT.PUT_LINE(v_error);
24 REG_PACIENTE('3-435-4323', 'Murat', 'Martinez', TO_DATE('1982-01-12',
'yyyy-mm-dd'), 'masculino', '234-6789', 'Casa real', 'Ancón', 'PRO8', 'a111', 1
011, v_error);
25 DBMS_OUTPUT.PUT_LINE(v_error);
26 REG_PACIENTE('4-545-1234', 'Javier', 'Zea', TO_DATE('1950-03-06', 'yyy
y/mm/dd'), 'masculino', '333-1232', 'enfrente del parque andres bello', 'Bel
lavista', 'PRO8', 'a112', 1012, v_error);
27 DBMS_OUTPUT.PUT_LINE(v_error);
28 REG_PACIENTE('3-903-2133', 'Roxana', 'Mesa', TO_DATE('1963-09-16', 'yy
yy/mm/dd'), 'femenino', '321-3213', 'Santa Maria', 'Betania', 'PRO8', 'a113', 1
013, v_error);
29 DBMS_OUTPUT.PUT_LINE(v_error);
30 REG_PACIENTE('5-435-12', 'Dara', 'Taborda', TO_DATE('1961-05-17', 'yyy
y-mm-dd'), 'femenino', '323-3232', 'Guarumal', 'Caimitillo', 'PRO8', 'a114', 10
14, v_error);
31 DBMS_OUTPUT.PUT_LINE(v_error);
32 REG_PACIENTE('8-34-2343', 'Sarah', 'Henao', TO_DATE('1975-02-27', 'yyy
y-mm-dd'), 'femenino', '255-5555', 'enfrente del parque los aburridos', 'Cal
idonia', 'PRO8', 'a115', 1015, v_error);
33 DBMS_OUTPUT.PUT_LINE(v_error);
34 End;
35 /
PACIENTE REGISTRADO: Alexander, Betancur, 8-423-1988
PACIENTE REGISTRADO: Maria, Pineda, 2-399-456
PACIENTE REGISTRADO: Egidio, Bustamante, 5-786-8009
PACIENTE REGISTRADO: Haygnes, Taborda, 8-987-999
PACIENTE REGISTRADO: Alexandra, Cardenas, 1-809-1004
PACIENTE REGISTRADO: Elena, Chaverra, 9-786-892
PACIENTE REGISTRADO: Jairo, Cardona, 9-877-8768
PACIENTE REGISTRADO: Albeiro, Loaiza, 8-903-1189
PACIENTE REGISTRADO: Reinaldo, Arango, 9-545-234
PACIENTE REGISTRADO: Alexia, Posada, 1-434-540
PACIENTE REGISTRADO: Murat, Martinez, 3-435-4323
PACIENTE REGISTRADO: Javier, Zea, 4-545-1234
PACIENTE REGISTRADO: Roxana, Mesa, 3-903-2133
PACIENTE REGISTRADO: Dara, Taborda, 5-435-12
PACIENTE REGISTRADO: Sarah, Henao, 8-34-2343

PL/SQL procedure successfully completed.

```

Ingresando datos que describen el personal de salud cuando visita al paciente.

```
set serveroutput on;
DECLARE
v_error varchar2(100);
BEGIN
HIST_PERSALUD ('a101','34568', 'No presenta fiebre, ni dolor de
garganta.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a102','98754', 'Dolor de garganta.', v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a103','12768', 'Presenta mucho de dolor de
cabeza.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a104','97450', 'Presenta dolores estomacales.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a105','45387', 'Falta de apetito y dolores musculares.',
v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a106','98433', 'No presenta fiebre.', v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a107','47765', 'Dolor de cabeza fuerte.', v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a108','45667', 'Presenta Fiebre alta, dolor de
garganta.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a109','45777', 'Presenta mareos, vomito y
cansancio.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a110','47725', 'Presenta mareos.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
```

```
HIST_PERSALUD      ('a111','35449',    'Presenta vomito, dolores
musculares.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a112','88823', 'Presenta glandulas inflamadas.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a113','47715', 'Presenta mala circulacion.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a114','95122', 'Perdida de apetito.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
HIST_PERSALUD ('a115','54311', 'Presenta migraña.',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
END;
/
```

```

SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4  HIST_PERSALUD ('a101','34568', 'No presenta fiebre, ni dolor de ga
rganta.',v_error);
  5  DBMS_OUTPUT.PUT_LINE(v_error);
  6  HIST_PERSALUD ('a102','98754', 'Dolor de garganta.', v_error);
  7  DBMS_OUTPUT.PUT_LINE(v_error);
  8  HIST_PERSALUD ('a103','12768', 'Presenta mucho de dolor de cabeza.
',v_error);
  9  DBMS_OUTPUT.PUT_LINE(v_error);
 10  HIST_PERSALUD ('a104','97450', 'Presenta dolores estomacales.',v_e
rror);
 11  DBMS_OUTPUT.PUT_LINE(v_error);
 12  HIST_PERSALUD ('a105','45387', 'Falta de apetito y dolores muscula
res.', v_error);
 13  DBMS_OUTPUT.PUT_LINE(v_error);
 14  HIST_PERSALUD ('a106','98433', 'No presenta fiebre.', v_error);
 15  DBMS_OUTPUT.PUT_LINE(v_error);
 16  HIST_PERSALUD ('a107','47765', 'Dolor de cabeza fuerte.', v_error)
;
 17  DBMS_OUTPUT.PUT_LINE(v_error);
 18  HIST_PERSALUD ('a108','45667', 'Presenta Fiebre alta, dolor de gar
ganta.',v_error);
 19  DBMS_OUTPUT.PUT_LINE(v_error);
 20  HIST_PERSALUD ('a109','45777', 'Presenta mareos, vomito y cansanci
o.',v_error);
 21  DBMS_OUTPUT.PUT_LINE(v_error);
 22  HIST_PERSALUD ('a110','47725', 'Presenta mareos.',v_error);
 23  DBMS_OUTPUT.PUT_LINE(v_error);
 24  HIST_PERSALUD ('a111','35449', 'Presenta vomito, dolores musculare
s.',v_error);
 25  DBMS_OUTPUT.PUT_LINE(v_error);
 26  HIST_PERSALUD ('a112','88823', 'Presenta glandulas inflamadas.',v_
error);
 27  DBMS_OUTPUT.PUT_LINE(v_error);
 28  HIST_PERSALUD ('a113','47715', 'Presenta mala circulacion.',v erro

```

```
r);
29 DBMS_OUTPUT.PUT_LINE(v_error);
30 HIST_PERSALUD ('a114','95122', 'Perdida de apetito.',v_error);
31 DBMS_OUTPUT.PUT_LINE(v_error);
32 HIST_PERSALUD ('a115','54311', 'Presenta migraña.',v_error);
33 DBMS_OUTPUT.PUT_LINE(v_error);
34 END;
35 /
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
REGISTRO CREADO
PL/SQL procedure successfully completed.
```

Ingresa enlace entre historial medico y el nombre de enfermedad que el paciente padezca.

```
set serveroutput on;
DECLARE
    v_error varchar2(100);
BEGIN
    EN_HISTORIAL('a101','ENF8',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a102','ENF12',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a103','ENF4',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a104','ENF2',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a105','ENF5',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a106','ENF4',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a107','ENF2',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a108','ENF8',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a109','ENF2',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a110','ENF2',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a111','ENF2',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a112','ENF5',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a113','ENF2',v_error);
```

```

        DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a114','ENF6',v_error);
        DBMS_OUTPUT.PUT_LINE(v_error);
    EN_HISTORIAL('a115','ENF2',v_error);
        DBMS_OUTPUT.PUT_LINE(v_error);
END;
/

```

```

SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4      EN_HISTORIAL('a101','ENF8',v_error);
  5      DBMS_OUTPUT.PUT_LINE(v_error);
  6      EN_HISTORIAL('a102','ENF12',v_error);
  7      DBMS_OUTPUT.PUT_LINE(v_error);
  8      EN_HISTORIAL('a103','ENF4',v_error);
  9      DBMS_OUTPUT.PUT_LINE(v_error);
 10      EN_HISTORIAL('a104','ENF2',v_error);
 11      DBMS_OUTPUT.PUT_LINE(v_error);
 12      EN_HISTORIAL('a105','ENF5',v_error);
 13      DBMS_OUTPUT.PUT_LINE(v_error);
 14      EN_HISTORIAL('a106','ENF4',v_error);
 15      DBMS_OUTPUT.PUT_LINE(v_error);
 16      EN_HISTORIAL('a107','ENF2',v_error);
 17      DBMS_OUTPUT.PUT_LINE(v_error);
 18      EN_HISTORIAL('a108','ENF8',v_error);
 19      DBMS_OUTPUT.PUT_LINE(v_error);
 20      EN_HISTORIAL('a109','ENF2',v_error);
 21      DBMS_OUTPUT.PUT_LINE(v_error);
 22      EN_HISTORIAL('a110','ENF2',v_error);
 23      DBMS_OUTPUT.PUT_LINE(v_error);
 24      EN_HISTORIAL('a111','ENF2',v_error);
 25      DBMS_OUTPUT.PUT_LINE(v_error);
 26      EN_HISTORIAL('a112','ENF5',v_error);
 27      DBMS_OUTPUT.PUT_LINE(v_error);
 28      EN_HISTORIAL('a113','ENF2',v_error);
 29      DBMS_OUTPUT.PUT_LINE(v_error);
 30      EN_HISTORIAL('a114','ENF6',v_error);
 31      DBMS_OUTPUT.PUT_LINE(v_error);
 32      EN_HISTORIAL('a115','ENF2',v_error);
 33      DBMS_OUTPUT.PUT_LINE(v_error);
 34  END;
 35  /

```

```
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
ENLACE MEDICO REGISTRADO
PL/SQL procedure successfully completed.
```

Ingresa la información de las personas que son contactos con los pacientes positivos Covid-19.

```
set serveroutput on;
DECLARE
    v_error varchar2(100);
BEGIN

    REG_CONTACINFECT ('Melly Santos', '298-7664', 'Bellavista',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_CONTACINFECT('Gabriel Diaz', '291-7856', 'Chepo',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_CONTACINFECT('Amelia Cami', '391-8765', 'Cañita',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_CONTACINFECT('Mirna Quintero', '290-1717', 'Chorillo',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_CONTACINFECT('Diego Ramirez', '345-8973', 'Pedregal',v_error);
    DBMS_OUTPUT.PUT_LINE(v_error);
    REG_CONTACINFECT('Evelyn Caibera', '391-5643', 'Mañanitas',v_error);
```



```
DBMS_OUTPUT.PUT_LINE(v_error);
REG_CONTACINFECT('Juan Samaniego', '291-5653', 'Las americas',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_CONTACINFECT('Roberto Escamilla', '291-6832', 'Don bosco',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_CONTACINFECT('Aulio Quintero', '390-1822', 'Torrijos Carter',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_CONTACINFECT('Alberto Gonzales', '394-5661', 'Pan de azucar',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_CONTACINFECT('Azucena Robles', '298-2473', 'Calidonia',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_CONTACINFECT('Amilcar Pinzon', '299-2222', 'Villa Zaita',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_CONTACINFECT('Valeria Gongora', '298-1211', 'Chilibre',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_CONTACINFECT('Angel Gonzales', '299-7865', 'Tocumen',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
REG_CONTACINFECT('Rafael Gonzales', '390-1515', 'Tocumen',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
END;
/
```

```

SQL> set serveroutput on;
SQL> DECLARE
  2  v_error varchar2(100);
  3  BEGIN
  4
  5  REG_CONTACINFECT ('Melly Santos', '298-7664', 'Bellavista',v_error)
  ;
  6  DBMS_OUTPUT.PUT_LINE(v_error);
  7  REG_CONTACINFECT('Gabriel Díaz', '291-7856', 'Chepo',v_error);
  8  DBMS_OUTPUT.PUT_LINE(v_error);
  9  REG_CONTACINFECT('Amelia Cami', '391-8765', 'Cañita',v_error);
 10  DBMS_OUTPUT.PUT_LINE(v_error);
 11  REG_CONTACINFECT('Mirna Quintero', '290-1717', 'Chorillo',v_error);

 12  DBMS_OUTPUT.PUT_LINE(v_error);
 13  REG_CONTACINFECT('Diego Ramirez', '345-8973', 'Pedregal',v_error);

 14  DBMS_OUTPUT.PUT_LINE(v_error);
 15  REG_CONTACINFECT('Evelyn Caibera', '391-5643', 'Mañanitas',v_error)
  ;
 16  DBMS_OUTPUT.PUT_LINE(v_error);
 17  REG_CONTACINFECT('Juan Samaniego', '291-5653', 'Las americas',v_err
or);
 18  DBMS_OUTPUT.PUT_LINE(v_error);
 19  REG_CONTACINFECT('Roberto Escamilla', '291-6832', 'Don bosco',v_err
or);
 20  DBMS_OUTPUT.PUT_LINE(v_error);
 21  REG_CONTACINFECT('Aulio Quintero', '390-1822', 'Torrijos Carter',v_
error);
 22  DBMS_OUTPUT.PUT_LINE(v_error);
 23  REG_CONTACINFECT('Alberto Gonzales', '394-5661', 'Pan de azucar',v_
error);
 24  DBMS_OUTPUT.PUT_LINE(v_error);
 25  REG_CONTACINFECT('Azucena Robles', '298-2473', 'Calidonia',v_error)
  ;
 26  DBMS_OUTPUT.PUT_LINE(v_error);
 27  REG_CONTACINFECT('Amilcar Pinzon', '299-2222', 'Villa Zaita',v_erro
r);
 28  DBMS_OUTPUT.PUT_LINE(v_error);
 29  REG_CONTACINFECT('Valeria Gongora', '298-1211', 'Chilibre',v_error)
  ;

```



```
DBMS_OUTPUT.PUT_LINE(v_error);
PACIEN_CONTACINFECT('PA7','PI7',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PACIEN_CONTACINFECT('PA8','PI8',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PACIEN_CONTACINFECT('PA9','PI9',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PACIEN_CONTACINFECT('PA10','PI10',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PACIEN_CONTACINFECT('PA11','PI11',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PACIEN_CONTACINFECT('PA12','PI12',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PACIEN_CONTACINFECT('PA13','PI13',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PACIEN_CONTACINFECT('PA14','PI14',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
PACIEN_CONTACINFECT('PA15','PI15',v_error);
DBMS_OUTPUT.PUT_LINE(v_error);
END;
/
```

```
SQL> set serveroutput on;
SQL>
SQL> DECLARE
  2
  3   v_error varchar2(100);
  4
  5 BEGIN
  6
  7
  8
  9
10
11 PACIEN_CONTACINFECT('PA1','PI1',v_error);
12
13 DBMS_OUTPUT.PUT_LINE(v_error);
14
15
16
17 PACIEN_CONTACINFECT('PA2','PI2',v_error);
18
19 DBMS_OUTPUT.PUT_LINE(v_error);
20
21 PACIEN_CONTACINFECT('PA3','PI3',v_error);
22
23 DBMS_OUTPUT.PUT_LINE(v_error);
24
25 PACIEN_CONTACINFECT('PA4','PI4',v_error);
26
27 DBMS_OUTPUT.PUT_LINE(v_error);
28
29 PACIEN_CONTACINFECT('PA5','PI5',v_error);
30
31 DBMS_OUTPUT.PUT_LINE(v_error);
32
33 PACIEN_CONTACINFECT('PA6','PI6',v_error);
34
35 DBMS_OUTPUT.PUT_LINE(v_error);
36
37 PACIEN_CONTACINFECT('PA7','PI7',v_error);
```

```
37 PACIEN_CONTACINFECT('PA7','PI7',v_error);
38
39 DBMS_OUTPUT.PUT_LINE(v_error);
40
41 PACIEN_CONTACINFECT('PA8','PI8',v_error);
42
43 DBMS_OUTPUT.PUT_LINE(v_error);
44
45 PACIEN_CONTACINFECT('PA9','PI9',v_error);
46
47 DBMS_OUTPUT.PUT_LINE(v_error);
48
49 PACIEN_CONTACINFECT('PA10','PI10',v_error);
50
51 DBMS_OUTPUT.PUT_LINE(v_error);
52
53 PACIEN_CONTACINFECT('PA11','PI11',v_error);
54
55 DBMS_OUTPUT.PUT_LINE(v_error);
56
57 PACIEN_CONTACINFECT('PA12','PI12',v_error);
58
59 DBMS_OUTPUT.PUT_LINE(v_error);
60
61 PACIEN_CONTACINFECT('PA13','PI13',v_error);
62
63 DBMS_OUTPUT.PUT_LINE(v_error);
64
65 PACIEN_CONTACINFECT('PA14','PI14',v_error);
66
67 DBMS_OUTPUT.PUT_LINE(v_error);
68
69 PACIEN_CONTACINFECT('PA15','PI15',v_error);
70
71 DBMS_OUTPUT.PUT_LINE(v_error);
72
73 END;
74
75 /
```

```
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
ENLACE REGISTRADO  
PL/SQL procedure successfully completed.
```

*REALIZAREMOS LAS BUSQUEDAS

```
select *from Tipo_profesion;
```

```
SQL> select *from Tipo_profesion;
```

```
COD_P
```

```
-----
```

```
PROFESION
```

```
-----
```

```
-----
```

```
OCU1
```

```
MEDICO
```

```
-----
```

```
OCU2
```

```
ENFERMERA
```


select *from enfermedades;

```
SQL> select *from Tipo_profesion;
```

```
COD_P
```

```
-----
```

```
PROFESION
```

```
-----
```

```
--
```

```
OCU1
```

```
MEDICO
```

```
OCU2
```

```
ENFERMERA
```

```
SQL> select *from enfermedades;
```

```
COD_ENFER
```

```
-----
```

```
NOMBRE_ENFERMEDAD
```

```
-----
```

```
ENF1
```

```
CARDIOPATIA CRONICA
```

```
ENF2
```

```
HIPERTENSION
```

```
ENF3
```

```
NEUMOPATIA CRONICA
```

```
COD_ENFER
```

```
-----
```

```
NOMBRE_ENFERMEDAD
```

```
-----
```

```
ENF4
```

```
ASMA
```

```
ENF5
```

```
INSUFICIENCIA RENAL CRONICA
```

```
ENF6
```

```
HEPATOPATIA CRONICA
```

```

COD_ENFER
-----
NOMBRE_ENFERMEDAD
-----
ENF7
  TRASTORNO NEUROLOGICO CRONICO

ENF8
  INFECCIÓN POR EL VIH

ENF9
  DIABETES

COD_ENFER
-----
NOMBRE_ENFERMEDAD
-----
ENF10
  TUBERCULOSIS

ENF11
  ASPLENIA

ENF12
  NEOPLASIA MALIGNA

12 rows selected.

```

select *from provincia;

```

SQL> select *from provincia;

COD_P NOMBRE_PROVINCIA
-----
PRO1  BOCAS DEL TORO
PRO2  COCLE
PRO3  COLON
PRO4  CHIRIQUI
PRO5  DARIEN
PRO6  HERRERA
PRO7  LOS SANTOS
PRO8  PANAMA
PRO9  VERAGUAS
PRO10 PANAMA OESTE

10 rows selected.

```

select *from historialmedico;

```
SQL> select *from historialmedico;

COD_HM      FECHA_PR  RESULTADO
-----
a101        10/05/20  positivo
a102        01/06/20  positivo
a103        15/06/20  positivo
a104        29/05/20  positivo
a105        17/05/20  positivo
a106        10/05/20  positivo
a107        12/06/20  positivo
a108        26/05/20  positivo
a109        20/06/20  positivo
a110        28/06/20  positivo
a111        03/06/20  positivo

COD_HM      FECHA_PR  RESULTADO
-----
a112        04/06/20  positivo
a113        10/05/20  positivo
a114        12/06/20  positivo
a115        23/06/20  positivo

15 rows selected.
```

select *from hotel;

```
SQL> select *from hotel;
```

COD_HOTEL	NOMBRE_HOTEL		

CORREGIMIENTO		PROVINCIA	CANT_TRABAJADOR
-----			-----
1001	Royal Mansion		
Chilibre		PRO8	20
1002	Hotel Plaza		
Santiago		PRO9	15
1003	Hotel Mykonos		
Santiago		PRO9	12
COD_HOTEL	NOMBRE_HOTEL		

CORREGIMIENTO		PROVINCIA	CANT_TRABAJADOR
-----			-----
1004	Hotel Lisboa		
Calidonia		PRO8	10
1005	Hotel Latino		
Calidonia		PRO8	29
1006	Hotel Marbella		
Bella Vista		PRO8	20
COD_HOTEL	NOMBRE_HOTEL		

CORREGIMIENTO		PROVINCIA	CANT_TRABAJADOR
-----			-----
1007	Hotel Terranova		
Bella Vista		PRO8	15
1008	Residencial y Hotel Palazzo		
Calidonia		PRO8	16

```

      1009 Hotel Aramo
Alanje                                PR09                                12

      COD_HOTEL NOMBRE_HOTEL
-----
CORREGIMIENTO                PROVINCIA                CANT_TRABAJADOR
-----
      1010 Hotel Costa Inn
Calidonia                        PR08                                12

      1011 Hotel Montreal
Calidonia                        PR08                                25

      1012 Hotel Santa
Santiago                        PR09                                15

      COD_HOTEL NOMBRE_HOTEL
-----
CORREGIMIENTO                PROVINCIA                CANT_TRABAJADOR
-----
      1013 Hotel Mira Azul
Santiago                        PR09                                24

      1014 Hotel Cabu
David                            PR09                                16

      1015 Hotel Mariano
David                            PR09                                10

15 rows selected.

```

```
select * from personal_de_salud;
```

```
SQL> select * from personal_de_salud;
```

REGISTRO	NOMBRE	TELEFONO	COD_HOTEL
34568	Juan Mendoza	6245-6479	1001
8-43-4324			
OCU1			
98754	Jose Cabrera	6285-3545	1002
8-430-890			
OCU1			

REGISTRO	NOMBRE	TELEFONO	COD_HOTEL
12768	Manuel Chanapi	6462-7932	1003
8-160-565			
OCU1			
97450	Maria Escobar	6245-6479	
8-7382-43			

REGISTRO	NOMBRE	TELEFONO	COD_HOTEL

```

REGISTRO  NOMBRE
-----
CEDULA          TELEFONO
-----
OCUPACION          COD_HOTEL
-----
OCU1              1011

88823    Pedro Sandoval
1-354-6688    6232-5455
OCU1              1012

47715    Helen Cardenas

REGISTRO  NOMBRE
-----
CEDULA          TELEFONO
-----
OCUPACION          COD_HOTEL
-----
1-390-607    6832-1235
OCU2              1013

95122    Martin Vengara
6-948-3838    6312-3490
OCU1              1014

REGISTRO  NOMBRE
-----
CEDULA          TELEFONO
-----
OCUPACION          COD_HOTEL
-----
54311    Margarita Arjona
6-894-607    6423-7943
OCU2              1015

15 rows selected.

```

```
select * from historial_personalsalud;
```

```
SQL> select * from historial_personalsalud;
```

COD_HM	REGIST	FECHA
--------	--------	-------

DETALLE

a101	34568	30/11/20
No presenta fiebre, ni dolor de garganta.		

a102	98754	30/11/20
Dolor de garganta.		

a103	12768	30/11/20
Presenta mucho de dolor de cabeza.		

COD_HM	REGIST	FECHA
--------	--------	-------

DETALLE

a104	97450	30/11/20
Presenta dolores estomacales.		

a105	45387	30/11/20
Falta de apetito y dolores musculares.		

a106	98433	30/11/20
No presenta fiebre.		

COD_HM	REGIST	FECHA
--------	--------	-------

DETALLE

a107	47765	30/11/20
Dolor de cabeza fuerte.		

a108	45667	30/11/20
------	-------	----------


```

Dolor de cabeza fuerte.

a108      45667      30/11/20
Presenta Fiebre alta, dolor de garganta.

a109      45777      30/11/20
Presenta mareos, vomito y cansancio.


COD_HM      REGIST      FECHA
-----
DETALLE
-----
--
a110      47725      30/11/20
Presenta mareos.

a111      35449      30/11/20
Presenta vomito, dolores musculares.

a112      88823      30/11/20
Presenta glandulas inflamadas.


COD_HM      REGIST      FECHA
-----
DETALLE
-----
--
a113      47715      30/11/20
Presenta mala circulacion.

a114      95122      30/11/20
Perdida de apetito.

a115      54311      30/11/20
Presenta migraña.

15 rows selected.

```

7. Conclusiones

En la actualidad las Bases de Datos juegan un papel muy importante en la mayoría de las áreas porque estas permiten almacenar grandes volúmenes de datos acerca de la empresa o instituciones, los cuales son percibidos a través de los usuarios, de la misma manera la información obtenida de los datos almacenados debe estar en una forma que sirva para administrar, planear, controlar y tomar decisiones dentro de una organización. La necesidad de mantener un control de información aumenta cada día más, la pandemia ha golpeado a muchos, y por ende las instituciones de salud velando por el bienestar público crea esta medida de Hoteles Hospitales, donde he investigado como funciona esto y aun los formularios que los pacientes tienen que llenar se realiza de forma manual, creando un retraso para la toma de decisiones ante esta crisis de salud mundial. Aprender a diseñar e implementar una base de datos me es muy interesante porque me puedo plantear problemas de la vida diaria o de empresas y puedo crear la solución a estos.

Gómez, Andreina

Cabe destacar que el uso de las bases de datos en la actualidad, son unas de las formas más en la labor cotidiana. El hecho de poner en práctica cada una de las lecciones impartidas, demuestra la importancia y usabilidad de los sistemas de gestión de Bases de datos. Esta nos facilita la manipulación de la información, nos permiten realizar algunas consultas detalladas de esta información y nos permiten tomar decisiones precisas. Puede ser ejecutadas y tomadas en cuenta, desde un simple ejemplo como la creación de una base de datos para una despensa de una casa, para llevar el control de los enseres de esta, hasta algo tan recién y actual como lo es la pandemia de este nuevo virus. Utilizando más detalladamente este último, las bases de datos, en este caso, son utilizadas para ver tanto el recorrido e involucrados como la información de los involucrados con este virus, es decir, los pacientes, ya que se almacena todo lo respecto a él, como a las personas con quien tuvo contacto.

Arosemena, Stefanie

Por medio de este proyecto puedo concluir que el planificar, analizar los requisitos, programar e implementar una base de datos requiere de un gran esfuerzo y dedicación de tiempo. Bajo este proyecto pude entender que si no planificamos todo de manera correcta desde la base no importa las grandes habilidades de programación que uno tenga uno no puede hacer milagros. Cada paso a la hora de crear una base de datos requiere su tiempo de análisis y estudio. A la hora de que nosotros hagamos o trabajemos con una base de datos debemos de comprender todas las funciones tanto de la empresa como de la estructura de la base de datos para poder aplicar o crear sistemas más robustos, funcionales y óptimos que satisfagan las necesidades de la empresa. Por medio de este proyecto pude comprender esto al tener que pensar y analizar cómo funcionaba un hospital y de qué manera una base de datos ayudaría en la administración de esta.

Gonzales, Emanol

A través de este proyecto pude comprender el grado de complejidad y planeación que hay que utilizar a la hora de crear una base de datos e implementarla de manera completa a un entorno u aplicación. Comúnmente las personas solo vemos las cosas del lado del usuario final y pensamos que toda la estructura de la base de datos es algo sencillo como simplemente guardar los datos en un archivo y ya pero gracias a este proyecto he podido ver como se estructura de datos, como se organizan y lo más importante como se relacionan unos con otros, este detalle es fundamental para comprender como funcionan las bases de datos. Otra cosa que puede entender a través de este proyecto es forma o la lógica en que nosotros aplicamos los elementos (Trigger, procesos, secuencias, etc...) la manera de como nosotros utilizamos estos en nuestras bases de datos refleja mucho la manera en cómo nosotros organizamos la información en nuestras mentes.

Gonzales, Jose

8. Referencias Bibliográficas

- Organización Panamericana de la Salud. (Versión 2020, abril 5.). Consideraciones para el uso de hoteles durante la pandemia de la COVID-19. 2020 julio., de ORGANIZACIÓN PANAMERICANA DE LA SALUD OPS/OMS | www.paho.org Sitio web:
<https://www.paho.org/es/documentos/consideraciones-para-uso-hoteles-durante-pandemia-covid-19>
- Organización Mundial de la Salud. (abril 8, 2020). Plataforma clínica mundial sobre la COVID-19 NUEVO CORONAVIRUS (COVID-19) – VERSIÓN RÁPIDA. julio, 2020, de Organización Mundial de la Salud Sitio web:
https://apps.who.int/iris/bitstream/handle/10665/331793/WHO-2019-nCoVClinical_CRF-2020.3-spa.pdf
- LOOR, C. (27 de 10 de 2014). SlideShera. Obtenido de <https://es.slideshare.net/CesarLoor/cesar-loor>