PROYECTO CLÍNICA GINECOLÓGICA INGENIERÍA DEL SOFTWARE Y SISTEMAS DE INFORMACIÓN

Proyecto Realizado por: Javier Herraiz Olivas (IS3)

ÍNDICE

1. Introducción del problema

1.1 ¿Qué es CareMujer?

CareMujer es una clínica privada de ginecología ubicada en Sevilla, la cual ofrece servicios de ginecología general, seguimiento del embarazo, y fecundación, en la cual se especializan, teniendo una de las mejores instalaciones con material innovador. También consta de un grupo médico altamente cualificado y de alta experiencia, en cada uno de sus respectivos campos médicos.

1.2 Necesidades del cliente

El dueño de la clínica, debido a la obsolescencia de los sistemas de administración de datos en su clínica, los cuales están en su gran mayoría almacenados en papel o en numerosos excels, ha visto la necesidad de actualizar su plataforma, así como de añadir nuevas funcionalidades para dar un mejor servicio a los administradores del equipo y a sus pacientes.

Vista la necesidad del cliente, se ha pedido la creación de un sistema de control por identificación de los trabajadores, una manera de organizar óptimamente los historiales médicos, también una mejora del sistema de citas, facilitando el servicio al cliente usando el sistema mediante la página web.

1.3 Expectativas

La expectativa del cliente para el proyecto es la de tener una aplicación que cubra las siguientes funcionalidades:
Una mejor recopilación de información concerniente al funcionamiento y eficiencia de las clínicas.
Una accesibilidad y organización mejoradas respecto a las citas.
Se espera un ahorro de tiempo al semi automatizar la inclusión de datos necesarios para la gestión empresarial.

1.4 Clientes y usuarios

El cliente es el dueño de la clínica ginecológica (que a su vez también será usuario del sistema), el cual debido a las dificultades que supone la obsolescencia de algunas áreas en su clínica, ha pedido el desarrollo de ciertos sistemas.
Los usuarios los diferenciaremos en dos grupos:

Personal de la clínica:

1. Médico: Jefe y coordinador del resto del equipo.
2. Enfermero: Acceso a historiales y apoyo del médico.
3. Recepcionista: Recibir y organizar las citas.
4. Ginecólogo: Recibir por parte de la recepcionista las citas de su área.
5. Embriólogo: Recibir por parte de la recepcionista las citas de su área.
6. Limpiador: Acceso a identificación en la página.

Área de acceso para pacientes:

1. Revisión general ginecología: Acceso para pedir cita un dia puntual .
2. Fecundación: Acceso para pedir cita previa con el médico, para posteriormente acceder a la cita con la embrióloga.
3. Seguimiento del embarazo: Poder tener acceso a planificación de citas, cada ciertos meses, para un correcto seguimiento del embarazo.

2. Glosario de términos

- Almacén de historiales médicos: Lugar de la clínica donde se almacenan los historiales médicos de los pacientes, ordenados alfabéticamente y por fecha de todos los pacientes, accesibles por el enfermero.

- Agenda de citas: sitio donde estarán recogidas las horas de consultas, revisiones y atención al paciente.

- Consulta: Lugar donde un enfermero atenderá a las preguntas de los pacientes y los remitirá donde sea conveniente.

Fig. 1 Consulta médica

- Embriólogo: Médico especialista en el tratamiento de los óvulos de cara a la reproducción asistida.

- Enfermero: Ayudante del médico. Y encargado de proporcionar los historiales médicos del paciente al especialista en cuestión.

- Fecundación: Fase de la reproducción en la que el elemento reproductor masculino se une con el femenino.

- Ginecólogo: Médico especializado en el tratamiento de enfermedades del aparato reproductor femenino, así como asistencia en el tratamiento de fecundación in vitro.

- Médico jefe: Máximo responsable del personal médico.

- Recepción: Primer lugar al que accede el paciente, donde le atiende la recepcionista, corroborando que su cita es correcta y dirigiendola a la sala de espera.

- Reproducción asistida: Forma de reproducción en la que se cuenta con asistencia médica para realizar la fecundación en casos en las que existe algún tipo de problema médico que lo impide.

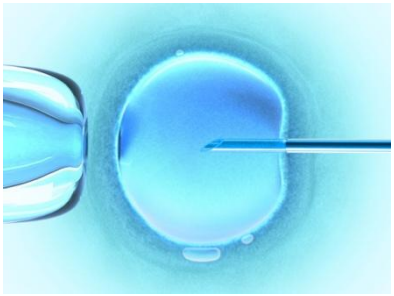


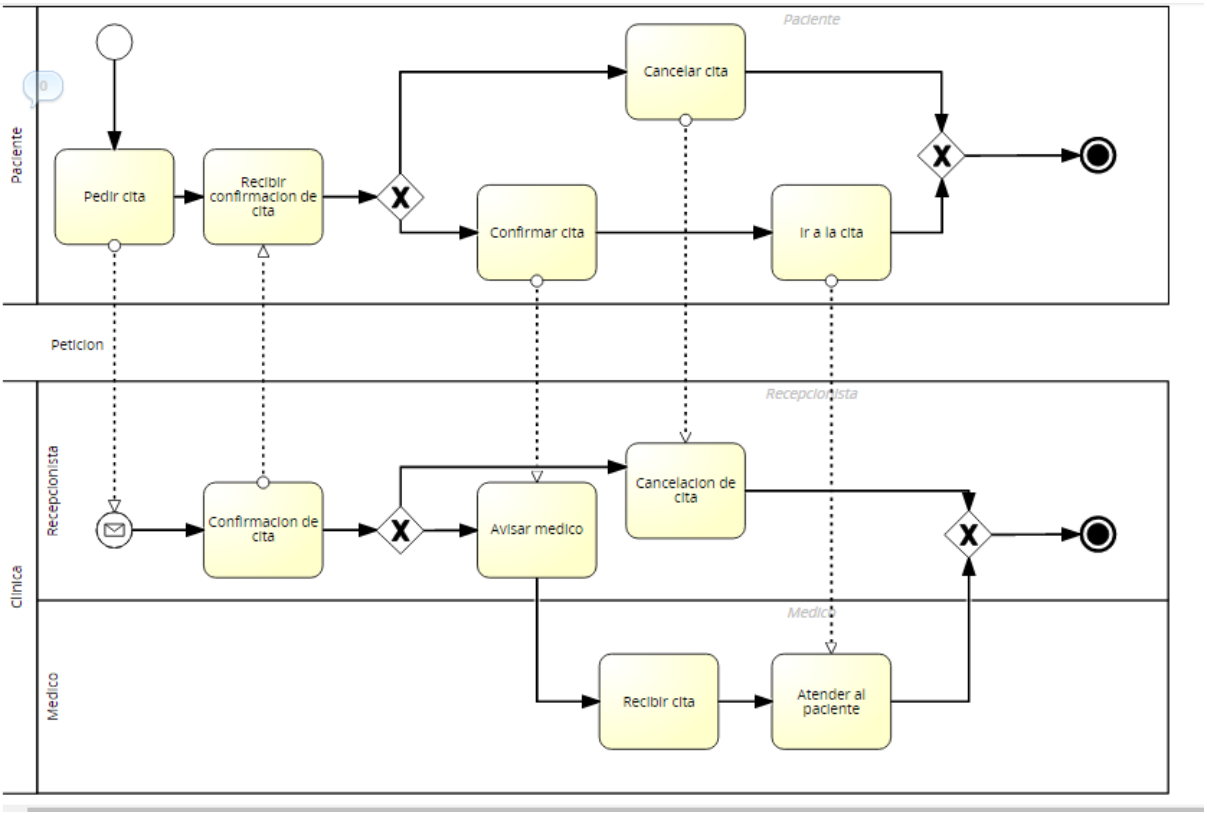
Fig.2 Óvulo fecundado

- Tumor: Crecimiento anómalo de ciertas células.

3. Modelo de negocios

3.1 Modelo para pedir una cita:

El paciente inicialmente solicita una cita, la cual llega a la recepcionista y confirma la solicitud, el paciente recibe esa confirmación y puede elegir si cancelarla o seguir con la cita.
Si cancela la cita, la recepcionista cancelará esa cita para que el hueco quede libre.
En caso de que vaya a la cita, se debe avisar al médico para que atienda al paciente.
Una vez atendido o cancelada la cita, se finaliza el proceso.



4. Visión general del sistema

Equipo de trabajo de la clínica:

R-001 Gestión de citas

Como Recepcionista:
Quiero una manera para gestionar las citas de cada paciente,
Para poder llevar un mejor control de la gestión de citas, para que haya una mejor organización

R-002 Gestión de personal

Como Médico Jefe:
Quiero una gestión de planning del personal de la clínica
Para tener una clarificada organización coherente y específica de las tareas de cada trabajador

5. Catálogo de requisitos

5.1 Requisitos de información:

RI-000 Datos sobre las clínicas

Como Médico
Quiero poder almacenar los siguientes datos de mis clínicas:
Nombre de la clínica
Dirección de la clínica

RI-001 Datos sobre los trabajadores

Como Médico
Quiero poder almacenar los siguientes datos de mis trabajadores:
Nombre del trabajador.
Apellidos del trabajador.
DNI del trabajador.
e-mail del trabajador.

RI-002 Datos de los pacientes

Como Médico
Quiero poder almacenar los siguientes datos de los pacientes:
Nombre del paciente.
Contraseña del paciente.
Apellidos del paciente.
DNI del paciente.
Número de su seguro, ya sea seguridad social o seguro privado.
Informe médico, el cual estará vacío si es su primera visita.

RI-003 Información sobre citas

Como Médico:
Quiero recibir la siguiente información sobre las citas
Hora a la que se pide la cita.
Fecha a la que se pide la cita.

RI-004 Información sobre recepcionista

Como Médico
Quiero que mi recepcionista tenga lo siguiente además de los datos del trabajador:
Un número de teléfono único en recepción, con el que poder contactar con el médico o con el proveedor.
Usuario.
Contraseña.

RI-005 Informes médicos

Como Médico
Quiero Tener un informe médico de los pacientes con los siguientes datos:
Código del informe, que será único para cada informe.
Nombre del paciente.
Apellido del paciente.
Fecha del informe.
Historial clínico, recogiendo las observaciones del médico

RI-006 Datos específicos del médico
Como Médico
Quiero que los médicos tengan los siguientes datos específicos:
Especialidad médica.
Número máximo de citas.
Usuario.
Contraseña.

5.2 Reglas de negocio:

RN-000 Máximo de citas

Como Médico Jefe
Quiero que un médico no pueda exceder el máximo de citas asignadas
Para no desatender a ningún paciente

RN-001 Citas cada 20 minutos

Como Médico Jefe
Quiero que las citas se asignen en intervalos de 20 min
Para tener tiempo suficiente al tratar el problema del cliente

RN-002 Horario de la clínica

Como Médico Jefe
Quiero que las citas estén comprendidas en el horario 9:00 a 13:00– 16:00 a 19:00
Para respetar el horario de apertura de la clínica

RN-003 Cita idéntica

Como Médico Jefe
Quiero que no pueda haber más de una cita con la misma fecha, hora y médico
Para que se pueda atender todas las citas correctamente

RN-004 Email trabajador

Como Médico Jefe
Quiero que el email asociado a todo trabajador termine con “@caremujer.es”
Para tener un email exclusivo para la clínica

RN-005 Citas de Lunes a Viernes

Como Médico Jefe
Quiero que solamente se puedan pedir citas de Lunes a Viernes
Para evitar citas en fin de semana

5.3 Requisitos funcionales:

RF-001 Lista de Trabajadores por Clínica

Como Médico Jefe
Quiero que el sistema permita obtener un informe de una lista con todos los trabajadores y la clínica en la que trabajan
Para planificación sobre el personal

RF-002 Lista de Pacientes por Clínica

Como Médico Jefe
Quiero que el sistema permita obtener un informe de una lista con todos los pacientes de que tienen cita en una clínica determinada
Para estudios sobre los pacientes de cada clínica

RF-003 Lista de Citas Disponibles por días para una Clínica determinada

Como Paciente
Quiero que el sistema permita obtener un informe de una lista con las horas de todas las citas disponibles de un día en la clínica especificada
Para elegir la cita más conveniente

RF-004 Lista de Citas Ocupadas por días para una Clínica determinada

Como Médico Jefe
Quiero que el sistema permita obtener un informe de una lista con todas las horas de las citas ocupadas de un día para una clínica determinada.
Para planificación sobre las citas ocupadas.

RF-005 Lista de Citas por Médico y por día

Como Médico
Quiero que el sistema permita obtener un informe de una lista con todas las citas de un médico para un día determinado
Para planificación personal de cada médico sobre sus propias citas

RF-006 Cuenta de Médicos por Clínica

Como Médico Jefe
Quiero que el sistema permita obtener el número de médicos que trabajan en cada clínica **Para** organización del personal médico

RF-007 Lista de Médicos por especialidad

Como Médico Jefe
Quiero que el sistema permita obtener un informe de una lista con los médicos de una clínica específica agrupados por especialidad.
Para organizar mejor casos concretos de pacientes

RF-008 Historiales Clínicos

Como Médico Jefe
Quiero que el sistema permita obtener los historiales clínicos de un paciente accesibles por el mismo paciente y los médicos asociados
Para obtener rápidamente la historia de los pacientes

RF-009 Número Seguro de Paciente

Como Médico Jefe
Quiero que el sistema permita obtener el número de seguro de un determinado cliente
Para obtener información sobre el costo de su tratamiento

RF-010 Cita más tarde de un Médico

Como Médico Jefe
Quiero que el sistema permita obtener la cita ocupada con hora más tarde de un médico para un día determinado
Para obtener la última cita de ese médico en ese día

RF-011 Informe de Médicos supervisando historiales

Como Médico Jefe
Quiero que el sistema permita obtener un informe que muestre que Médico está supervisando el Historial de cada Paciente
Para llevar un control de qué médicos están estudiando el caso de cada paciente

5.4 Requisitos no funcionales:

RNF-000 Privacidad de los clientes

Como Médico jefe
Quiero mantener la privacidad de las citas de los clientes
Para cumplir la Ley de Protección de Datos

RNF-001 Control de Usuarios

Como Médico jefe
Quiero acceso privilegiado para personal de la clínica
Para acceder únicamente éstos a plannings

RNF-002 Concurrencia del sistema

Como Médico jefe
Quiero que el sistema de citas soporte una cantidad grande de usuarios a la vez
Para que el servidor no se caiga

5.5 Mapa de historias de usuario:

MAPA DE HISTORIAS DE USUARIO						
GESTIÓN DE CITAS						
RI-000 Datos sobre las clínicas	**RI-001 Datos sobre los trabajadores**	**RI-002 Datos de los pacientes**	**RI-003 Información sobre citas**	**RI-004 Información sobre recepcionista**	**RI-005 Informes Médicos**	**RI-006 Datos Específicos Del Médico**
-	**RN-004 Email Trabajador**	-	**RN-001 Citas cada 20 minutos** **RN-002 Horario de la Clínica** **RN-003 Cita idéntica**	-	-	**RN-000 Máximo de Citas**
RF-003 Lista de Citas Disponibles por días para una Clínica determinada **RF-004 Lista de Citas Ocupadas por días para una Clínica determinada** **RF-006 Cuenta de Médicos por Clínica**	**RF-001 Lista de Trabajadores por Clínica**	**RF-002 Lista de Pacientes por Clínica** **RF-008 Historiales Clínicos** **RF-009 Número Seguro de Paciente**	**RF-003 Lista de Citas Disponibles por días para una Clínica determinada** **RF-004 Lista de Citas Ocupadas por días para una Clínica determinada** **RF-005 Lista de Citas por Médico y por día**	**RF-001 Asignación/ Cancelación de citas única**	**RF-008 Historiales Clínicos** **RF-011 Informe de Médicos supervisando historiales**	**RF-005 Lista de Citas por Médico y por día** **RF-006 Cuenta de Médicos por Clínica** **RF-007 Lista de Médicos por Especialidad** **RF-008 Historiales Clínicos** **RF-010 Cita más tarde de un Médico** **RF-011 Informe de Médicos supervisando historiales**
RNF-000 Privacidad de los clientes			**RNF-001 Control de Usuarios**		**RNF-002 Concurrencia del sistema**	

6. Pruebas de aceptación

RN-000 Máximo de citas:

- Se intenta añadir citas a un médico con el número máximo de citas ocupadas salta mensaje de error.
- Al realizar una nueva asignación de cita o cancelación a un médico se actualiza el planning del médico en cuestión

RN-001 Citas cada 20 minutos:

- Las citas disponibles almacenadas para una fecha tienen una duración de 20 min
- No es posible pedir/insertar una cita de mayor duración

RN-002 Horario de la clínica:

- Las citas para una fecha determinada empiezan a partir de las 9:00 y acaban a las 19:00
- No es posible pedir una cita fuera de ese intervalo

RN-003 Cita idéntica:

- Una vez asignada una cita en una fecha y hora para un médico determinado esta se elimina del sistema, impidiendo la posibilidad de escogerla de nuevo por otro paciente.

RN-005 Email Trabajadores:

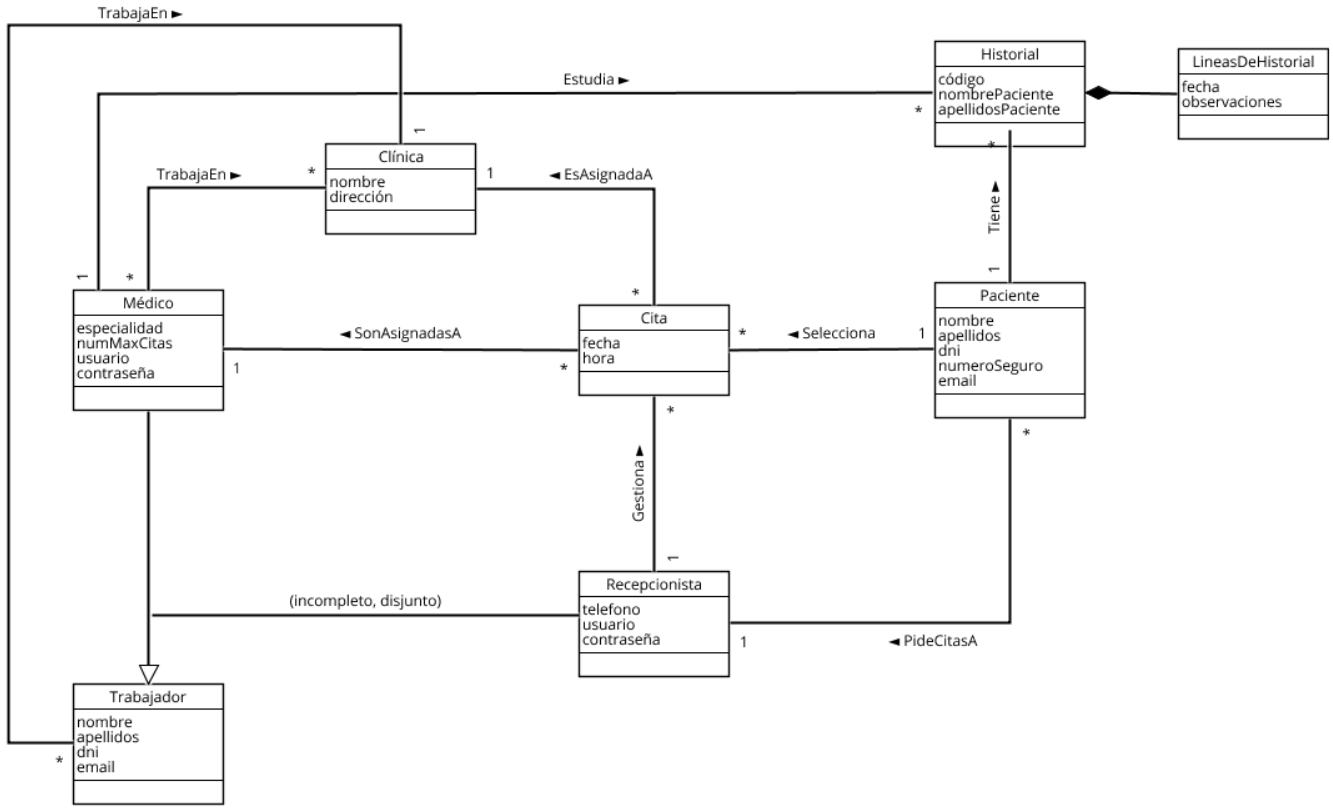
- Se intenta registrar un nuevo email para un trabajador que no termine según el patrón salta código de error

RN-006 Citas entre semana

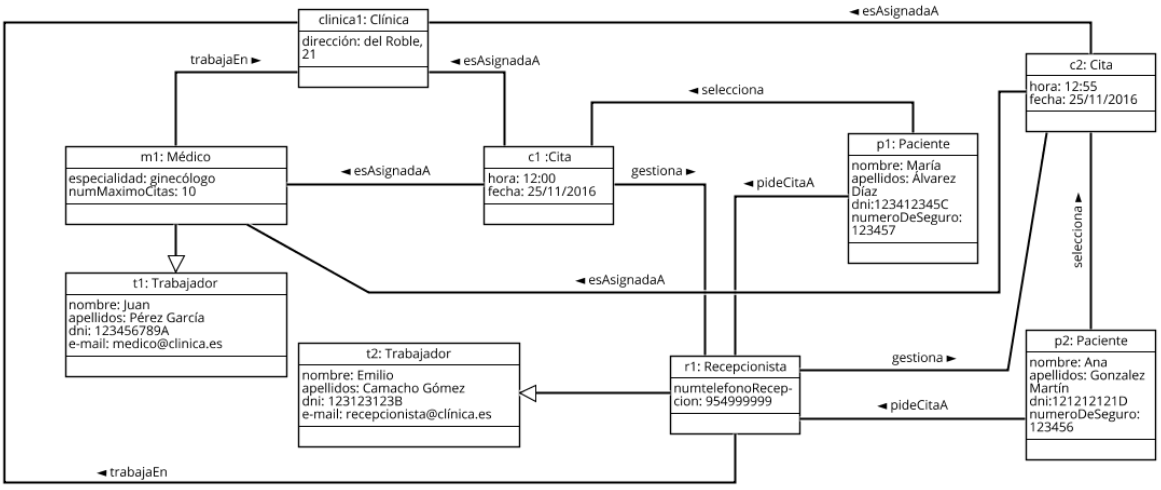
- Las citas deben insertarse en días entre semana.
- Si se intenta insertar una fecha que coincide con un día en fin de semana salta código de error y anula la inserción.

7. Modelo conceptual

UML Gestión de citas:



Escenario de pruebas gestión de citas



8. Matrices de Trazabilidad

8.1 Matriz de trazabilidad de Clases -> Requisitos de Información:

RI / CLASES	RI-000 Datos sobre las clínicas	RI-001 Datos sobre los trabajadores	RI-002 Datos de los pacientes	RI-003 Información sobre citas	RI-004 Información sobre recepcionista	RI-005 Informes médicos	RI-006 Datos específicos del médico
Clínica	X						
Médico		X				X	X
Cita				X			
Paciente			X			X	
Recepcionista		X			X		
Trabajador		X					
Historial						X	
LíneaDeHistorial						X	

8.2 Matriz de trazabilidad de Clases -> Reglas de Negocio:

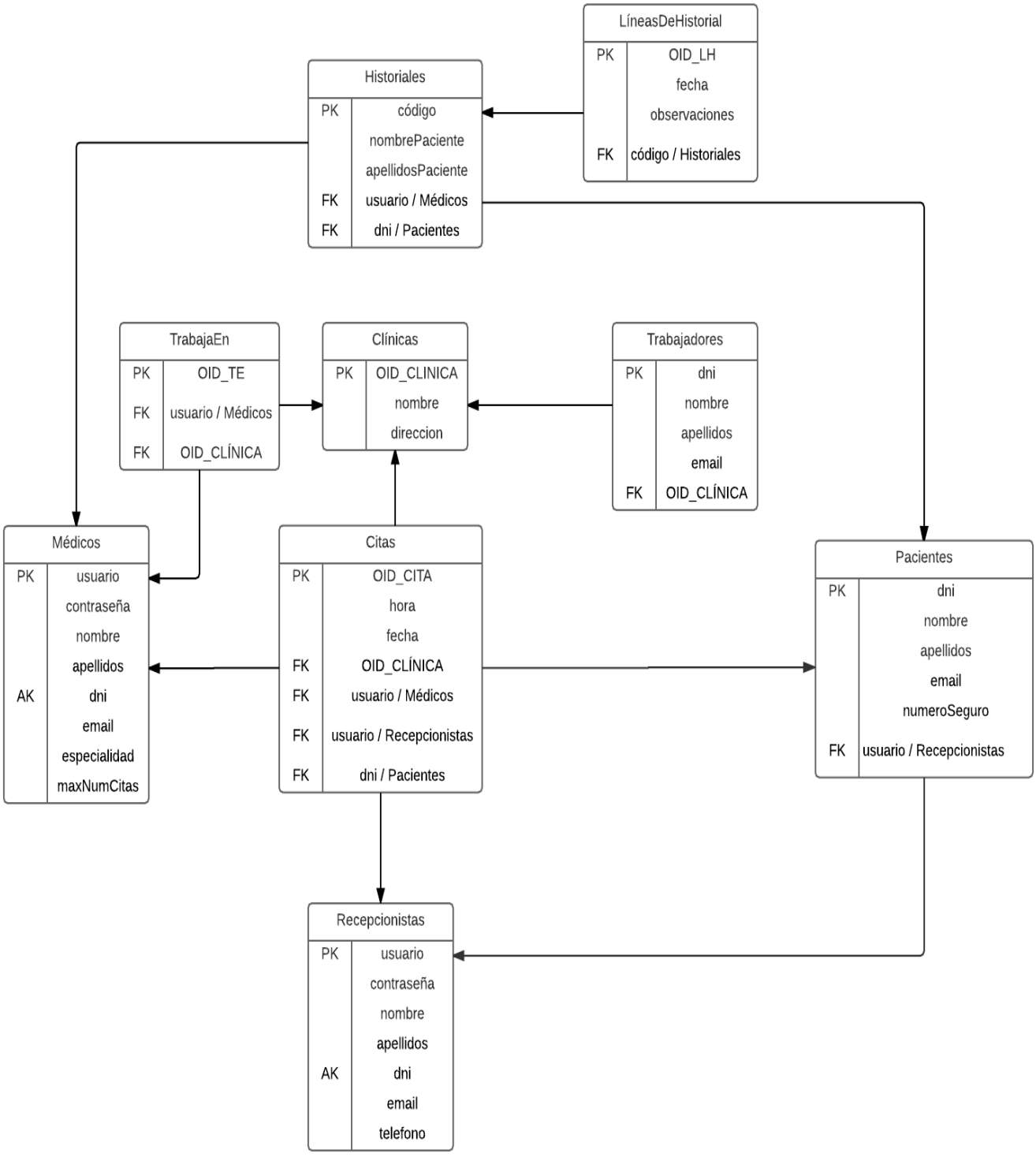
RN / CLASES	RN-000 Máximo de citas	RN-001 Citas cada 20 minutos	RN-002 Horario de la clínica	RN-003 Cita idéntica	RN-004 Email Trabajadores
Clínica		X	X		
Médico	X			X	X
Cita	X	X	X	X	
Paciente				X	
Recepcionista					X
Trabajador					X
Historial					
LíneaDeHistorial					

8.3 Matriz de trazabilidad de Clases -> Requisitos Funcionales:

RF / CLASES	RF-001 Lista de Trabajadores Por Clínica	RF-002 Lista de Pacientes Por Clínica	RF-003 Lista de Citas Disponibles por días para una Clínica determinada	RF-004 Lista de Citas Ocupadas por días para una Clínica determinada	RF-005 Lista de Citas por Médico y por día	RF-006 Cuenta de Médicos por Clínica	RF-007 Lista de Médicos por Especialidad	RF-008 Historiales Médicos	RF-009 Número de seguro de Paciente	RF-010 Cita más tarde de Médico	RF-011 Informe de Médicos supervisando historiales
Clínica	X	X	X	X		X					
Médico	X				X	X	X	X		X	X
Cita			X	X	X				X	X	
Paciente		X		X	X			X			X
Recepcionista	X										
Trabajador	X										
Historial								X			X
LíneaDe Historial								X			X

9. Modelo Relacional

9.1 Modelo Relacional para la Gestión de Citas:



9.2 Justificación de la estrategia de transformación de jerarquías:

Transformación de entidades:

Al realizar la transformación del modelo conceptual al modelo relacional, para cada entidad se ha trasformado en una relación cuyo nombre ha pasado a estar en plural, con sus correspondientes atributos necesarios.

En aquellas relaciones donde hubiese atributos que sean claves semánticas sencillas, se han usado para definir las claves primarias. En el resto, se han creado unas OIDs para definir como clave primaria.

En aquellas relaciones donde hubiese atributos que sean claves semánticas y no se hayan definido como claves primarias, se han definidos claves alternativas.

Transformación de asociaciones:

En el caso de las asociaciones con cardinalidad 1:N, se ha creado un atributo en la tabla con participación [N] que será una clave ajena apuntando al atributo con clave primaria de la tabla con participación [1].

En el caso de las asociaciones con cardinalidad 1:1, se ha creado un atributo en una de las tablas que será una clave ajena apuntando al atributo con clave primaria de la otra tabla.

En el caso de las asociaciones con cardinalidad N:M, se ha creado una tabla auxiliar que relacione ambas entidades, la cual tendrán atributos que serán una clave primaria compuesta y ajenas que apuntan al atributo con clave primaria de cada tabla de cada lado de la asociación.

Transformación de clasificaciones:

El modelo conceptual cuenta con entidades que son heredadas de otras entidades jerárquicamente superiores:
Es el caso de *Trabajador,* que hereda a *Médico* y *Recepcionista,* se trata de una clasificación disjunta e incompleta, por lo que la mejor opción es crear 3 tablas: *Trabajadores, Médicos* y *Recepcionistas.*

Normalización del modelo relacional:

Con todas las transformaciones anteriores se obtiene el modelo relacional en 3FN, justificación:

- Está en 1FN porque los atributos son monoevaluados.
- Está en 2FN porque está en 1FN y todos los atributos que no forman parte de ninguna clave candidata son completamente dependientes de las claves candidatas de la relación.
- Está en 3FN porque está en 2FN y no existen dependencias transitivas.

10. Modelo tecnológico

CREACIONES DE TABLAS, RESTRICCIONES, SECUENCIAS Y TRIGGERS ASOCIADOS A SECUENCIAS

```
--CREACIÓN DE TABLAS
CREATE TABLE CLINICAS
  (OID_CLINICA NUMBER NOT NULL,
  NOMBRE_CLINICA VARCHAR2(100)NOT NULL,
  DIRECCION VARCHAR2(100) NOT NULL,
  PRIMARY KEY (OID_CLINICA));

CREATE TABLE RECEPCIONISTAS
  (USUARIO VARCHAR2(75) PRIMARY KEY,
  CONTRASEÑA VARCHAR2(75) NOT NULL,
  NOMBRE VARCHAR2(40) NOT NULL,
  APELLIDOS VARCHAR2(40) NOT NULL,
  DNI CHAR(9) NOT NULL UNIQUE,
  EMAIL VARCHAR2(40) NOT NULL,
  TELEFONO VARCHAR2(9) NOT NULL);

CREATE TABLE TRABAJADORES
  (DNI CHAR(9) PRIMARY KEY,
  NOMBRE VARCHAR2(30) NOT NULL,
  APELLIDOS VARCHAR2(40) NOT NULL,
  EMAIL VARCHAR2(40) NOT NULL ,
  OID_CLINICA NUMBER NOT NULL,
  FOREIGN KEY(OID_CLINICA) REFERENCES CLINICAS(OID_CLINICA));

CREATE TABLE MEDICOS
  (USUARIO_MEDICO VARCHAR2(75) PRIMARY KEY,
  CONTRASEÑA VARCHAR2(75) NOT NULL,
  NOMBRE VARCHAR2(40) NOT NULL,
  APELLIDOS VARCHAR2(40) NOT NULL,
  DNI CHAR(9) NOT NULL UNIQUE,
  EMAIL VARCHAR2(40) NOT NULL,
  ESPECIALIDAD VARCHAR2(40) NOT NULL,
  MAXNUMCITAS NUMBER(3) NOT NULL);

CREATE TABLE TRABAJAEN
  (OID_TE NUMBER(2) PRIMARY KEY,
  USUARIO_MEDICO VARCHAR2(75) NOT NULL,
  OID_CLINICA NUMBER NOT NULL,
  FOREIGN KEY(USUARIO_MEDICO) REFERENCES MEDICOS(USUARIO_MEDICO),
  FOREIGN KEY(OID_CLINICA) REFERENCES CLINICAS(OID_CLINICA));
```

CREATE TABLE PACIENTES
(DNI CHAR(9) PRIMARY KEY,
NOMBRE VARCHAR2(40) NOT NULL,
CONTRASEÑA VARCHAR(40) NOT NULL,
APELLIDOS VARCHAR2(40) NOT NULL,
EMAIL VARCHAR2(40) NOT NULL,
NUMEROSEGURO NUMBER(6),
USUARIO_RECEPCIONISTA VARCHAR2(75)NOT NULL,
FOREIGN KEY(USUARIO_RECEPCIONISTA) REFERENCES RECEPCIONISTAS(USUARIO));

CREATE TABLE CITAS
(OID_CITA NUMBER PRIMARY KEY,
HORA VARCHAR2(5) NOT NULL,
FECHA DATE NOT NULL,
OID_CLINICA NUMBER,
USUARIO_MEDICO VARCHAR2(75) NOT NULL,
USUARIO_RECEPCIONISTA VARCHAR2(75),
DNI_PACIENTE CHAR(9),
FOREIGN KEY(OID_CLINICA) REFERENCES CLINICAS(OID_CLINICA),
FOREIGN KEY(USUARIO_MEDICO) REFERENCES MEDICOS(USUARIO_MEDICO),
FOREIGN KEY(USUARIO_RECEPCIONISTA) REFERENCES RECEPCIONISTAS(USUARIO),
FOREIGN KEY(DNI_PACIENTE) REFERENCES PACIENTES(DNI));

CREATE TABLE HISTORIALES
(CODIGO VARCHAR2(6) PRIMARY KEY,
NOMBREPACIENTE VARCHAR2(40) NOT NULL,
APELLIDOSPACIENTE VARCHAR2(60) NOT NULL,
USUARIO_MEDICO VARCHAR2(75),
DNI_PACIENTE CHAR(9),
FOREIGN KEY (USUARIO_MEDICO) REFERENCES MEDICOS(USUARIO_MEDICO),
FOREIGN KEY (DNI_PACIENTE) REFERENCES PACIENTES(DNI));

CREATE TABLE LINEASDEHISTORIAL
(OID_LH NUMBER PRIMARY KEY,
FECHA DATE NOT NULL,
OBSERVACIONES VARCHAR2(400) NOT NULL,
CODIGO_HISTORIAL VARCHAR2(6) NOT NULL,
FOREIGN KEY (CODIGO_HISTORIAL) REFERENCES HISTORIALES(CODIGO) ON DELETE CASCADE);

--RESTRICCIONES DE TABLAS

ALTER TABLE RECEPCIONISTAS ADD CONSTRAINT CK_DNI_RECEPCIONISTAS CHECK (REGEXP_LIKE(DNI, '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]'));

ALTER TABLE TRABAJADORES ADD CONSTRAINT CK_DNI_TRABAJADORES CHECK (REGEXP_LIKE(DNI, '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]'));

ALTER TABLE MEDICOS ADD CONSTRAINT CK_DNI_MEDICOS CHECK (REGEXP_LIKE(DNI, '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]'));

ALTER TABLE PACIENTES ADD CONSTRAINT CK_DNI_PACIENTES CHECK (REGEXP_LIKE(DNI, '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]'));

ALTER TABLE CITAS ADD CONSTRAINT CK_DNI_CITAS CHECK (REGEXP_LIKE(DNI_PACIENTE,'[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]')OR DNI_PACIENTE IS NULL);

ALTER TABLE HISTORIALES ADD CONSTRAINT CK_DNI_HISTORIALES CHECK (REGEXP_LIKE(DNI_PACIENTE, '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]'));

ALTER TABLE PACIENTES ADD CONSTRAINT CK_NUMEROSEGURO_PACIENTES CHECK (REGEXP_LIKE(NUMEROSEGURO, '[0-9][0-9][0-9][0-9][0-9][0-9]'));

ALTER TABLE HISTORIALES ADD CONSTRAINT CK_CÓDIGO_HISTORIALES CHECK (REGEXP_LIKE(CODIGO, '[0-9][0-9][0-9][0-9][0-9][0-9]'));

ALTER TABLE LINEASDEHISTORIAL ADD CONSTRAINT CK_CÓDIGO_LINEASDEHISTORIAL CHECK (REGEXP_LIKE(CODIGO_HISTORIAL, '[0-9][0-9][0-9][0-9][0-9][0-9]'));

ALTER TABLE MEDICOS ADD CONSTRAINT CK_ESPECIALIDAD_MÉDICOS CHECK (ESPECIALIDAD IN ('ONCÓLOGO', 'GINECÓLOGO', 'EMBRIÓLOGO'));
ALTER TABLE CITAS ADD CONSTRAINT CK_HORA_CITAS CHECK (REGEXP_LIKE(Hora, '[0-2][0-9]:[0-5][0-9]'));

--SECUENCIAS

CREATE SEQUENCE SEC_CLINICAS INCREMENT BY 1 START WITH 1 MAXVALUE 99999;

CREATE SEQUENCE SEC_TRABAJAEN INCREMENT BY 1 START WITH 1 MAXVALUE 99999;

CREATE SEQUENCE SEC_CITAS INCREMENT BY 1 START WITH 1 MAXVALUE 99999;

CREATE SEQUENCE SEC_LINEASDEHISTORIAL INCREMENT BY 1 START WITH 1 MAXVALUE 99999;

```
--TRIGGERS ASOCIADOS A SECUENCIAS

CREATE OR REPLACE TRIGGER TR_SEC_CLINICAS
  BEFORE INSERT ON CLINICAS
  FOR EACH ROW
  BEGIN
    SELECT SEC_CLINICAS.NEXTVAL INTO :NEW.OID_CLINICA FROM DUAL;
  END;
/

CREATE OR REPLACE TRIGGER TR_SEC_TRABAJAEN
  BEFORE INSERT ON TRABAJAEN
  FOR EACH ROW
  BEGIN
    SELECT SEC_TRABAJAEN.NEXTVAL INTO :NEW.OID_TE FROM DUAL;
  END;
/


CREATE OR REPLACE TRIGGER TR_SEC_CITAS
  BEFORE INSERT ON CITAS
  FOR EACH ROW
  BEGIN
    SELECT SEC_CITAS.NEXTVAL INTO :NEW.OID_CITA FROM DUAL;
  END;
/

CREATE OR REPLACE TRIGGER TR_SEC_LINEASDEHISTORIAL
  BEFORE INSERT ON LINEASDEHISTORIAL
  FOR EACH ROW
  BEGIN
    SELECT SEC_LINEASDEHISTORIAL.NEXTVAL INTO :NEW.OID_LH FROM DUAL;
  END;
/
```

FUNCIONES Y PROCEDIMIENTOS

```
--PROCEDIMIENTOS Y FUNCIONES ASOCIADAS A LOS REQUISITOS FUNCIONALES
--RF1
CREATE OR REPLACE PROCEDURE PR_TRABAJADORES_CLINICA
IS
  CURSOR C IS
    SELECT * FROM TRABAJADORES NATURAL JOIN CLINICAS;
V_TRABAJADORES C%ROWTYPE;
BEGIN
  OPEN C;
  FETCH C INTO V_TRABAJADORES;
  DBMS_OUTPUT.PUT_LINE( 'TRABAJADORES AGRUPADOS POR CLÍNICA EN LA QUE
TRABAJAN:');
  DBMS_OUTPUT.PUT_LINE( RPAD('DNI:',15) || RPAD('NOMBRE:',25) || RPAD('APELLIDOS',35) ||
RPAD('EMAIL',25) || RPAD('OID_CLINICA',15) || RPAD('NOMBRE_CLINICA',50) ||
RPAD('DIRECCION',70));
  WHILE C%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE( RPAD(V_TRABAJADORES.DNI,15) ||
RPAD(V_TRABAJADORES.NOMBRE,25) || RPAD(V_TRABAJADORES.APELLIDOS,35) ||
RPAD(V_TRABAJADORES.EMAIL,25) || RPAD(V_TRABAJADORES.OID_CLINICA,15) ||
RPAD(V_TRABAJADORES.NOMBRE_CLINICA,50) || RPAD(V_TRABAJADORES.DIRECCION,100));
  FETCH C INTO V_TRABAJADORES;
  END LOOP;
  CLOSE C;
END;
/

--RF2
CREATE OR REPLACE PROCEDURE PR_PACIENTES_CLINICA (V_OID_CLINICA IN
CITAS.OID_CLINICA%TYPE)
IS
  CURSOR C IS
     SELECT * FROM PACIENTES WHERE DNI IN (SELECT DNI_PACIENTE FROM CITAS WHERE
OID_CLINICA = V_OID_CLINICA);
V_PACIENTES PACIENTES%ROWTYPE;
BEGIN
  OPEN C;
  FETCH C INTO V_PACIENTES;
  DBMS_OUTPUT.PUT_LINE('PACIENTES CON CITA EN CLÍNICA DETERMINADA:');
  DBMS_OUTPUT.PUT_LINE(RPAD('DNI:',15) || RPAD('NOMBRE:',20) || RPAD('APELLIDOS:',30) ||
RPAD('EMAIL:',30) || RPAD('NUMEROSEGURO:',15) );
  WHILE C%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(RPAD(V_PACIENTES.DNI,15) || RPAD(V_PACIENTES.NOMBRE,20) ||
RPAD(V_PACIENTES.APELLIDOS,30) || RPAD(V_PACIENTES.EMAIL,30) ||
RPAD(V_PACIENTES.NUMEROSEGURO,10) );
  FETCH C INTO V_PACIENTES;
  END LOOP;
  CLOSE C;
END;
/
```

```
--RF3
CREATE OR REPLACE PROCEDURE PR_CITAS_DISPONIBLES (V_FECHA IN
CITAS.FECHA%TYPE, V_OID_CLINICA IN CITAS.OID_CLINICA%TYPE, V_USUARIO_MEDICO IN
CITAS.USUARIO_MEDICO%TYPE)
IS
  CURSOR C IS
    SELECT HORA FROM CITAS WHERE FECHA=V_FECHA AND OID_CLINICA = V_OID_CLINICA
AND USUARIO_MEDICO = V_USUARIO_MEDICO AND DNI_PACIENTE IS NULL;
V_HORA CITAS.HORA%TYPE;
BEGIN
  OPEN C;
  FETCH C INTO V_HORA;
  DBMS_OUTPUT.PUT_LINE(RPAD('HORA:',20));
  WHILE C%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(RPAD(V_HORA,20));
  FETCH C INTO V_HORA;
  END LOOP;
  CLOSE C;
END;
/

--RF4
CREATE OR REPLACE PROCEDURE PR_CITAS_NO_DISPONIBLES (V_FECHA IN
CITAS.FECHA%TYPE, V_OID_CLINICA IN CITAS.OID_CLINICA%TYPE)
IS
  CURSOR C IS
    SELECT * FROM CITAS WHERE FECHA=V_FECHA  AND OID_CLINICA = V_OID_CLINICA AND
DNI_PACIENTE IS NOT NULL;
V_CITAS CITAS%ROWTYPE;
BEGIN
  OPEN C;
  FETCH C INTO V_CITAS;

DBMS_OUTPUT.PUT_LINE(RPAD('HORA:',20)||RPAD('DNI_PACIENTE:',20)||RPAD('USUARIO_MEDI
CO:',20));
  WHILE C%FOUND LOOP

DBMS_OUTPUT.PUT_LINE(RPAD(V_CITAS.HORA,20)||RPAD(V_CITAS.DNI_PACIENTE,20)||RPAD(
V_CITAS.USUARIO_MEDICO,20));
  FETCH C INTO V_CITAS;
  END LOOP;
  CLOSE C;
END;
/
```

```
--RF5
CREATE OR REPLACE PROCEDURE PR_CITAS_MEDICO (V_USUARIO IN
CITAS.USUARIO_MEDICO%TYPE, V_FECHA IN CITAS.FECHA%TYPE)
IS
  CURSOR C IS
    SELECT HORA,DNI_PACIENTE,OID_CLINICA FROM CITAS WHERE FECHA = V_FECHA AND
USUARIO_MEDICO = V_USUARIO AND DNI_PACIENTE IS NOT NULL;
V_CITA C%ROWTYPE;
BEGIN
  OPEN C;
  FETCH C INTO V_CITA;
  DBMS_OUTPUT.PUT_LINE(RPAD('HORA:',20) || RPAD('DNI_PACIENTE:',20) ||
RPAD('OID_CLINICA:',20));
  WHILE C%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(RPAD(V_CITA.HORA,20) || RPAD(V_CITA.DNI_PACIENTE,20) ||
RPAD(V_CITA.OID_CLINICA,20));
  FETCH C INTO V_CITA;
  END LOOP;
  CLOSE C;
END;
/

--RF6
CREATE OR REPLACE PROCEDURE PR_MEDICOS_CLINICA
IS
  CURSOR C IS
    SELECT OID_CLINICA,COUNT(*) FROM TRABAJAEN GROUP BY OID_CLINICA;
V_OID_CLINICA TRABAJAEN.OID_CLINICA%TYPE;
V_COUNT NUMBER;
BEGIN
  OPEN C;
  FETCH C INTO V_OID_CLINICA,V_COUNT;
  DBMS_OUTPUT.PUT_LINE(RPAD('OID_CLINICA:',20) || RPAD('Nº MÉDICOS:',20));
  WHILE C%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(RPAD(V_OID_CLINICA,20) || RPAD(V_COUNT,20));
  FETCH C INTO V_OID_CLINICA,V_COUNT;
  END LOOP;
  CLOSE C;
END;
/
```

```
--RF7
CREATE OR REPLACE PROCEDURE PR_MEDICOS_ESPECIALIDAD (V_OID_CLINICA IN
TRABAJAEN.OID_CLINICA%TYPE)
IS
  CURSOR C IS
     SELECT USUARIO_MEDICO,NOMBRE,APELLIDOS,EMAIL,ESPECIALIDAD
    FROM MEDICOS WHERE USUARIO_MEDICO IN (SELECT USUARIO_MEDICO FROM
TRABAJAEN WHERE OID_CLINICA=V_OID_CLINICA) ;
V_USUARIO MEDICOS.USUARIO_MEDICO%TYPE;
V_NOMBRE MEDICOS.NOMBRE%TYPE;
V_APELLIDOS MEDICOS.APELLIDOS%TYPE;
V_EMAIL MEDICOS.EMAIL%TYPE;
V_ESPECIALIDAD MEDICOS.ESPECIALIDAD%TYPE;
BEGIN
 OPEN C;
 FETCH C INTO V_USUARIO,V_NOMBRE,V_APELLIDOS,V_EMAIL,V_ESPECIALIDAD;
 DBMS_OUTPUT.PUT_LINE(RPAD('USUARIO_MEDICO:',30) || RPAD('NOMBRE:',20) ||
RPAD('APELLIDOS:',30) || RPAD('EMAIL:',30) || RPAD('ESPECIALIDAD:',30));
 WHILE C%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(RPAD(V_USUARIO,30) || RPAD(V_NOMBRE,20) ||
RPAD(V_APELLIDOS,30) || RPAD(V_EMAIL,30) || RPAD(V_ESPECIALIDAD,30));
   FETCH C INTO V_USUARIO,V_NOMBRE,V_APELLIDOS,V_EMAIL,V_ESPECIALIDAD;
  END LOOP;
  CLOSE C;
END;
/

--RF8
CREATE OR REPLACE PROCEDURE PR_HISTORIALES (V_DNI_PACIENTE IN
HISTORIALES.DNI_PACIENTE%TYPE)
IS
  CURSOR C IS
    SELECT *
  FROM LINEASDEHISTORIAL
  WHERE CODIGO_HISTORIAL IN (SELECT CODIGO FROM HISTORIALES WHERE
DNI_PACIENTE = V_DNI_PACIENTE);
V_LINEASDEHISTORIAL LINEASDEHISTORIAL%ROWTYPE;
BEGIN
  OPEN C;
  FETCH C INTO V_LINEASDEHISTORIAL;
  DBMS_OUTPUT.PUT_LINE(RPAD('OID_LH:',20) || RPAD('FECHA:',20) ||
RPAD('OBSERVACIONES:',200) || RPAD('CODIGO_HISTORIAL:',20));
  WHILE C%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(RPAD(V_LINEASDEHISTORIAL.OID_LH,20) ||
RPAD(V_LINEASDEHISTORIAL.FECHA,20) ||
RPAD(V_LINEASDEHISTORIAL.OBSERVACIONES,200) ||
RPAD(V_LINEASDEHISTORIAL.CODIGO_HISTORIAL,20));
  FETCH C INTO V_LINEASDEHISTORIAL;
  END LOOP;
  CLOSE C;
END;
/
```

```
--RF9
CREATE OR REPLACE FUNCTION FN_SEGURO_PACIENTE (V_DNI_PACIENTE IN
PACIENTES.DNI%TYPE)
RETURN NUMBER
IS
V_SEGURO PACIENTES.NUMEROSEGURO%TYPE;
BEGIN
  SELECT NUMEROSEGURO INTO V_SEGURO FROM PACIENTES WHERE
DNI=V_DNI_PACIENTE;
  IF V_SEGURO = NULL THEN
    RETURN 'EL PACIENTE NO TIENE SEGURO';
  ELSE
    RETURN (V_SEGURO);
  END IF;
END;
/

--RF10
CREATE OR REPLACE FUNCTION FN_CITA_MAS_TARDE (V_USUARIO_MEDICO IN
CITAS.USUARIO_MEDICO%TYPE, V_FECHA IN CITAS.FECHA%TYPE)
RETURN VARCHAR2
IS
V_HORA CITAS.HORA%TYPE;
BEGIN
    SELECT MAX(HORA) INTO V_HORA FROM CITAS WHERE FECHA=V_FECHA AND
USUARIO_MEDICO=V_USUARIO_MEDICO AND DNI_PACIENTE IS NOT NULL;
    RETURN (V_HORA);
END;
/
--RF11
CREATE OR REPLACE PROCEDURE PR_MEDICO_SUPERVISA_HISTORIAL
IS
  CURSOR C IS
    SELECT
USUARIO_MEDICO,NOMBRE,APELLIDOS,DNI,CODIGO,NOMBREPACIENTE,APELLIDOSPACIENT
E,DNI_PACIENTE FROM MEDICOS NATURAL JOIN HISTORIALES;
  V_RES C%ROWTYPE;
BEGIN
  OPEN C;
  FETCH C INTO V_RES;
  DBMS_OUTPUT.PUT_LINE(RPAD('USUARIO_MEDICO:',20) || RPAD('NOMBRE:',20) ||
RPAD('APELLIDOS:',30) || RPAD('DNI:',20) || RPAD('NOMBREPACIENTE:',20) ||
RPAD('APELLIDOSPACIENTE:',30) || RPAD('DNI_PACIENTE:',20));
  WHILE C%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(RPAD(V_RES.USUARIO_MEDICO,20) || RPAD(V_RES.NOMBRE,20)
|| RPAD(V_RES.APELLIDOS,30) || RPAD(V_RES.DNI,20) || RPAD(V_RES.NOMBREPACIENTE,20) ||
RPAD(V_RES.APELLIDOSPACIENTE,30) || RPAD(V_RES.DNI_PACIENTE,20));
  FETCH C INTO V_RES;
  END LOOP;
  CLOSE C;
END;
/
```

TRIGGERS NO ASOCIADOS A SECUENCIAS

--TRIGGERS ASOCIADOS A REGLAS DE NEGOCIO

--RN0

```
CREATE OR REPLACE TRIGGER TR_MAXNUMCITAS
BEFORE INSERT OR UPDATE OF USUARIO_MEDICO,FECHA ON CITAS
FOR EACH ROW
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
V_FECHA DATE := :NEW.FECHA;
V_USUARIO VARCHAR2(40) := :NEW.USUARIO_MEDICO;
V_CUENTA NUMBER;
V_MAX NUMBER;
BEGIN
  SELECT COUNT(*)INTO V_CUENTA FROM CITAS WHERE USUARIO_MEDICO = V_USUARIO
AND FECHA=V_FECHA;
  SELECT MAXNUMCITAS INTO V_MAX FROM MEDICOS WHERE USUARIO_MEDICO =
V_USUARIO;
  IF V_CUENTA >= V_MAX THEN
    RAISE_APPLICATION_ERROR(-20001,'Un médico no puede tener más citas un mismo día que su
número máximo de citas permitido');
  END IF;
END;
/
```

--RN1

```
CREATE OR REPLACE TRIGGER TR_CITAS_CADA_20M
BEFORE INSERT OR UPDATE OF HORA ON CITAS
REFERENCING NEW AS N
FOR EACH ROW
DECLARE
  V_HORA TIMESTAMP := TO_TIMESTAMP(:N.HORA,'HH24:MI');
BEGIN
   IF EXTRACT(MINUTE FROM (V_HORA)) NOT IN (0,20,40) THEN
      RAISE_APPLICATION_ERROR (-20002, 'Las citas tienen que ser a en punto, y veinte o menos
veinte');
   END IF;
END;
/
```

--RN2

```
CREATE OR REPLACE TRIGGER TR_HORARIO_CLINICA
BEFORE INSERT OR UPDATE OF HORA ON CITAS
FOR EACH ROW
DECLARE
  V_HORA TIMESTAMP := TO_TIMESTAMP(:NEW.HORA,'HH24:MI');
BEGIN
  IF EXTRACT(HOUR FROM (V_HORA)) NOT IN(9,10,11,12,13,16,17,18,19) THEN
    RAISE_APPLICATION_ERROR (-20003, 'Las citas tienen respetar el horario de la clínica 9:00 a
13:00 y de 16:00 a 19:00');
  END IF;
END;
/
```

```
--RN3
CREATE OR REPLACE TRIGGER TR_CITA_IDENTICA
BEFORE INSERT OR UPDATE OF USUARIO_MEDICO,FECHA,HORA ON CITAS
FOR EACH ROW
DECLARE
  PRAGMA AUTONOMOUS_TRANSACTION;
  V_MEDICO VARCHAR(40) := :NEW.USUARIO_MEDICO;
  V_FECHA DATE := :NEW.FECHA;
  V_HORA VARCHAR2(20) := :NEW.HORA;
  V_CUENTA NUMBER;
BEGIN
  SELECT COUNT(*) INTO V_CUENTA FROM CITAS WHERE USUARIO_MEDICO = V_MEDICO
AND FECHA = V_FECHA AND HORA = V_HORA;
  IF V_CUENTA > 0 THEN
    RAISE_APPLICATION_ERROR(-20004,'No puede haber más de una cita de un mismo médico para
una fecha y una hora específica');
  END IF;
END;
/

--RN4
CREATE OR REPLACE TRIGGER TR_EMAIL_TRABAJADORES
AFTER INSERT OR UPDATE OF EMAIL ON TRABAJADORES
FOR EACH ROW
DECLARE
  V_EMAIL VARCHAR2(40) := :NEW.EMAIL;
BEGIN
  IF V_EMAIL NOT LIKE '%@caremujer.es' THEN
    RAISE_APPLICATION_ERROR(-20005, 'Un trabajador debe tener un email acabado en
@caremujer.es.');
   END IF;
END;
/

CREATE OR REPLACE TRIGGER TR_EMAIL_MEDICOS
AFTER INSERT OR UPDATE OF EMAIL ON MEDICOS
FOR EACH ROW
DECLARE
  V_EMAIL VARCHAR2(40) := :NEW.EMAIL;
BEGIN
  IF V_EMAIL NOT LIKE '%@caremujer.es' THEN
    RAISE_APPLICATION_ERROR(-20006, 'Un médico debe tener un email acabado en
@caremujer.es.');
   END IF;
END;
/
```

```
CREATE OR REPLACE TRIGGER TR_EMAIL_RECEPCIONISTA
AFTER INSERT OR UPDATE OF EMAIL ON RECEPCIONISTAS
FOR EACH ROW
DECLARE
  V_EMAIL VARCHAR2(40) := :NEW.EMAIL;
BEGIN
  IF V_EMAIL NOT LIKE '%@caremujer.es' THEN
    RAISE_APPLICATION_ERROR(-20007, 'Un recepcionista debe tener un email acabado en
@caremujer.es.');
   END IF;
END;
/

--RN5
CREATE OR REPLACE TRIGGER TR_CITAS_ENTRE_SEMANA
BEFORE INSERT OR UPDATE OF FECHA ON CITAS
FOR EACH ROW
DECLARE
V_FECHA DATE := :NEW.FECHA;
V_DIA VARCHAR2(20);
BEGIN
  SELECT TO_CHAR(V_FECHA, 'D') INTO V_DIA FROM DUAL;
  IF V_DIA IN (6,7) THEN
  RAISE_APPLICATION_ERROR(-20008, 'Los días de la semana donde se pueden pedir citas deben
ser entre lunes y viernes');
  END IF;
END;
/
```

PRUEBAS

```
--PAQUETES
--FUNCION AUXILIAR
CREATE OR REPLACE FUNCTION ASSERT_EQUALS (V_SALIDA BOOLEAN,
SALIDAESPERADA BOOLEAN)
RETURN VARCHAR2
IS
BEGIN
  IF V_SALIDA =SALIDAESPERADA THEN
    RETURN 'EXITO';
  ELSE
    RETURN 'FALLO';
  END IF;
END;
/

--CABECERAS DE PAQUETES
--TABLA CLINICAS
CREATE OR REPLACE PACKAGE PCK_CLINICAS
IS
PROCEDURE INICIALIZAR;
PROCEDURE CONSULTAR;
PROCEDURE INSERTAR (NOMBREPRUEBA VARCHAR2,V_NOMBRE_CLINICA IN
CLINICAS.NOMBRE_CLINICA%TYPE, V_DIRECCION IN CLINICAS.DIRECCION%TYPE,
SALIDAESPERADA BOOLEAN);
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2,V_OID_CLINICA IN
CLINICAS.OID_CLINICA%TYPE, V_NOMBRE_CLINICA IN
CLINICAS.NOMBRE_CLINICA%TYPE, V_DIRECCION IN CLINICAS.DIRECCION%TYPE,
SALIDAESPERADA BOOLEAN);
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_OID_CLINICA IN
CLINICAS.OID_CLINICA%TYPE, SALIDAESPERADA BOOLEAN);
END;
/

--TABLA RECEPCIONISTAS
CREATE OR REPLACE PACKAGE PCK_RECEPCIONISTAS
IS
PROCEDURE INICIALIZAR;
PROCEDURE CONSULTAR;
PROCEDURE INSERTAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
RECEPCIONISTAS.USUARIO%TYPE, V_CONTRASEÑA IN
RECEPCIONISTAS.CONTRASEÑA%TYPE,
  V_NOMBRE IN RECEPCIONISTAS.NOMBRE%TYPE, V_APELLIDOS IN
RECEPCIONISTAS.APELLIDOS%TYPE, V_DNI IN RECEPCIONISTAS.DNI%TYPE,
  V_EMAIL IN RECEPCIONISTAS.EMAIL%TYPE,V_TELEFONO IN
RECEPCIONISTAS.TELEFONO%TYPE, SALIDAESPERADA BOOLEAN);
```

```
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
RECEPCIONISTAS.USUARIO%TYPE, V_CONTRASEÑA IN
RECEPCIONISTAS.CONTRASEÑA%TYPE,
  V_NOMBRE IN RECEPCIONISTAS.NOMBRE%TYPE, V_APELLIDOS IN
RECEPCIONISTAS.APELLIDOS%TYPE, V_DNI IN RECEPCIONISTAS.DNI%TYPE,
  V_EMAIL IN RECEPCIONISTAS.EMAIL%TYPE,V_TELEFONO IN
RECEPCIONISTAS.TELEFONO%TYPE, SALIDAESPERADA BOOLEAN);
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
RECEPCIONISTAS.USUARIO%TYPE, SALIDAESPERADA BOOLEAN);
END;
/
```

--TABLA MEDICOS

```
CREATE OR REPLACE PACKAGE PCK_MEDICOS
IS
PROCEDURE INICIALIZAR;
PROCEDURE CONSULTAR;
PROCEDURE INSERTAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
MEDICOS.USUARIO_MEDICO%TYPE, V_CONTRASEÑA IN
MEDICOS.CONTRASEÑA%TYPE,
  V_NOMBRE IN MEDICOS.NOMBRE%TYPE, V_APELLIDOS IN
MEDICOS.APELLIDOS%TYPE, V_DNI IN MEDICOS.DNI%TYPE,
  V_EMAIL IN MEDICOS.EMAIL%TYPE,V_ESPECIALIDAD IN
MEDICOS.ESPECIALIDAD%TYPE, V_MAXNUMCITAS IN
MEDICOS.MAXNUMCITAS%TYPE, SALIDAESPERADA BOOLEAN);
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
MEDICOS.USUARIO_MEDICO%TYPE, V_CONTRASEÑA IN
MEDICOS.CONTRASEÑA%TYPE,
  V_NOMBRE IN MEDICOS.NOMBRE%TYPE, V_APELLIDOS IN
MEDICOS.APELLIDOS%TYPE, V_DNI IN MEDICOS.DNI%TYPE,
  V_EMAIL IN MEDICOS.EMAIL%TYPE,V_ESPECIALIDAD IN
MEDICOS.ESPECIALIDAD%TYPE, V_MAXNUMCITAS IN
MEDICOS.MAXNUMCITAS%TYPE, SALIDAESPERADA BOOLEAN);
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
MEDICOS.USUARIO_MEDICO%TYPE, SALIDAESPERADA BOOLEAN);
END;
/
```

--TABLA TRABAJADORES

```
CREATE OR REPLACE PACKAGE PCK_TRABAJADORES
IS
PROCEDURE INICIALIZAR;
PROCEDURE CONSULTAR;
PROCEDURE INSERTAR (NOMBREPRUEBA VARCHAR2, V_DNI IN
TRABAJADORES.DNI%TYPE, V_NOMBRE IN TRABAJADORES.NOMBRE%TYPE,
V_APELLIDOS IN TRABAJADORES.APELLIDOS%TYPE,
  V_EMAIL IN TRABAJADORES.EMAIL%TYPE, V_OID_CLINICA IN
TRABAJADORES.OID_CLINICA%TYPE, SALIDAESPERADA BOOLEAN);
```

```
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_DNI IN
TRABAJADORES.DNI%TYPE, V_NOMBRE IN TRABAJADORES.NOMBRE%TYPE,
V_APELLIDOS IN TRABAJADORES.APELLIDOS%TYPE,
  V_EMAIL IN TRABAJADORES.EMAIL%TYPE, V_OID_CLINICA IN
TRABAJADORES.OID_CLINICA%TYPE, SALIDAESPERADA BOOLEAN);
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_DNI IN
TRABAJADORES.DNI%TYPE, SALIDAESPERADA BOOLEAN);
END;
/

--TABLA TRABAJAEN
CREATE OR REPLACE PACKAGE PCK_TRABAJAEN
IS
PROCEDURE INICIALIZAR;
PROCEDURE CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_USUARIO IN
TRABAJAEN.USUARIO_MEDICO%TYPE, V_OID_CLINICA IN
TRABAJAEN.OID_CLINICA%TYPE,SALIDAESPERADA BOOLEAN);
PROCEDURE ACTUALIZAR(NOMBREPRUEBA VARCHAR2, V_OID_TE IN
TRABAJAEN.OID_TE%TYPE, V_USUARIO IN TRABAJAEN.USUARIO_MEDICO%TYPE,
V_OID_CLINICA IN TRABAJAEN.OID_CLINICA%TYPE, SALIDAESPERADA BOOLEAN);
PROCEDURE ELIMINAR(NOMBREPRUEBA VARCHAR2, V_OID_TE IN
TRABAJAEN.OID_TE%TYPE, SALIDAESPERADA BOOLEAN);
END;
/

--TABLA PACIENTES
CREATE OR REPLACE PACKAGE PCK_PACIENTES
IS
PROCEDURE INICIALIZAR;
PROCEDURE CONSULTAR;
PROCEDURE INSERTAR (NOMBREPRUEBA VARCHAR2, V_DNI IN
PACIENTES.DNI%TYPE,V_CONTRASEÑA IN PACIENTES.CONTRASEÑA%TYPE,
V_NOMBRE IN PACIENTES.NOMBRE%TYPE, V_APELLIDOS IN
PACIENTES.APELLIDOS%TYPE,
  V_EMAIL IN PACIENTES.EMAIL%TYPE,V_NUMEROSEGURO IN
PACIENTES.NUMEROSEGURO%TYPE, V_USUARIO_RECEPCIONISTA IN
PACIENTES.USUARIO_RECEPCIONISTA%TYPE, SALIDAESPERADA BOOLEAN);
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_DNI IN
PACIENTES.DNI%TYPE,V_CONTRASEÑA IN PACIENTES.CONTRASEÑA%TYPE,
V_NOMBRE IN PACIENTES.NOMBRE%TYPE, V_APELLIDOS IN
PACIENTES.APELLIDOS%TYPE,
  V_EMAIL IN PACIENTES.EMAIL%TYPE,V_NUMEROSEGURO IN
PACIENTES.NUMEROSEGURO%TYPE, V_USUARIO_RECEPCIONISTA IN
PACIENTES.USUARIO_RECEPCIONISTA%TYPE, SALIDAESPERADA BOOLEAN);
PROCEDURE ELIMINAR(NOMBREPRUEBA VARCHAR2, V_DNI IN
PACIENTES.DNI%TYPE, SALIDAESPERADA BOOLEAN);
END;
```

```
/
--TABLA CITAS
CREATE OR REPLACE PACKAGE PCK_CITAS
IS
PROCEDURE INICIALIZAR;
PROCEDURE CONSULTAR;
PROCEDURE INSERTAR (NOMBREPRUEBA VARCHAR2, V_HORA IN
CITAS.HORA%TYPE, V_FECHA IN CITAS.FECHA%TYPE, V_OID_CLINICA IN
CITAS.OID_CLINICA%TYPE,
  V_USUARIO_MEDICO IN CITAS.USUARIO_MEDICO%TYPE,
V_USUARIO_RECEPCIONISTA IN CITAS.USUARIO_RECEPCIONISTA%TYPE,
V_DNI_PACIENTE IN CITAS.DNI_PACIENTE%TYPE,
  SALIDAESPERADA BOOLEAN);
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2,V_OID_CITA IN
CITAS.OID_CITA%TYPE, V_DNI_PACIENTE IN CITAS.DNI_PACIENTE%TYPE,
  SALIDAESPERADA BOOLEAN);
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_OID_CITA IN
CITAS.OID_CITA%TYPE, SALIDAESPERADA BOOLEAN);
END;
/

--TABLA HISTORIALES
CREATE OR REPLACE PACKAGE PCK_HISTORIALES
IS
PROCEDURE INICIALIZAR;
PROCEDURE CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2,V_CODIGO IN
HISTORIALES.CODIGO%TYPE, V_NOMBREPACIENTE IN
HISTORIALES.NOMBREPACIENTE%TYPE, V_APELLIDOSPACIENTE IN
HISTORIALES.APELLIDOSPACIENTE%TYPE,
  V_USUARIO_MEDICO IN HISTORIALES.USUARIO_MEDICO%TYPE,V_DNI_PACIENTE
IN HISTORIALES.DNI_PACIENTE%TYPE, SALIDAESPERADA BOOLEAN);
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2,V_CODIGO IN
HISTORIALES.CODIGO%TYPE, V_NOMBREPACIENTE IN
HISTORIALES.NOMBREPACIENTE%TYPE, V_APELLIDOSPACIENTE IN
HISTORIALES.APELLIDOSPACIENTE%TYPE,
  V_USUARIO_MEDICO IN HISTORIALES.USUARIO_MEDICO%TYPE,V_DNI_PACIENTE
IN HISTORIALES.DNI_PACIENTE%TYPE, SALIDAESPERADA BOOLEAN);
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_CODIGO IN
HISTORIALES.CODIGO%TYPE, SALIDAESPERADA BOOLEAN);
END;
/

--TABLA LINEASDEHISTORIAL
CREATE OR REPLACE PACKAGE PCK_LINEASDEHISTORIAL
IS
PROCEDURE INICIALIZAR;
PROCEDURE CONSULTAR;
```

```
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_FECHA IN
LINEASDEHISTORIAL.FECHA%TYPE, V_OBSERVACIONES IN
LINEASDEHISTORIAL.OBSERVACIONES%TYPE,
  V_CODIGO_HISTORIAL IN LINEASDEHISTORIAL.CODIGO_HISTORIAL%TYPE,
SALIDAESPERADA BOOLEAN);
PROCEDURE ACTUALIZAR(NOMBREPRUEBA VARCHAR2, V_OID_LH IN
LINEASDEHISTORIAL.OID_LH%TYPE, V_FECHA IN
LINEASDEHISTORIAL.FECHA%TYPE, V_OBSERVACIONES IN
LINEASDEHISTORIAL.OBSERVACIONES%TYPE,
  V_CODIGO_HISTORIAL IN LINEASDEHISTORIAL.CODIGO_HISTORIAL%TYPE,
SALIDAESPERADA BOOLEAN);
PROCEDURE ELIMINAR(NOMBREPRUEBA VARCHAR2, V_OID_LH IN
LINEASDEHISTORIAL.OID_LH%TYPE,SALIDAESPERADA BOOLEAN);
END;
/

--CUERPO DE PAQUETES
--TABLA CLINICAS
CREATE OR REPLACE PACKAGE BODY PCK_CLINICAS
IS
  CURSOR C IS
    SELECT * FROM CLINICAS;
  V_SALIDA BOOLEAN := TRUE;
  V_CLINICAS CLINICAS%ROWTYPE;
PROCEDURE INICIALIZAR
  IS
  BEGIN
    DELETE FROM CLINICAS;
END INICIALIZAR;
PROCEDURE CONSULTAR
  IS
  BEGIN
    OPEN C;
    FETCH C INTO V_CLINICAS;
    DBMS_OUTPUT.PUT_LINE(RPAD('OID_CLINICA:',35) || RPAD('NOMBRE_CLINICA:',35)
|| RPAD('DIRECCION:',35));
    DBMS_OUTPUT.PUT_LINE(LPAD('-',100,'-'));
    WHILE C%FOUND LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(V_CLINICAS.OID_CLINICA,35) ||
RPAD(V_CLINICAS.NOMBRE_CLINICA,35) || RPAD(V_CLINICAS.DIRECCION,35));
    FETCH C INTO V_CLINICAS;
    END LOOP;
    CLOSE C;
  END CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_NOMBRE_CLINICA IN
CLINICAS.NOMBRE_CLINICA%TYPE, V_DIRECCION IN CLINICAS.DIRECCION%TYPE,
SALIDAESPERADA BOOLEAN)
  IS
```

```
    V_OID_CLINICA CLINICAS.OID_CLINICA%TYPE;
  BEGIN
    INSERT INTO CLINICAS (NOMBRE_CLINICA,DIRECCION) VALUES
(V_NOMBRE_CLINICA, V_DIRECCION);
    V_OID_CLINICA := SEC_CLINICAS.CURRVAL;
    SELECT * INTO V_CLINICAS FROM CLINICAS WHERE OID_CLINICA =
V_OID_CLINICA;
    IF V_CLINICAS.NOMBRE_CLINICA != V_NOMBRE_CLINICA AND
V_CLINICAS.DIRECCION != V_DIRECCION THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END INSERTAR;
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_OID_CLINICA IN
CLINICAS.OID_CLINICA%TYPE, V_NOMBRE_CLINICA IN
CLINICAS.NOMBRE_CLINICA%TYPE, V_DIRECCION IN CLINICAS.DIRECCION%TYPE,
SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    UPDATE CLINICAS SET NOMBRE_CLINICA = V_NOMBRE_CLINICA, DIRECCION =
V_DIRECCION WHERE OID_CLINICA = V_OID_CLINICA;
    SELECT * INTO V_CLINICAS FROM CLINICAS WHERE OID_CLINICA =
V_OID_CLINICA;
    IF V_CLINICAS.NOMBRE_CLINICA != V_NOMBRE_CLINICA AND
V_CLINICAS.DIRECCION != V_DIRECCION THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE (NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END ACTUALIZAR;
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_OID_CLINICA IN
CLINICAS.OID_CLINICA%TYPE, SALIDAESPERADA BOOLEAN)
  IS
     V_NUMCLINICAS NUMBER := 0;
  BEGIN
```

```
    DELETE FROM CLINICAS WHERE OID_CLINICA = V_OID_CLINICA;
    SELECT COUNT(*) INTO V_NUMCLINICAS FROM CLINICAS WHERE OID_CLINICA =
V_OID_CLINICA;
    IF V_NUMCLINICAS != 0 THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (FALSE,
SALIDAESPERADA));
        ROLLBACK;
    END ELIMINAR;
  END;
/
--TABLA RECEPCIONISTAS
create or replace PACKAGE BODY PCK_RECEPCIONISTAS
IS
  CURSOR C IS
    SELECT * FROM RECEPCIONISTAS;
  V_SALIDA BOOLEAN := TRUE;
  V_RECEPCIONISTAS RECEPCIONISTAS%ROWTYPE;
PROCEDURE INICIALIZAR
  IS
  BEGIN
    DELETE FROM RECEPCIONISTAS;
  END INICIALIZAR;
PROCEDURE CONSULTAR
  IS
  BEGIN
    OPEN C;
    FETCH C INTO V_RECEPCIONISTAS;
    DBMS_OUTPUT.PUT_LINE(RPAD('USUARIO:',25) || RPAD('CONTRASEÑA:',25) ||
RPAD('NOMBRE:',25)|| RPAD('APELLIDOS:',25)|| RPAD('DNI:',25)
      || RPAD('EMAIL:',25)|| RPAD('TELEFONO:',25));
    DBMS_OUTPUT.PUT_LINE(LPAD('-',200,'-'));
    WHILE C%FOUND LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(V_RECEPCIONISTAS.USUARIO,25) ||
RPAD(V_RECEPCIONISTAS.CONTRASEÑA,25) ||
RPAD(V_RECEPCIONISTAS.NOMBRE,25)|| RPAD(V_RECEPCIONISTAS.APELLIDOS,25)
      || RPAD(V_RECEPCIONISTAS.DNI,25)|| RPAD(V_RECEPCIONISTAS.EMAIL,25)||
RPAD(V_RECEPCIONISTAS.TELEFONO,25));
    FETCH C INTO V_RECEPCIONISTAS;
    END LOOP;
    CLOSE C;
  END CONSULTAR;
```

```
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_USUARIO IN
RECEPCIONISTAS.USUARIO%TYPE, V_CONTRASEÑA IN
RECEPCIONISTAS.CONTRASEÑA%TYPE,V_NOMBRE IN
RECEPCIONISTAS.NOMBRE%TYPE,
V_APELLIDOS IN RECEPCIONISTAS.APELLIDOS%TYPE, V_DNI IN
RECEPCIONISTAS.DNI%TYPE, V_EMAIL IN RECEPCIONISTAS.EMAIL%TYPE,
V_TELEFONO IN RECEPCIONISTAS.TELEFONO%TYPE,SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    INSERT INTO RECEPCIONISTAS
(USUARIO,CONTRASEÑA,NOMBRE,APELLIDOS,DNI,EMAIL,TELEFONO) VALUES
(V_USUARIO,V_CONTRASEÑA,V_NOMBRE, V_APELLIDOS, V_DNI, V_EMAIL,
V_TELEFONO);
    SELECT * INTO V_RECEPCIONISTAS FROM RECEPCIONISTAS WHERE USUARIO =
V_USUARIO;
    IF V_RECEPCIONISTAS.USUARIO != V_USUARIO AND
V_RECEPCIONISTAS.CONTRASEÑA != V_CONTRASEÑA AND
V_RECEPCIONISTAS.NOMBRE != V_NOMBRE
      AND V_RECEPCIONISTAS.APELLIDOS != V_APELLIDOS AND
V_RECEPCIONISTAS.DNI != V_DNI AND V_RECEPCIONISTAS.EMAIL != V_EMAIL AND
      V_RECEPCIONISTAS.TELEFONO != V_TELEFONO THEN
        V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END INSERTAR;
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
RECEPCIONISTAS.USUARIO%TYPE, V_CONTRASEÑA IN
RECEPCIONISTAS.CONTRASEÑA%TYPE,V_NOMBRE IN
RECEPCIONISTAS.NOMBRE%TYPE,
V_APELLIDOS IN RECEPCIONISTAS.APELLIDOS%TYPE, V_DNI IN
RECEPCIONISTAS.DNI%TYPE, V_EMAIL IN RECEPCIONISTAS.EMAIL%TYPE,
V_TELEFONO IN RECEPCIONISTAS.TELEFONO%TYPE,SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    UPDATE RECEPCIONISTAS SET CONTRASEÑA = V_CONTRASEÑA, NOMBRE =
V_NOMBRE, APELLIDOS = V_APELLIDOS, DNI= V_DNI, EMAIL = V_EMAIL, TELEFONO =
V_TELEFONO WHERE USUARIO = V_USUARIO;
    SELECT * INTO V_RECEPCIONISTAS FROM RECEPCIONISTAS WHERE USUARIO =
V_USUARIO;
```

```
    IF V_RECEPCIONISTAS.USUARIO != V_USUARIO AND
V_RECEPCIONISTAS.CONTRASEÑA != V_CONTRASEÑA AND
V_RECEPCIONISTAS.NOMBRE != V_NOMBRE
      AND V_RECEPCIONISTAS.APELLIDOS != V_APELLIDOS AND
V_RECEPCIONISTAS.DNI != V_DNI AND V_RECEPCIONISTAS.EMAIL != V_EMAIL AND
      V_RECEPCIONISTAS.TELEFONO != V_TELEFONO THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE (NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END ACTUALIZAR;
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
RECEPCIONISTAS.USUARIO%TYPE, SALIDAESPERADA BOOLEAN)
  IS
      V_NUMRECEPCIONISTAS NUMBER := 0;
  BEGIN
    DELETE FROM RECEPCIONISTAS WHERE USUARIO = V_USUARIO;
    SELECT COUNT(*) INTO V_NUMRECEPCIONISTAS FROM RECEPCIONISTAS WHERE
USUARIO = V_USUARIO;
    IF V_NUMRECEPCIONISTAS != 0 THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (FALSE,
SALIDAESPERADA));
        ROLLBACK;
    END ELIMINAR;
  END;
/

--TABLA MEDICOS
CREATE OR REPLACE PACKAGE BODY PCK_MEDICOS
IS
  CURSOR C IS
    SELECT * FROM MEDICOS;
  V_SALIDA BOOLEAN := TRUE;
  V_MEDICOS MEDICOS%ROWTYPE;
PROCEDURE INICIALIZAR
```

```
  IS
  BEGIN
    DELETE FROM MEDICOS;
  END INICIALIZAR;
PROCEDURE CONSULTAR
  IS
  BEGIN
    OPEN C;
    FETCH C INTO V_MEDICOS;
    DBMS_OUTPUT.PUT_LINE(RPAD('USUARIO_MEDICO:',25) ||
RPAD('CONTRASEÑA:',25) || RPAD('NOMBRE:',25)|| RPAD('APELLIDOS:',25)||
RPAD('DNI:',25)
      || RPAD('EMAIL:',25)|| RPAD('ESPECIALIDAD:',25)|| RPAD('MAXNUMCITAS:',25));
    DBMS_OUTPUT.PUT_LINE(LPAD('-',200,'-'));
    WHILE C%FOUND LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(V_MEDICOS.USUARIO_MEDICO,25) ||
RPAD(V_MEDICOS.CONTRASEÑA,25) || RPAD(V_MEDICOS.NOMBRE,25)||
RPAD(V_MEDICOS.APELLIDOS,25)
      || RPAD(V_MEDICOS.DNI,25)|| RPAD(V_MEDICOS.EMAIL,25)||
RPAD(V_MEDICOS.ESPECIALIDAD,25)|| RPAD(V_MEDICOS.MAXNUMCITAS,25));
    FETCH C INTO V_MEDICOS;
    END LOOP;
    CLOSE C;
  END CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_USUARIO IN
MEDICOS.USUARIO_MEDICO%TYPE, V_CONTRASEÑA IN
MEDICOS.CONTRASEÑA%TYPE,V_NOMBRE IN MEDICOS.NOMBRE%TYPE,
V_APELLIDOS IN MEDICOS.APELLIDOS%TYPE, V_DNI IN MEDICOS.DNI%TYPE,
V_EMAIL IN MEDICOS.EMAIL%TYPE, V_ESPECIALIDAD IN
MEDICOS.ESPECIALIDAD%TYPE,V_MAXNUMCITAS IN
MEDICOS.MAXNUMCITAS%TYPE,SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    INSERT INTO MEDICOS
(USUARIO_MEDICO,CONTRASEÑA,NOMBRE,APELLIDOS,DNI,EMAIL,ESPECIALIDAD,MA
XNUMCITAS) VALUES (V_USUARIO,V_CONTRASEÑA,V_NOMBRE, V_APELLIDOS,
V_DNI, V_EMAIL, V_ESPECIALIDAD, V_MAXNUMCITAS);
    SELECT * INTO V_MEDICOS FROM MEDICOS WHERE USUARIO_MEDICO =
V_USUARIO;
    IF V_MEDICOS.USUARIO_MEDICO != V_USUARIO AND V_MEDICOS.CONTRASEÑA !=
V_CONTRASEÑA AND V_MEDICOS.NOMBRE != V_NOMBRE
      AND V_MEDICOS.APELLIDOS != V_APELLIDOS AND V_MEDICOS.DNI != V_DNI AND
V_MEDICOS.EMAIL != V_EMAIL AND
      V_MEDICOS.ESPECIALIDAD != V_ESPECIALIDAD AND V_MEDICOS.MAXNUMCITAS
!= V_MAXNUMCITAS THEN
        V_SALIDA := FALSE;
    END IF;
    COMMIT;
```

```
   DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (V_SALIDA,
SALIDAESPERADA));
   EXCEPTION
     WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
       ROLLBACK;
  END INSERTAR;
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
MEDICOS.USUARIO_MEDICO%TYPE, V_CONTRASEÑA IN
MEDICOS.CONTRASEÑA%TYPE,V_NOMBRE IN MEDICOS.NOMBRE%TYPE,
V_APELLIDOS IN MEDICOS.APELLIDOS%TYPE, V_DNI IN MEDICOS.DNI%TYPE,
V_EMAIL IN MEDICOS.EMAIL%TYPE, V_ESPECIALIDAD IN
MEDICOS.ESPECIALIDAD%TYPE,V_MAXNUMCITAS IN
MEDICOS.MAXNUMCITAS%TYPE,SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    UPDATE MEDICOS SET CONTRASEÑA = V_CONTRASEÑA, NOMBRE = V_NOMBRE,
APELLIDOS = V_APELLIDOS, DNI= V_DNI, EMAIL = V_EMAIL,
      ESPECIALIDAD = V_ESPECIALIDAD, MAXNUMCITAS = V_MAXNUMCITAS WHERE
USUARIO_MEDICO = V_USUARIO;
    SELECT * INTO V_MEDICOS FROM MEDICOS WHERE USUARIO_MEDICO =
V_USUARIO;
    IF V_MEDICOS.USUARIO_MEDICO != V_USUARIO AND V_MEDICOS.CONTRASEÑA !=
V_CONTRASEÑA AND V_MEDICOS.NOMBRE != V_NOMBRE
      AND V_MEDICOS.APELLIDOS != V_APELLIDOS AND V_MEDICOS.DNI != V_DNI AND
V_MEDICOS.EMAIL != V_EMAIL AND
      V_MEDICOS.ESPECIALIDAD != V_ESPECIALIDAD AND V_MEDICOS.MAXNUMCITAS
!= V_MAXNUMCITAS THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE (NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END ACTUALIZAR;
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_USUARIO IN
MEDICOS.USUARIO_MEDICO%TYPE, SALIDAESPERADA BOOLEAN)
  IS
     V_NUMMEDICOS NUMBER := 0;
  BEGIN
    DELETE FROM MEDICOS WHERE USUARIO_MEDICO = V_USUARIO;
    SELECT COUNT(*) INTO V_NUMMEDICOS FROM MEDICOS WHERE
USUARIO_MEDICO = V_USUARIO;
```

```
    IF V_NUMMEDICOS != 0 THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (FALSE,
SALIDAESPERADA));
        ROLLBACK;
    END ELIMINAR;
  END;
/

--TABLA TRABAJADORES
CREATE OR REPLACE PACKAGE BODY PCK_TRABAJADORES
IS
  CURSOR C IS
    SELECT * FROM TRABAJADORES;
  V_SALIDA BOOLEAN := TRUE;
  V_TRABAJADORES TRABAJADORES%ROWTYPE;
PROCEDURE INICIALIZAR
  IS
  BEGIN
    DELETE FROM TRABAJADORES;
  END INICIALIZAR;
PROCEDURE CONSULTAR
  IS
  BEGIN
    OPEN C;
    FETCH C INTO V_TRABAJADORES;
    DBMS_OUTPUT.PUT_LINE( RPAD('DNI:',25) || RPAD('NOMBRE:',25)||
RPAD('APELLIDOS:',25)
      || RPAD('EMAIL:',25)|| RPAD('OID_CLINICA:',25));
    DBMS_OUTPUT.PUT_LINE(LPAD('-',100,'-'));
    WHILE C%FOUND LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(V_TRABAJADORES.DNI,25) ||
RPAD(V_TRABAJADORES.NOMBRE,25)|| RPAD(V_TRABAJADORES.APELLIDOS,25)
      || RPAD(V_TRABAJADORES.EMAIL,25)||
RPAD(V_TRABAJADORES.OID_CLINICA,25));
    FETCH C INTO V_TRABAJADORES;
    END LOOP;
    CLOSE C;
  END CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_DNI IN
TRABAJADORES.DNI%TYPE, V_NOMBRE IN TRABAJADORES.NOMBRE%TYPE,
```

```
V_APELLIDOS IN TRABAJADORES.APELLIDOS%TYPE, V_EMAIL IN
TRABAJADORES.EMAIL%TYPE, V_OID_CLINICA IN
TRABAJADORES.OID_CLINICA%TYPE,SALIDAESPERADA BOOLEAN)
 IS
 BEGIN
   INSERT INTO TRABAJADORES (DNI,NOMBRE,APELLIDOS,EMAIL,OID_CLINICA)
VALUES (V_DNI, V_NOMBRE, V_APELLIDOS, V_EMAIL, V_OID_CLINICA);
   SELECT * INTO V_TRABAJADORES FROM TRABAJADORES WHERE DNI = V_DNI;
   IF V_TRABAJADORES.NOMBRE != V_NOMBRE
    AND V_TRABAJADORES.APELLIDOS != V_APELLIDOS AND
V_TRABAJADORES.EMAIL != V_EMAIL AND
     V_TRABAJADORES.OID_CLINICA != V_OID_CLINICA THEN
      V_SALIDA := FALSE;
   END IF;
   COMMIT;
   DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (V_SALIDA,
SALIDAESPERADA));
   EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
      ROLLBACK;
 END INSERTAR;
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_DNI IN
TRABAJADORES.DNI%TYPE, V_NOMBRE IN TRABAJADORES.NOMBRE%TYPE,
V_APELLIDOS IN TRABAJADORES.APELLIDOS%TYPE, V_EMAIL IN
TRABAJADORES.EMAIL%TYPE, V_OID_CLINICA IN
TRABAJADORES.OID_CLINICA%TYPE,SALIDAESPERADA BOOLEAN)
 IS
 BEGIN
   UPDATE TRABAJADORES SET  NOMBRE = V_NOMBRE, APELLIDOS = V_APELLIDOS,
EMAIL = V_EMAIL,
    OID_CLINICA = V_OID_CLINICA WHERE DNI = V_DNI;
   SELECT * INTO V_TRABAJADORES FROM TRABAJADORES WHERE DNI = V_DNI;
   IF V_TRABAJADORES.NOMBRE != V_NOMBRE AND V_TRABAJADORES.APELLIDOS
!= V_APELLIDOS AND V_TRABAJADORES.EMAIL != V_EMAIL AND
     V_TRABAJADORES.OID_CLINICA != V_OID_CLINICA THEN
     V_SALIDA := FALSE;
   END IF;
   COMMIT;
   DBMS_OUTPUT.PUT_LINE (NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
   EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
      ROLLBACK;
 END ACTUALIZAR;
```

```
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_DNI IN
TRABAJADORES.DNI%TYPE, SALIDAESPERADA BOOLEAN)
  IS
      V_NUMTRABAJADORES NUMBER := 0;
  BEGIN
    DELETE FROM TRABAJADORES WHERE DNI = V_DNI;
    SELECT COUNT(*) INTO V_NUMTRABAJADORES FROM TRABAJADORES WHERE
DNI = V_DNI;
    IF V_NUMTRABAJADORES != 0 THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (FALSE,
SALIDAESPERADA));
        ROLLBACK;
    END ELIMINAR;
  END;
/

--TABLA TRABAJAEN
CREATE OR REPLACE PACKAGE BODY PCK_TRABAJAEN
IS
  CURSOR C IS
    SELECT * FROM TRABAJAEN;
  V_SALIDA BOOLEAN := TRUE;
  V_TRABAJAEN TRABAJAEN%ROWTYPE;
PROCEDURE INICIALIZAR
  IS
  BEGIN
    DELETE FROM TRABAJAEN;
  END INICIALIZAR;
PROCEDURE CONSULTAR
  IS
  BEGIN
    OPEN C;
    FETCH C INTO V_TRABAJAEN;
    DBMS_OUTPUT.PUT_LINE(RPAD('OID_TE:',25) || RPAD('USUARIO_MEDICO:',25) ||
RPAD('OID_CLINICA:',25));
    DBMS_OUTPUT.PUT_LINE(LPAD('-',100,'-'));
    WHILE C%FOUND LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(V_TRABAJAEN.OID_TE,25) ||
RPAD(V_TRABAJAEN.USUARIO_MEDICO,25) || RPAD(V_TRABAJAEN.OID_CLINICA,25));
    FETCH C INTO V_TRABAJAEN;
    END LOOP;
```

```
    CLOSE C;
  END CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_USUARIO IN
TRABAJAEN.USUARIO_MEDICO%TYPE, V_OID_CLINICA IN
TRABAJAEN.OID_CLINICA%TYPE, SALIDAESPERADA BOOLEAN)
  IS
    V_OID_TE TRABAJAEN.OID_TE%TYPE;
  BEGIN
    INSERT INTO TRABAJAEN (USUARIO_MEDICO,OID_CLINICA) VALUES (V_USUARIO,
V_OID_CLINICA);
    V_OID_TE := SEC_TRABAJAEN.CURRVAL;
    SELECT * INTO V_TRABAJAEN FROM TRABAJAEN WHERE OID_TE = V_OID_TE;
    IF V_TRABAJAEN.USUARIO_MEDICO != V_USUARIO AND
V_TRABAJAEN.OID_CLINICA != V_OID_CLINICA THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END INSERTAR;
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_OID_TE IN
TRABAJAEN.OID_TE%TYPE, V_USUARIO IN TRABAJAEN.USUARIO_MEDICO%TYPE,
V_OID_CLINICA IN TRABAJAEN.OID_CLINICA%TYPE, SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    UPDATE TRABAJAEN SET USUARIO_MEDICO = V_USUARIO, OID_CLINICA =
V_OID_CLINICA WHERE OID_TE = V_OID_TE;
    SELECT * INTO V_TRABAJAEN FROM TRABAJAEN WHERE OID_TE = V_OID_TE;
    IF V_TRABAJAEN.USUARIO_MEDICO != V_USUARIO AND
V_TRABAJAEN.OID_CLINICA != V_OID_CLINICA THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE (NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END ACTUALIZAR;
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_OID_TE IN
TRABAJAEN.OID_TE%TYPE, SALIDAESPERADA BOOLEAN)
```

```
  IS
     V_NUMTRABAJAEN NUMBER := 0;
  BEGIN
    DELETE FROM TRABAJAEN WHERE OID_TE = V_OID_TE;
    SELECT COUNT(*) INTO V_NUMTRABAJAEN FROM TRABAJAEN WHERE OID_TE =
V_OID_TE;
    IF V_NUMTRABAJAEN != 0 THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (FALSE,
SALIDAESPERADA));
        ROLLBACK;
    END ELIMINAR;
  END;
/

--TABLA PACIENTES
CREATE OR REPLACE PACKAGE BODY PCK_PACIENTES
IS
  CURSOR C IS
    SELECT * FROM PACIENTES;
  V_SALIDA BOOLEAN := TRUE;
  V_PACIENTES PACIENTES%ROWTYPE;
PROCEDURE INICIALIZAR
  IS
  BEGIN
    DELETE FROM PACIENTES;
  END INICIALIZAR;
PROCEDURE CONSULTAR
  IS
  BEGIN
    OPEN C;
    FETCH C INTO V_PACIENTES;
    DBMS_OUTPUT.PUT_LINE( RPAD('DNI:',25) || RPAD('CONTRASEÑA:',25)
||RPAD('NOMBRE:',25)|| RPAD('APELLIDOS:',25)
      || RPAD('EMAIL:',25)|| RPAD('NUMEROSEGURO:',25)||
RPAD('USUARIO_RECEPCIONISTA:',25));
    DBMS_OUTPUT.PUT_LINE(LPAD('-',200,'-'));
    WHILE C%FOUND LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(V_PACIENTES.DNI,25) ||
RPAD(V_PACIENTES.CONTRASEÑA,25) || RPAD(V_PACIENTES.NOMBRE,25)||
RPAD(V_PACIENTES.APELLIDOS,25)
```

```
     || RPAD(V_PACIENTES.EMAIL,25)|| RPAD(V_PACIENTES.NUMEROSEGURO,25)||
RPAD(V_PACIENTES.USUARIO_RECEPCIONISTA,25));
    FETCH C INTO V_PACIENTES;
    END LOOP;
    CLOSE C;
  END CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_DNI IN
PACIENTES.DNI%TYPE, V_CONTRASEÑA IN
PACIENTES.CONTRASEÑA%TYPE,V_NOMBRE IN PACIENTES.NOMBRE%TYPE,
V_APELLIDOS IN PACIENTES.APELLIDOS%TYPE, V_EMAIL IN
PACIENTES.EMAIL%TYPE, V_NUMEROSEGURO IN
PACIENTES.NUMEROSEGURO%TYPE,V_USUARIO_RECEPCIONISTA IN
PACIENTES.USUARIO_RECEPCIONISTA%TYPE,SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    INSERT INTO PACIENTES
(DNI,CONTRASEÑA,NOMBRE,APELLIDOS,EMAIL,NUMEROSEGURO,USUARIO_RECEPC
IONISTA) VALUES (V_DNI,V_CONTRASEÑA, V_NOMBRE, V_APELLIDOS, V_EMAIL,
V_NUMEROSEGURO, V_USUARIO_RECEPCIONISTA);
    SELECT * INTO V_PACIENTES FROM PACIENTES WHERE DNI = V_DNI;
    IF V_PACIENTES.NOMBRE != V_NOMBRE AND V_PACIENTES.CONTRASEÑA !=
V_CONTRASEÑA
      AND V_PACIENTES.APELLIDOS != V_APELLIDOS AND V_PACIENTES.EMAIL !=
V_EMAIL AND
      V_PACIENTES.NUMEROSEGURO != V_NUMEROSEGURO  AND
V_PACIENTES.USUARIO_RECEPCIONISTA != V_USUARIO_RECEPCIONISTA THEN
        V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END INSERTAR;
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_DNI IN
PACIENTES.DNI%TYPE, V_CONTRASEÑA IN PACIENTES.CONTRASEÑA%TYPE,
V_NOMBRE IN PACIENTES.NOMBRE%TYPE,
V_APELLIDOS IN PACIENTES.APELLIDOS%TYPE, V_EMAIL IN
PACIENTES.EMAIL%TYPE, V_NUMEROSEGURO IN
PACIENTES.NUMEROSEGURO%TYPE,V_USUARIO_RECEPCIONISTA IN
PACIENTES.USUARIO_RECEPCIONISTA%TYPE,SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    UPDATE PACIENTES SET  CONTRASEÑA=V_CONTRASEÑA, NOMBRE = V_NOMBRE,
APELLIDOS = V_APELLIDOS, EMAIL = V_EMAIL,
```

```
      NUMEROSEGURO = V_NUMEROSEGURO, USUARIO_RECEPCIONISTA =
V_USUARIO_RECEPCIONISTA  WHERE DNI = V_DNI;
    SELECT * INTO V_PACIENTES FROM PACIENTES WHERE DNI = V_DNI;
    IF V_PACIENTES.CONTRASEÑA != V_CONTRASEÑA AND V_PACIENTES.NOMBRE !=
V_NOMBRE
      AND V_PACIENTES.APELLIDOS != V_APELLIDOS AND V_PACIENTES.EMAIL !=
V_EMAIL AND
      V_PACIENTES.NUMEROSEGURO != V_NUMEROSEGURO  AND
V_PACIENTES.USUARIO_RECEPCIONISTA != V_USUARIO_RECEPCIONISTA THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE (NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END ACTUALIZAR;
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_DNI IN
PACIENTES.DNI%TYPE, SALIDAESPERADA BOOLEAN)
  IS
      V_NUMPACIENTES NUMBER := 0;
  BEGIN
    DELETE FROM PACIENTES WHERE DNI = V_DNI;
    SELECT COUNT(*) INTO V_NUMPACIENTES FROM PACIENTES WHERE DNI = V_DNI;
    IF V_NUMPACIENTES != 0 THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (FALSE,
SALIDAESPERADA));
        ROLLBACK;
    END ELIMINAR;
  END;
/
--TABLA CITAS
CREATE OR REPLACE PACKAGE BODY PCK_CITAS
IS
  CURSOR C IS
    SELECT * FROM CITAS;
  V_SALIDA BOOLEAN := TRUE;
  V_CITAS CITAS%ROWTYPE;
```

```
PROCEDURE INICIALIZAR
  IS
  BEGIN
    DELETE FROM CITAS;
  END INICIALIZAR;
PROCEDURE CONSULTAR
  IS
  BEGIN
    OPEN C;
    FETCH C INTO V_CITAS;
    DBMS_OUTPUT.PUT_LINE(RPAD('OID_CITA:',25) || RPAD('HORA:',25) ||
RPAD('FECHA:',25)|| RPAD('OID_CLINICA:',25)|| RPAD('USUARIO_MEDICO:',25)
      || RPAD('USUARIO_RECEPCIONISTA:',25)|| RPAD('DNI_PACIENTE:',25));
    DBMS_OUTPUT.PUT_LINE(LPAD('-',200,'-'));
    WHILE C%FOUND LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(V_CITAS.OID_CITA,25) || RPAD(V_CITAS.HORA,25)
|| RPAD(V_CITAS.FECHA,25)|| RPAD(V_CITAS.OID_CLINICA,25)
       || RPAD(V_CITAS.USUARIO_MEDICO,25)||
RPAD(V_CITAS.USUARIO_RECEPCIONISTA,25)|| RPAD(V_CITAS.DNI_PACIENTE,25));
    FETCH C INTO V_CITAS;
    END LOOP;
    CLOSE C;
  END CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_HORA IN
CITAS.HORA%TYPE, V_FECHA IN CITAS.FECHA%TYPE, V_OID_CLINICA IN
CITAS.OID_CLINICA%TYPE,
  V_USUARIO_MEDICO IN CITAS.USUARIO_MEDICO%TYPE,
V_USUARIO_RECEPCIONISTA IN CITAS.USUARIO_RECEPCIONISTA%TYPE,
V_DNI_PACIENTE IN CITAS.DNI_PACIENTE%TYPE,
  SALIDAESPERADA BOOLEAN)
  IS
    V_OID_CITA CITAS.OID_CITA%TYPE;
  BEGIN
    INSERT INTO CITAS (HORA, FECHA, OID_CLINICA, USUARIO_MEDICO,
USUARIO_RECEPCIONISTA, DNI_PACIENTE) VALUES (V_HORA, V_FECHA,
V_OID_CLINICA, V_USUARIO_MEDICO,
      V_USUARIO_RECEPCIONISTA, V_DNI_PACIENTE);
    V_OID_CITA := SEC_CITAS.CURRVAL;
    SELECT * INTO V_CITAS FROM CITAS WHERE OID_CITA = V_OID_CITA;
    IF V_CITAS.HORA != V_HORA AND V_CITAS.FECHA != V_FECHA  AND
V_CITAS.OID_CLINICA != V_OID_CLINICA
      AND V_CITAS.USUARIO_MEDICO != V_USUARIO_MEDICO AND
V_CITAS.USUARIO_RECEPCIONISTA != V_USUARIO_RECEPCIONISTA AND
V_CITAS.DNI_PACIENTE != V_DNI_PACIENTE  THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
```

```
   DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (V_SALIDA,
SALIDAESPERADA));
   EXCEPTION
     WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
       ROLLBACK;
  END INSERTAR;
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2,V_OID_CITA IN
CITAS.OID_CITA%TYPE, V_DNI_PACIENTE IN CITAS.DNI_PACIENTE%TYPE,
  SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
   UPDATE CITAS SET  DNI_PACIENTE = V_DNI_PACIENTE WHERE OID_CITA =
V_OID_CITA;
   SELECT * INTO V_CITAS FROM CITAS WHERE OID_CITA = V_OID_CITA;
   IF  V_CITAS.DNI_PACIENTE != V_DNI_PACIENTE THEN
     V_SALIDA := FALSE;
   END IF;
   COMMIT;
   DBMS_OUTPUT.PUT_LINE (NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
   EXCEPTION
     WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
       ROLLBACK;
  END ACTUALIZAR;
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_OID_CITA IN
CITAS.OID_CITA%TYPE, SALIDAESPERADA BOOLEAN)
  IS
     V_NUMCITAS NUMBER := 0;
  BEGIN
   DELETE FROM CITAS WHERE OID_CITA = V_OID_CITA;
   SELECT COUNT(*) INTO V_NUMCITAS FROM CITAS WHERE OID_CITA = V_OID_CITA;
   IF V_NUMCITAS != 0 THEN
     V_SALIDA := FALSE;
   END IF;
   COMMIT;
   DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
   EXCEPTION
     WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (FALSE,
SALIDAESPERADA));
       ROLLBACK;
   END ELIMINAR;
  END;
```

```
/

--TABLA HISTORIALES
CREATE OR REPLACE PACKAGE BODY PCK_HISTORIALES
IS
  CURSOR C IS
    SELECT * FROM HISTORIALES;
  V_SALIDA BOOLEAN := TRUE;
  V_HISTORIALES HISTORIALES%ROWTYPE;
PROCEDURE INICIALIZAR
  IS
  BEGIN
    DELETE FROM HISTORIALES;
  END INICIALIZAR;
PROCEDURE CONSULTAR
  IS
  BEGIN
    OPEN C;
    FETCH C INTO V_HISTORIALES;
    DBMS_OUTPUT.PUT_LINE( RPAD('CODIGO:',25) || RPAD('NOMBREPACIENTE:',25)||
RPAD('APELLIDOSPACIENTE:',25)
      || RPAD('USUARIO_MEDICO:',25)|| RPAD('DNI_PACIENTE:',25));
    DBMS_OUTPUT.PUT_LINE(LPAD('-',125,'-'));
    WHILE C%FOUND LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(V_HISTORIALES.CODIGO,25) ||
RPAD(V_HISTORIALES.NOMBREPACIENTE,25)||
RPAD(V_HISTORIALES.APELLIDOSPACIENTE,25)
      || RPAD(V_HISTORIALES.USUARIO_MEDICO,25)||
RPAD(V_HISTORIALES.DNI_PACIENTE,25));
    FETCH C INTO V_HISTORIALES;
    END LOOP;
    CLOSE C;
  END CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2,V_CODIGO IN
HISTORIALES.CODIGO%TYPE, V_NOMBREPACIENTE IN
HISTORIALES.NOMBREPACIENTE%TYPE, V_APELLIDOSPACIENTE IN
HISTORIALES.APELLIDOSPACIENTE%TYPE,
  V_USUARIO_MEDICO IN HISTORIALES.USUARIO_MEDICO%TYPE,V_DNI_PACIENTE
IN HISTORIALES.DNI_PACIENTE%TYPE, SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    INSERT INTO HISTORIALES
(CODIGO,NOMBREPACIENTE,APELLIDOSPACIENTE,USUARIO_MEDICO,DNI_PACIENTE
) VALUES (V_CODIGO, V_NOMBREPACIENTE, V_APELLIDOSPACIENTE,
V_USUARIO_MEDICO, V_DNI_PACIENTE);
    SELECT * INTO V_HISTORIALES FROM HISTORIALES WHERE CODIGO = V_CODIGO;
    IF V_HISTORIALES.CODIGO != V_CODIGO AND V_HISTORIALES.NOMBREPACIENTE
!= V_NOMBREPACIENTE
```

```
     AND V_HISTORIALES.APELLIDOSPACIENTE != V_APELLIDOSPACIENTE AND
V_HISTORIALES.USUARIO_MEDICO != V_USUARIO_MEDICO AND
     V_HISTORIALES.DNI_PACIENTE != V_DNI_PACIENTE  THEN
       V_SALIDA := FALSE;
   END IF;
   COMMIT;
   DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (V_SALIDA,
SALIDAESPERADA));
   EXCEPTION
     WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
       ROLLBACK;
 END INSERTAR;
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2,V_CODIGO IN
HISTORIALES.CODIGO%TYPE, V_NOMBREPACIENTE IN
HISTORIALES.NOMBREPACIENTE%TYPE, V_APELLIDOSPACIENTE IN
HISTORIALES.APELLIDOSPACIENTE%TYPE,
 V_USUARIO_MEDICO IN HISTORIALES.USUARIO_MEDICO%TYPE,V_DNI_PACIENTE
IN HISTORIALES.DNI_PACIENTE%TYPE, SALIDAESPERADA BOOLEAN)
 IS
 BEGIN
   UPDATE HISTORIALES SET  NOMBREPACIENTE = V_NOMBREPACIENTE,
APELLIDOSPACIENTE = V_APELLIDOSPACIENTE, USUARIO_MEDICO =
V_USUARIO_MEDICO,
     DNI_PACIENTE = V_DNI_PACIENTE  WHERE CODIGO = V_CODIGO;
   SELECT * INTO V_HISTORIALES FROM HISTORIALES WHERE CODIGO = V_CODIGO;
   IF V_HISTORIALES.CODIGO != V_CODIGO AND V_HISTORIALES.NOMBREPACIENTE
!= V_NOMBREPACIENTE
     AND V_HISTORIALES.APELLIDOSPACIENTE != V_APELLIDOSPACIENTE AND
V_HISTORIALES.USUARIO_MEDICO != V_USUARIO_MEDICO AND
     V_HISTORIALES.DNI_PACIENTE != V_DNI_PACIENTE THEN
     V_SALIDA := FALSE;
   END IF;
   COMMIT;
   DBMS_OUTPUT.PUT_LINE (NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
   EXCEPTION
     WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
       ROLLBACK;
 END ACTUALIZAR;
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_CODIGO IN
HISTORIALES.CODIGO%TYPE, SALIDAESPERADA BOOLEAN)
 IS
    V_NUMHISTORIALES NUMBER := 0;
 BEGIN
```

```
    DELETE FROM HISTORIALES WHERE CODIGO = V_CODIGO;
    SELECT COUNT(*) INTO V_NUMHISTORIALES FROM HISTORIALES WHERE CODIGO
= V_CODIGO;
    IF V_NUMHISTORIALES != 0 THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (FALSE,
SALIDAESPERADA));
        ROLLBACK;
    END ELIMINAR;
  END;
/


--TABLA LINEASDEHISTORIAL
CREATE OR REPLACE PACKAGE BODY PCK_LINEASDEHISTORIAL
IS
  CURSOR C IS
    SELECT * FROM LINEASDEHISTORIAL;
  V_SALIDA BOOLEAN := TRUE;
  V_LINEASDEHISTORIAL LINEASDEHISTORIAL%ROWTYPE;
PROCEDURE INICIALIZAR
  IS
  BEGIN
    DELETE FROM LINEASDEHISTORIAL;
  END INICIALIZAR;
PROCEDURE CONSULTAR
  IS
  BEGIN
    OPEN C;
    FETCH C INTO V_LINEASDEHISTORIAL;
    DBMS_OUTPUT.PUT_LINE( RPAD('OID_LH:',25) || RPAD('FECHA:',25)||
RPAD('OBSERVACIONES:',200)
      || RPAD('CODIGO_HISTORIAL:',25));
    DBMS_OUTPUT.PUT_LINE(LPAD('-',200,'-'));
    WHILE C%FOUND LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(V_LINEASDEHISTORIAL.OID_LH,25) ||
RPAD(V_LINEASDEHISTORIAL.FECHA,25)||
RPAD(V_LINEASDEHISTORIAL.OBSERVACIONES,200)
      || RPAD(V_LINEASDEHISTORIAL.CODIGO_HISTORIAL,25));
    FETCH C INTO V_LINEASDEHISTORIAL;
    END LOOP;
    CLOSE C;
```

```
  END CONSULTAR;
PROCEDURE INSERTAR(NOMBREPRUEBA VARCHAR2, V_FECHA IN
LINEASDEHISTORIAL.FECHA%TYPE, V_OBSERVACIONES IN
LINEASDEHISTORIAL.OBSERVACIONES%TYPE,
  V_CODIGO_HISTORIAL IN LINEASDEHISTORIAL.CODIGO_HISTORIAL%TYPE,
SALIDAESPERADA BOOLEAN)
  IS
    V_OID_LH LINEASDEHISTORIAL.OID_LH%TYPE;
  BEGIN
    INSERT INTO LINEASDEHISTORIAL (FECHA,OBSERVACIONES,CODIGO_HISTORIAL)
VALUES (V_FECHA, V_OBSERVACIONES, V_CODIGO_HISTORIAL);
    V_OID_LH := SEC_LINEASDEHISTORIAL.CURRVAL;
    SELECT * INTO V_LINEASDEHISTORIAL FROM LINEASDEHISTORIAL WHERE OID_LH
= V_OID_LH;
    IF V_LINEASDEHISTORIAL.FECHA != V_FECHA AND
V_LINEASDEHISTORIAL.OBSERVACIONES != V_OBSERVACIONES
      AND V_LINEASDEHISTORIAL.CODIGO_HISTORIAL != V_CODIGO_HISTORIAL  THEN
        V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END INSERTAR;
PROCEDURE ACTUALIZAR (NOMBREPRUEBA VARCHAR2, V_OID_LH IN
LINEASDEHISTORIAL.OID_LH%TYPE, V_FECHA IN
LINEASDEHISTORIAL.FECHA%TYPE, V_OBSERVACIONES IN
LINEASDEHISTORIAL.OBSERVACIONES%TYPE,
  V_CODIGO_HISTORIAL IN LINEASDEHISTORIAL.CODIGO_HISTORIAL%TYPE,
SALIDAESPERADA BOOLEAN)
  IS
  BEGIN
    UPDATE LINEASDEHISTORIAL SET  FECHA = V_FECHA, OBSERVACIONES =
V_OBSERVACIONES, CODIGO_HISTORIAL = V_CODIGO_HISTORIAL
       WHERE OID_LH = V_OID_LH;
    SELECT * INTO V_LINEASDEHISTORIAL FROM LINEASDEHISTORIAL WHERE OID_LH
= V_OID_LH;
    IF V_LINEASDEHISTORIAL.FECHA != V_FECHA AND
V_LINEASDEHISTORIAL.OBSERVACIONES != V_OBSERVACIONES
      AND V_LINEASDEHISTORIAL.CODIGO_HISTORIAL != V_CODIGO_HISTORIAL   THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
```

```
    DBMS_OUTPUT.PUT_LINE (NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(FALSE,
SALIDAESPERADA));
        ROLLBACK;
  END ACTUALIZAR;
PROCEDURE ELIMINAR (NOMBREPRUEBA VARCHAR2, V_OID_LH IN
LINEASDEHISTORIAL.OID_LH%TYPE, SALIDAESPERADA BOOLEAN)
  IS
      V_NUMLINEASDEHISTORIAL NUMBER := 0;
  BEGIN
    DELETE FROM LINEASDEHISTORIAL WHERE OID_LH = V_OID_LH;
    SELECT COUNT(*) INTO V_NUMLINEASDEHISTORIAL FROM LINEASDEHISTORIAL
WHERE OID_LH = V_OID_LH;
    IF V_NUMLINEASDEHISTORIAL != 0 THEN
      V_SALIDA := FALSE;
    END IF;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS(V_SALIDA,
SALIDAESPERADA));
    EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(NOMBREPRUEBA || ': ' || ASSERT_EQUALS (FALSE,
SALIDAESPERADA));
        ROLLBACK;
    END ELIMINAR;
  END;
/



--PRUEBA DE PAQUETES
SET SERVEROUTPUT ON;
--TABLA CLINICAS
BEGIN
PCK_CLINICAS.INICIALIZAR;
PCK_CLINICAS.INSERTAR('INSERTAR UNA CLINICA', 'Clínica CareMujer La Paz','Avda.
Luis Montoto, 81 41018 Sevilla (España)',TRUE);
PCK_CLINICAS.INSERTAR('INSERTAR UNA CLINICA', 'Clínica CareMujer Santa
Isabel','C/Rafael Salgado, 3 41013 – Sevilla (España)',TRUE);
PCK_CLINICAS.INSERTAR('INSERTAR UNA CLINICA', 'Clínica CareMujer Los
Olivos','C/Almendralejo, 3 41015 – Sevilla (España)',TRUE);
PCK_CLINICAS.INSERTAR('INSERTAR UNA CLINICA CON NOMBRE NULL',
NULL,'C/Rafael Salgado, 3 41013 – Sevilla (España)',FALSE);
PCK_CLINICAS.INSERTAR('INSERTAR UNA CLINICA CON DIRECCION NULL', 'Clínica
CareMujer Santa Isabel',NULL,FALSE);
```

```
PCK_CLINICAS.ACTUALIZAR('ACTUALIZAR UNA CLINICA',3, 'Clínica CareMujer Los
OlivosActualizado','C/Águila,1 41015 – Sevilla (España)',TRUE);
PCK_CLINICAS.ACTUALIZAR('ACTUALIZAR UNA CLINICA CON NOMBRE NULL',3,
NULL,'C/Rafael Salgado, 3 41013 – Sevilla (España)',FALSE);
PCK_CLINICAS.ACTUALIZAR('ACTUALIZAR UNA CLINICA CON DIRECCION NULL',3,
'HOLA',NULL,FALSE);
PCK_CLINICAS.ELIMINAR('ELMINAR UNA CLINICA',3,TRUE);
PCK_CLINICAS.CONSULTAR;
END;
/
```

--TABLA RECEPCIONISTAS

```
BEGIN
PCK_RECEPCIONISTAS.INICIALIZAR;
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA',
'jaisanfor','123456','Jaime', 'Sánchez
Forte','47852624S','jaisanfor@caremujer.es','954786925',TRUE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA',
'estrodbar','asdfjklñ','Estrella', 'Rodríguez
Barahona','78982564X','estrodbar@caremujer.es','954786926',TRUE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA',
'cationcli','cata19','Catalina', 'Ionel
Clim','45878482E','cationcli@caremujer.es','954786927',TRUE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA CONTRASEÑA
NULL', 'cationcli',NULL,'Catalina', 'Ionel
Clim','45878482E','cationcli@caremujer.es','954786927',FALSE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA NOMBRE NULL',
'cationcli','cata19',NULL, 'Ionel
Clim','45878482E','cationcli@caremujer.es','954786927',FALSE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA APELLIDOS NULL',
'cationcli','cata19','Catalina', NULL,'45878482E','cationcli@caremujer.es','954786927',FALSE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA DNI NULL',
'cationcli','cata19','Catalina', 'Ionel Clim',NULL,'cationcli@caremujer.es','954786927',FALSE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA EMAIL NULL',
'cationcli','cata19','Catalina', 'Ionel Clim','45878482E',NULL,'954786927',FALSE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA TELEFONO NULL',
'cationcli','cata19','Catalina', 'Ionel Clim','45878482E','cationcli@caremujer.es',NULL,FALSE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA VIOLANDO PK',
'jaisanfor','jjjj','Jaime', 'Santos
Forlán','15469878S','jaisanfor@caremujer.es','954786927',FALSE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA VIOLANDO AK',
'paugorbar','pauli','Paula', 'Gordillo
Bardón','47852624S','paugorbar@caremujer.es','954587896',FALSE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA VIOLANDO
TELEFONO', 'test','test','test',
'test','11111111A','test@caremujer.es','9547869274646',FALSE);
```

```
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA VIOLANDO DNI',
'test','cata19','Catalina', 'Ionel
Clim','45878482456E','cationcli@caremujer.es','954786927',FALSE);
PCK_RECEPCIONISTAS.INSERTAR('INSERTAR UN RECEPCIONISTA VIOLANDO RN4',
'test2','cata19','Catalina', 'Ionel Clim','58888888L','cationcli@hotmail.es','954786927',FALSE);
PCK_RECEPCIONISTAS.ACTUALIZAR('ACTUALIZAR UN RECEPCIONISTA',
'cationcli','ASDFA','Actualizado', 'test
test','45878482E','cationcli@caremujer.es','954786927',TRUE);
PCK_RECEPCIONISTAS.ELIMINAR('ELIMINAR UN RECEPCIONISTA','cationcli',TRUE);
PCK_RECEPCIONISTAS.CONSULTAR;
END;
/

--TABLA MEDICOS
BEGIN
PCK_MEDICOS.INICIALIZAR;
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO','anacabmor','anac','Ana','Cabanillas
Mora','43286217D','anacabmor@caremujer.es','GINECÓLOGO',23,TRUE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO','carortcam','camachin','Carlos','Ortiz
Camacho','63281975A','carortcam@caremujer.es','EMBRIÓLOGO',23,TRUE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO','lucceslag','lucy88','Lucía','Cespedes
Lagos','22587946X','lucceslag@caremujer.es','ONCÓLOGO',23,TRUE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO','javbelmat','jbm','Javier','Beltrán
Matamoros','45878689T','javbelmat@caremujer.es','GINECÓLOGO',23,TRUE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO','felolizan','qwer','Felipe','Oliveira
Zanutti','23584999P','felolizan@caremujer.es','EMBRIÓLOGO',23,TRUE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO','belbardur','qwerty','Belén','Barahona
Durán','55976235K','belbardur@caremujer.es','ONCÓLOGO',23,TRUE);
PCK_MEDICOS.INSERTAR('INSERTAR UN
MEDICO','test','test','test','test','45654565U','test@caremujer.es','GINECÓLOGO',23,TRUE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO VIOLANDO
DNI','a','anac','Ana','Cabanillas
Mora','432862517D','anacabmor@caremujer.es','GINECÓLOGO',23,FALSE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO VIOLANDO
EMAIL','b','anac','Ana','Cabanillas
Mora','43286251I','anacabmor@hotmail.es','GINECÓLOGO',23,FALSE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO VIOLANDO
ESPECIALIDAD','c','anac','Ana','Cabanillas
Mora','43286251O','anacabmor@caremujer.es','OCULISTA',28,FALSE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO VIOLANDO
PK','anacabmor','galsdf','Ana','Cabanillas
Mora','48987865P','anacabmor@caremujer.es','GINECÓLOGO',28,FALSE);
PCK_MEDICOS.INSERTAR('INSERTAR UN MEDICO VIOLANDO
AK','petisa','fafd','Anastasia','Autukoba
Larisa','43286217D','anastasia@caremujer.es','GINECÓLOGO',28,FALSE);
PCK_MEDICOS.ACTUALIZAR('ACTUALIZAR UN
MEDICO','test','test2','test2','test2','45654565U','test@caremujer.es','GINECÓLOGO',28,TRUE
);
```

```
PCK_MEDICOS.ELIMINAR('ELIMINAR UN MEDICO','test',TRUE);
PCK_MEDICOS.CONSULTAR;
END;
/

--TABLA TRABAJADORES
BEGIN
PCK_TRABAJADORES.INICIALIZAR;
PCK_TRABAJADORES.INSERTAR('INSERTAR UN
TRABAJADOR','45878583A','Margarita','Pérez Núñez','marpernuñ@caremujer.es',1,TRUE);
PCK_TRABAJADORES.INSERTAR('INSERTAR UN
TRABAJADOR','45456982F','Lorenzo','Peral Márquez','lorpermar@caremujer.es',1,TRUE);
PCK_TRABAJADORES.INSERTAR('INSERTAR UN
TRABAJADOR','25476969B','Elena','Ramírez Cano','eleramcan@caremujer.es',2,TRUE);
PCK_TRABAJADORES.INSERTAR('INSERTAR UN
TRABAJADOR','66123849Y','Manuel','Gómez
Izquierdo','mangomizq@caremujer.es',2,TRUE);
PCK_TRABAJADORES.INSERTAR('INSERTAR UN
TRABAJADOR','12121212E','test','test','test@caremujer.es',1,TRUE);
PCK_TRABAJADORES.INSERTAR('INSERTAR UN TRABAJADOR VIOLANDO
DNI','4587858322A','mal','MAL MAL','mal@caremujer.es',1,FALSE);
PCK_TRABAJADORES.INSERTAR('INSERTAR UN TRABAJADOR VIOLANDO
EMAIL','45878528H','Margarita','Pérez Núñez','emailmal@hotmail.com',1,FALSE);
PCK_TRABAJADORES.INSERTAR('INSERTAR UN TRABAJADOR VIOLANDO
PK','45878583A','MAL','MAL','MAL@caremujer.es',1,FALSE);
PCK_TRABAJADORES.ACTUALIZAR('ACTUALIZAR UN
TRABAJADAOR','12121212E','test2','test2','test2@caremujer.es',1,TRUE);
PCK_TRABAJADORES.ELIMINAR('ELIMINAR UN TRABAJADOR','12121212E',TRUE);
PCK_TRABAJADORES.CONSULTAR;
END;
/

--TABLA TRABAJAEN
BEGIN
PCK_TRABAJAEN.INICIALIZAR;
PCK_TRABAJAEN.INSERTAR('INSERTAR UNA ENTRADA','anacabmor',1,TRUE);
PCK_TRABAJAEN.INSERTAR('INSERTAR UNA ENTRADA','anacabmor',2,TRUE);
PCK_TRABAJAEN.INSERTAR('INSERTAR UNA ENTRADA','javbelmat',1,TRUE);
PCK_TRABAJAEN.INSERTAR('INSERTAR UNA ENTRADA','javbelmat',2,TRUE);
PCK_TRABAJAEN.INSERTAR('INSERTAR UNA ENTRADA','carortcam',1,TRUE);
PCK_TRABAJAEN.INSERTAR('INSERTAR UNA ENTRADA','lucceslag',1,TRUE);
PCK_TRABAJAEN.INSERTAR('INSERTAR UNA ENTRADA','felolizan',2,TRUE);
PCK_TRABAJAEN.INSERTAR('INSERTAR UNA ENTRADA','belbardur',2,TRUE);
PCK_TRABAJAEN.INSERTAR('INSERTAR UNA ENTRADA','belbardur',1,TRUE);
PCK_TRABAJAEN.ELIMINAR('ELIMINAR UNA ENTRADA',9,TRUE);
PCK_TRABAJAEN.CONSULTAR;
END;
/
```

```
--TABLA PACIENTES
BEGIN
PCK_PACIENTES.INICIALIZAR;
PCK_PACIENTES.INSERTAR('INSERTAR UN PACIENTE','43462819W','mjdl', 'María
Jesús','Domínguez Luz','mjdominguez@gmail.com',123456,'estrodbar',TRUE);
PCK_PACIENTES.INSERTAR('INSERTAR UN PACIENTE','88888888O','cdrm',
'Concepción','De la Rosa Martín','laconceploco@gmail.com',111111,'estrodbar',TRUE);
PCK_PACIENTES.INSERTAR('INSERTAR UN PACIENTE','58962819P','111',
'Rosario','Cordero López','charo@gmail.com',222222,'estrodbar',TRUE);
PCK_PACIENTES.INSERTAR('INSERTAR UN PACIENTE','78531564E','123',
'Inmaculada','Macho Becerra','macu@gmail.com',123587,'jaisanfor',TRUE);
PCK_PACIENTES.INSERTAR('INSERTAR UN PACIENTE','44445555Q',
'456','Josefina','Pérez García','josefayaya@hotmail.com',987654,'jaisanfor',TRUE);
PCK_PACIENTES.INSERTAR('INSERTAR UN PACIENTE','42262819L','789',
'Esmeralda','Diamantina Clementa','apicar@gmail.com',285396,'jaisanfor',TRUE);
PCK_PACIENTES.INSERTAR('INSERTAR UN
PACIENTE','11111111A','jho','TEST','TEST','TEST@GMAIL.COM',224455,'jaisanfor',TRUE);
PCK_PACIENTES.INSERTAR('INSERTAR UN
PACIENTE','11111111B','jho','TEST','TEST','TEST@GMAIL.COM',224457,'estrodbar',TRUE);
PCK_PACIENTES.INSERTAR('INSERTAR UN PACIENTE','45555519L',
'jho','test','tes','test@gmail.com',287946,'jaisanfor',TRUE);
PCK_PACIENTES.ACTUALIZAR('ACTUALIZAR UN PACIENTE','45555519L','jho',
'test2','tes2','test2@gmail.com',587946,'jaisanfor',TRUE);
PCK_PACIENTES.ELIMINAR('ELIMINAR UN PACIENTE','45555519L',TRUE);
PCK_PACIENTES.INSERTAR('INSERTAR UN PACIENTE VIOLANDO
PK','78531564E','mjdl',
'PETISA','PETISUI','PETISAmacu@gmail.com',123787,'jaisanfor',FALSE);
PCK_PACIENTES.INSERTAR('INSERTAR UN PACIENTE VIOLANDO
DNI','7853156544E','mjdl', 'Inmaculada','Macho
Becerra','macu@gmail.com',123587,'jaisanfor',FALSE);
PCK_PACIENTES.CONSULTAR;
END;
/
--TABLA HISTORIALES
BEGIN
PCK_HISTORIALES.INICIALIZAR;
PCK_HISTORIALES.INSERTAR('INSERTAR UN HISTORIAL','000001','María
Jesús','Domínguez Luz','anacabmor','43462819W',TRUE);
PCK_HISTORIALES.INSERTAR('INSERTAR UN HISTORIAL','000002','Concepción','De la
Rosa Martín','javbelmat','88888888O',TRUE);
PCK_HISTORIALES.INSERTAR('INSERTAR UN HISTORIAL','000003','Rosario','Cordero
López','carortcam','58962819P',TRUE);
PCK_HISTORIALES.INSERTAR('INSERTAR UN HISTORIAL','000004','Inmaculada','Macho
Becerra','lucceslag','78531564E',TRUE);
PCK_HISTORIALES.INSERTAR('INSERTAR UN HISTORIAL','000005','Josefina','Pérez
García','felolizan','44445555Q',TRUE);
```

```
PCK_HISTORIALES.INSERTAR('INSERTAR UN
HISTORIAL','000006','Esmeralda','Diamantina Clementa','belbardur','42262819L',TRUE);
PCK_HISTORIALES.INSERTAR('INSERTAR UN
HISTORIAL','000007','Esmeralda','Diamantina Clementa','belbardur','42262819L',TRUE);
PCK_HISTORIALES.ACTUALIZAR('ACTUALIZAR UN
HISTORIAL','000007','Esmeraldinha','Diamantina
Clementaniha','belbardur','42262819L',TRUE);
PCK_HISTORIALES.ELIMINAR('ELIMINAR UN HISTORIAL','000007',TRUE);
PCK_HISTORIALES.INSERTAR('INSERTAR UN HISTORIAL VIOLANDO
PK','000001','Esmeralda','Diamantina Clementa','belbardur','42262819L',FALSE);
PCK_HISTORIALES.INSERTAR('INSERTAR UN HISTORIAL VIOLANDO
CODIGO','000000007','Esmeralda','Diamantina Clementa','belbardur','42262819L',FALSE);
PCK_HISTORIALES.CONSULTAR;
END;
/

--TABLA LINEASDEHISTORIAL
BEGIN
PCK_LINEASDEHISTORIAL.INICIALIZAR;
PCK_LINEASDEHISTORIAL.INSERTAR('INSERTAR UNA
LINEADEHISTORIAL',TO_DATE('03/07/2017', 'DD/MM/YYYY'),'Revisión realizada con éxito,
estado perfecto','000001',TRUE);
PCK_LINEASDEHISTORIAL.INSERTAR('INSERTAR UNA
LINEADEHISTORIAL',TO_DATE('07/08/2017', 'DD/MM/YYYY'),'Aparición de lunar sin previo
aviso, nada preocupante, acudir si cambia tamaño o aspecto','000001',TRUE);
PCK_LINEASDEHISTORIAL.INSERTAR('INSERTAR UNA
LINEADEHISTORIAL',TO_DATE('15/08/2017', 'DD/MM/YYYY'),'Revisión tras 3 meses de
embarazo, estado de gestación perfecto, acudir nuevamente en 2 meses','000002',TRUE);
PCK_LINEASDEHISTORIAL.INSERTAR('INSERTAR UNA
LINEADEHISTORIAL',TO_DATE('15/08/2017', 'DD/MM/YYYY'),'Revisión post-operatoria,
sigue en perfecto estado','000003',TRUE);
PCK_LINEASDEHISTORIAL.INSERTAR('INSERTAR UNA
LINEADEHISTORIAL',TO_DATE('20/08/2017', 'DD/MM/YYYY'),'Revisión realizada con éxito,
estado perfecto','000004',TRUE);
PCK_LINEASDEHISTORIAL.INSERTAR('INSERTAR UNA
LINEADEHISTORIAL',TO_DATE('21/08/2017', 'DD/MM/YYYY'),'Infección localizada, recetado
tratamiento antibiótico durante 1 semana, tomar anti-inflamatorio si presenta
dolor','000005',TRUE);
PCK_LINEASDEHISTORIAL.INSERTAR('INSERTAR UNA
LINEADEHISTORIAL',TO_DATE('03/09/2017', 'DD/MM/YYYY'),'Revisión realizada con éxito,
estado perfecto','000006',TRUE);
PCK_LINEASDEHISTORIAL.INSERTAR('INSERTAR UNA
LINEADEHISTORIAL',TO_DATE('03/09/2017', 'DD/MM/YYYY'),'test','000001',TRUE);
PCK_LINEASDEHISTORIAL.ACTUALIZAR('ACTUALIZAR UNA
LINEADEHISTORIAL',8,TO_DATE('03/09/2017', 'DD/MM/YYYY'),'test2','000001',TRUE);
PCK_LINEASDEHISTORIAL.ELIMINAR('ELIMINAR UNA LINEADEHISTORIAL',8,TRUE);
PCK_LINEASDEHISTORIAL.INSERTAR('INSERTAR UNA LINEADEHISTORIAL VIOLANDO
CODIGO',TO_DATE('03/09/2017', 'DD/MM/YYYY'),'test','000000001',FALSE);
```

```
PCK_LINEASDEHISTORIAL.CONSULTAR;
END;
/

--TABLA CITAS
BEGIN
PCK_CITAS.INICIALIZAR;
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);

PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);

PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);

PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);

PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor','11111111A',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),1,'anacabmor','jaisanfor',NULL,TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('04/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);

PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('05/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);

PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('06/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);

PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('07/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);

PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','09:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','10:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','11:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','12:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','13:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','16:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','17:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
```

```
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:20',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','18:40',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',TRUE);


PCK_CITAS.INSERTAR('INSERTAR UNA CITA','19:00',TO_DATE('11/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,TRUE);
PCK_CITAS.ACTUALIZAR('ACTUALIZAR UNA CITA',231,'11111111B',TRUE);
PCK_CITAS.ELIMINAR('ELIMINAR UNA CITA',231,TRUE);


PCK_CITAS.INSERTAR('INSERTAR UNA CITA VIOLANDO RN1 CITAS CADA
20M','17:55',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',FALSE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA VIOLANDO RN1 CITAS CADA
20M','17:50',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',FALSE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA VIOLANDO RN1 CITAS CADA
20M','17:45',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',FALSE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA VIOLANDO RN1 CITAS CADA
20M','17:15',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',FALSE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA VIOLANDO RN2 HORARIO DE LA
CLINICA','08:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',FALSE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA VIOLANDO RN2 HORARIO DE LA
CLINICA','20:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',FALSE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA VIOLANDO RN2 HORARIO DE LA
CLINICA','14:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',FALSE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA VIOLANDO RN3 CITA
IDÉNTICA','19:00',TO_DATE('08/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar','11111111B',FALSE);
PCK_CITAS.INSERTAR('INSERTAR UNA CITA VIOLANDO RN5 CITAS DÍAS DE
DIARIO','19:00',TO_DATE('10/09/2017',
'DD/MM/YYYY'),2,'javbelmat','estrodbar',NULL,FALSE);
PCK_CITAS.CONSULTAR;

END;
/
```

```
--PRUEBAS DE PROCEDIMIENTOS Y FUNCIONES ASOCIADOS A RF
--RF-001
EXEC PR_TRABAJADORES_CLINICA;

--RF-002
EXEC PR_PACIENTES_CLINICA(1);

--RF-003
EXEC PR_CITAS_DISPONIBLES(TO_DATE('08/09/2017', 'DD/MM/YYYY'),1,'anacabmor');

--RF-004
EXEC PR_CITAS_NO_DISPONIBLES(TO_DATE('08/09/2017', 'DD/MM/YYYY'),1);

--RF-005
EXEC PR_CITAS_MEDICO('anacabmor',TO_DATE('05/09/2017', 'DD/MM/YYYY'));

--RF-006
EXEC PR_MEDICOS_CLINICA;

--RF-007
EXEC PR_MEDICOS_ESPECIALIDAD (1);

--RF-008
EXEC PR_HISTORIALES ('43462819W');

--RF-009
BEGIN
  DBMS_OUTPUT.PUT_LINE(FN_SEGURO_PACIENTE('43462819W'));
END;
/

--RF-010
BEGIN
  DBMS_OUTPUT.PUT_LINE(FN_CITA_MAS_TARDE('anacabmor',TO_DATE('04/09/2017',
'DD/MM/YYYY')));
END;
/

--RF-011
EXEC PR_MEDICO_SUPERVISA_HISTORIAL;
```