

Semana 5

Modelamiento de Bases de Datos

Formato de respuesta

| **Nombre estudiante:** | Nicolás Rivera |
| --- | --- |
| **Asignatura:** | **Carrera:** |
| **Profesor:** | **Fecha: 15/09/25** |

# Descripción de la actividad

En esta quinta semana, realizarás una actividad sumativa individual llamada "Transformando el MER en Modelo Relacional", donde deberás dar solución al caso de negocio planteado, desarrollando un Modelo Entidad-Relación-Extendido (MER-E) Normalizado. En esta actividad, será necesario que:

* Identifiques todas las entidades, los supertipos y subtipos con sus atributos identificadores, atributos obligatorios y opcionales.
* Identifiques las relaciones entre las entidades.
* Analices y determines los tipos de datos más adecuados para cada atributo, así como su dominio.
* Finalmente transformar a Modelo Relacional Normalizado.

## Instrucciones específicas

Para llevar a cabo la actividad sumativa de la semana, a continuación te presentaremos el  
contexto de negocio que deberás analizar en detalle:

### Contexto: Centro Médico Universitario San Rafael

El Centro Médico Universitario San Rafael es una clínica ambulatoria ubicada en el campus universitario, que ofrece servicios de medicina general, salud mental y laboratorio clínico a estudiantes, personal académico y administrativo. Su objetivo es entregar atención médica rápida y de calidad sin que los pacientes deban salir del entorno universitario.

El centro cuenta con tres unidades de atención (Medicina General, Salud Mental y Laboratorio Clínico). Cada unidad tiene su propio equipo médico y equipamiento, pero todas comparten un sistema centralizado de registros clínicos.

Los pacientes pueden agendar consultas presenciales o virtuales. Cada consulta es realizada por un médico y puede derivar a exámenes de laboratorio o a otra especialidad. En toda atención deben registrarse: paciente, médico tratante, fecha y hora, especialidad, diagnóstico y, si corresponde, exámenes solicitados.

Los médicos están asociados a una unidad específica y pueden atender múltiples consultas. Cada médico se registra con RUT, nombre, fecha de ingreso y especialidad. De manera opcional, puede establecerse una relación de supervisión entre médicos (por motivos académicos), indicando quién supervisa o es supervisado.

Los pacientes se registran con nombre, RUT, tipo de usuario (estudiante, funcionario, externo) y datos de contacto. Pueden tener historial de atenciones anteriores, el cual debe mantenerse con confidencialidad y trazabilidad.

Los exámenes de laboratorio se registran con un código, nombre, tipo de muestra y condiciones de preparación (ej.: ayuno). Pueden ser solicitados por distintos médicos, y los resultados se ingresan al sistema una vez completados, vinculando paciente, médico y solicitud.

Actualmente, la información se gestiona en una hoja de cálculo donde cada fila representa una atención completa. Esta estructura plana provoca duplicación de datos, errores de ingreso y dificultades en el seguimiento clínico.

**Como parte del equipo de implementación, deberás:**

* Analizar la información disponible en las planillas.
* Modificar el MER de la Figura 1 y normalizarlo aplicando las tres primeras formas normales para obtener un Modelo Entidad-Relación Extendido (MER-E).
* Transformar el MER-E en un Modelo Relacional (MR), definiendo tablas, claves primarias, claves foráneas, restricciones y relaciones.
* Generar el script DDL en Oracle SQL para crear las tablas del modelo físico.

**Figura 1**

*Modelo MER incompleto que se debe normalizar y convertir a MR*

*Nota*. En la figura se presenta un modelo de datos desnormalizado que ilustra las relaciones entre las entidades: ATENCION\_MEDICA, MEDICO, PACIENTE, PAGO\_ATENCION y ESPECIALIDAD. Oracle. (s.f.). *Oracle SQL Developer Data Modeler* [Software]. [Oracle.com](https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/)

### Reglas de Negocio:

* El centro médico atiende a estudiantes, funcionarios y externos, registrando cada atención en una ficha médica.
* Cada atención incluye: fecha, tipo (general, urgencia, preventiva), paciente, médico y monto pagado.
* Un paciente puede tener múltiples atenciones, pero cada atención es realizada por un único médico.
* Todos los pacientes deben registrarse con: RUT, nombre completo, sexo, fecha de nacimiento, dirección y comuna. Cada paciente se asocia a una comuna, y cada comuna pertenece a una región.
* Cada médico pertenece al equipo del centro y se registra con: RUT, nombre completo, fecha de ingreso y especialidad. Un médico tiene solo una especialidad.
* Una especialidad puede estar asociada a varios médicos (ej.: Medicina General, Ginecología, Psicología).
* Toda atención debe registrarse con su pago, indicando: monto cancelado, tipo de pago (efectivo, tarjeta, convenio) y la atención asociada.
* Cada médico está afiliado a una AFP y a una institución de salud (Fonasa o Isapre). Estas entidades solo aplican a médicos, no a pacientes.
* Una misma AFP o institución de salud puede estar vinculada a varios médicos; ambas se registran con nombre y código.
* La ficha de atención es la entidad central que vincula paciente, médico y pago.

Más detalle se muestra en las siguientes vistas de usuario:

**Figura 2**

*Vista de Usuario: Bono Fonasa*

El bono FONASA es un documento rectangular con fondo blanco y encabezado azul. En la parte superior izquierda se lee en letras grandes y blancas “BONO FONASA”. A la derecha, en el mismo encabezado, se muestra el texto “Monto pagado” seguido del valor $12.700 en números grandes.

Debajo, en un recuadro central, se indica el prestador: Centro Médico Universitario. Luego se presenta información detallada como el folio 830142229, la fecha 10-04-2024, el RUT del paciente 14567890-2, el nombre Carla Rodriguez, el tipo de atención Medicina General, el médico Roberto González con su RUT 14567890-2, y que el tipo de pago fue por convenio.

*Nota.* En la figura se muestra un formato de un Bono de Fonasa.

**Figura 3**

*Vista de Usuario: Agendamiento de hora*La imagen contiene un bono Fonasa con fondo blanco y encabezado azul. En la parte superior, en letras grandes blancas, aparece el título “BONO FONASA”, junto al texto “Monto pagado: $12.700” en color blanco y el valor en grande con números azules.
Debajo, centrado, dice “CENTRO MÉDICO UNIVERSITARIO”.
A continuación, se listan los siguientes campos y valores en texto negro sobre fondo blanco:

Folio: 830142229

Fecha: 10-04-2024

Rut paciente: 14567890-2

Nombre paciente: Carla Rodriguez

Tipo de atención: MEDICINA GENERAL

Médico: Roberto González

Rut médico: 14567890-2

Tipo de pago: Convenio

*Nota.* En la figura se muestra un formato de agendamiento de hora médica.

## Instrucciones de entrega

**Paso 1:** Para llevar a cabo este proceso, tendrás que utilizar la herramienta Oracle SQL Data Modeler, disponible de descarga a través del siguiente enlace:

[https://www.oracle.com/database/sqldeveloper/technologies/sql-data-modeler/download/](https://www.oracle.com/database/sqldeveloper/technologies/sql-data-modeler/download/%E2%80%AF)

Cuando tengas lista tu solución, deberás adjuntar en este documento dos capturas de:

1. Modelo Entidad-Relación-Extendido (MER-E) Normalizado solicitado en notación Barker.
2. Modelo Relacional (MR) Normalizado.
3. Script DDL generado con la herramienta Oracle Data Modeler.

**Ejemplos:**

|  |  |  |
| --- | --- | --- |
| Modelo MER-E Normalizado en notación Barker | Modelo Relacional Normalizado | Script DDL |
|  |  |  |

Adjunta tus evidencias en esta parte:

**Modelo Entidad Relación-Extendido Normalizado (MER-E) en notación Barker:**

A diagram of a company

Description automatically generated

**Modelo Relacional (MR) Normalizado:**

A diagram of a computer

Description automatically generated with medium confidence

**Script obtenido con la herramienta, a partir de tu Modelo Relacional**

-- Generated by Oracle SQL Developer Data Modeler 24.3.1.351.0831

-- at: 2025-09-15 23:07:02 CLST

-- site: Oracle Database 11g

-- type: Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO\_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE AFP

(

id\_afp INTEGER NOT NULL ,

nombre CHAR (50 CHAR) NOT NULL

)

;

ALTER TABLE AFP

ADD CONSTRAINT AFP\_PK PRIMARY KEY ( id\_afp ) ;

CREATE TABLE ATENCION\_DIAGNOSTICO

(

ATENCION\_MEDICA\_id\_atencion INTEGER NOT NULL ,

DIAGNOSTICO\_id\_diagnostico INTEGER NOT NULL

)

;

ALTER TABLE ATENCION\_DIAGNOSTICO

ADD CONSTRAINT ATENCION\_DIAGNOSTICO PRIMARY KEY ( ATENCION\_MEDICA\_id\_atencion, DIAGNOSTICO\_id\_diagnostico ) ;

CREATE TABLE ATENCION\_MEDICA

(

id\_atencion INTEGER NOT NULL ,

id\_medico INTEGER NOT NULL ,

fecha\_atencion DATE NOT NULL ,

hora\_atencion DATE NOT NULL ,

tipo CHAR (50 CHAR) NOT NULL ,

monto\_pagado INTEGER NOT NULL ,

PACIENTE\_rut CHAR

-- WARNING: CHAR size not specified

NOT NULL ,

PACIENTE\_id\_paciente INTEGER NOT NULL ,

PAGO\_id\_pago INTEGER NOT NULL

)

;

CREATE UNIQUE INDEX ATENCION\_MEDICA\_\_IDX ON ATENCION\_MEDICA

(

PAGO\_id\_pago ASC

)

;

ALTER TABLE ATENCION\_MEDICA

ADD CONSTRAINT ATENCION\_MEDICA\_PK PRIMARY KEY ( id\_atencion ) ;

CREATE TABLE ATENCION\_MEDICO

(

ATENCION\_MEDICA\_id\_atencion INTEGER NOT NULL ,

MEDICO\_rut CHAR (13 CHAR) NOT NULL ,

MEDICO\_id\_medico INTEGER NOT NULL

)

;

ALTER TABLE ATENCION\_MEDICO

ADD CONSTRAINT ATENCION\_MEDICO PRIMARY KEY ( ATENCION\_MEDICA\_id\_atencion, MEDICO\_rut, MEDICO\_id\_medico ) ;

CREATE TABLE COMUNA

(

id\_comuna INTEGER NOT NULL ,

nombre CHAR (50 CHAR) NOT NULL ,

REGION\_id\_region INTEGER NOT NULL

)

;

ALTER TABLE COMUNA

ADD CONSTRAINT COMUNA\_PK PRIMARY KEY ( id\_comuna ) ;

CREATE TABLE DIAGNOSTICO

(

id\_diagnostico INTEGER NOT NULL ,

nombre CHAR (100 CHAR) NOT NULL

)

;

ALTER TABLE DIAGNOSTICO

ADD CONSTRAINT DIAGNOSTICO\_PK PRIMARY KEY ( id\_diagnostico ) ;

CREATE TABLE DIRECCION

(

id\_direccion INTEGER NOT NULL ,

calle CHAR (100 CHAR) NOT NULL ,

numero\_calle INTEGER ,

COMUNA\_id\_comuna INTEGER NOT NULL

)

;

ALTER TABLE DIRECCION

ADD CONSTRAINT DIRECCION\_PK PRIMARY KEY ( id\_direccion ) ;

CREATE TABLE EFECTIVO

(

id\_pago INTEGER NOT NULL

)

;

ALTER TABLE EFECTIVO

ADD CONSTRAINT EFECTIVO\_PK PRIMARY KEY ( id\_pago ) ;

CREATE TABLE ESPECIALIDAD

(

id\_especialidad INTEGER NOT NULL ,

nombre CHAR (100 CHAR) NOT NULL

)

;

ALTER TABLE ESPECIALIDAD

ADD CONSTRAINT ESPECIALIDAD\_PK PRIMARY KEY ( id\_especialidad ) ;

CREATE TABLE EXAMENES\_LABORATORIO

(

id\_examen INTEGER NOT NULL ,

codigo CHAR (4 CHAR) NOT NULL ,

nombre CHAR (50 CHAR) NOT NULL ,

tipo\_muestra CHAR (50 CHAR) NOT NULL ,

preparacion CHAR (50 CHAR) NOT NULL

)

;

ALTER TABLE EXAMENES\_LABORATORIO

ADD CONSTRAINT EXAMENES\_LABORATORIO\_PK PRIMARY KEY ( id\_examen ) ;

CREATE TABLE INSTITUCION\_SALUD

(

id\_institucion INTEGER NOT NULL ,

nombre CHAR (50 CHAR) NOT NULL ,

tipo CHAR (50 CHAR) NOT NULL

)

;

ALTER TABLE INSTITUCION\_SALUD

ADD CONSTRAINT INSTITUCION\_SALUD\_PK PRIMARY KEY ( id\_institucion ) ;

CREATE TABLE MEDICO

(

id\_medico INTEGER NOT NULL ,

rut CHAR (13 CHAR) NOT NULL ,

nombre CHAR (50 CHAR) NOT NULL ,

fecha\_ingreso DATE NOT NULL ,

especialidad CHAR (50 CHAR) NOT NULL ,

ESPECIALIDAD\_id\_especialidad INTEGER NOT NULL ,

MEDICO\_rut CHAR (13 CHAR) ,

MEDICO\_id\_medico INTEGER ,

AFP\_id\_afp INTEGER NOT NULL ,

-- ERROR: Column name length exceeds maximum allowed length(30)

INSTITUCION\_SALUD\_id\_institucion INTEGER NOT NULL

)

;

ALTER TABLE MEDICO

ADD CONSTRAINT MEDICO\_PK PRIMARY KEY ( rut, id\_medico ) ;

CREATE TABLE PACIENTE

(

id\_paciente INTEGER NOT NULL ,

rut CHAR

-- WARNING: CHAR size not specified

NOT NULL ,

tipo\_usuario CHAR (50 CHAR) NOT NULL ,

nombre CHAR (50 CHAR) NOT NULL ,

sexo CHAR (10 CHAR) NOT NULL ,

fecha\_nacimiento DATE NOT NULL ,

numero\_contacto CHAR (9 CHAR) NOT NULL ,

diagnostico CHAR (50 CHAR)

)

;

ALTER TABLE PACIENTE

ADD CONSTRAINT PACIENTE\_PK PRIMARY KEY ( rut, id\_paciente ) ;

CREATE TABLE PAGO

(

id\_pago INTEGER NOT NULL ,

ATENCION\_MEDICA\_id\_atencion INTEGER NOT NULL

)

;

CREATE UNIQUE INDEX PAGO\_\_IDX ON PAGO

(

ATENCION\_MEDICA\_id\_atencion ASC

)

;

ALTER TABLE PAGO

ADD CONSTRAINT PAGO\_PK PRIMARY KEY ( id\_pago ) ;

CREATE TABLE REGION

(

id\_region INTEGER NOT NULL ,

nombre CHAR (100 CHAR) NOT NULL

)

;

ALTER TABLE REGION

ADD CONSTRAINT REGION\_PK PRIMARY KEY ( id\_region ) ;

CREATE TABLE Relation\_6

(

DIRECCION\_id\_direccion INTEGER NOT NULL ,

PACIENTE\_rut CHAR

-- WARNING: CHAR size not specified

NOT NULL ,

PACIENTE\_id\_paciente INTEGER NOT NULL

)

;

ALTER TABLE Relation\_6

ADD CONSTRAINT Relation\_6\_PK PRIMARY KEY ( DIRECCION\_id\_direccion, PACIENTE\_rut, PACIENTE\_id\_paciente ) ;

CREATE TABLE SOLICITUD\_EXAMEN

(

id\_solicitud INTEGER NOT NULL ,

fecha\_solicitud DATE NOT NULL ,

estado CHAR (50 CHAR) NOT NULL ,

ATENCION\_MEDICA\_id\_atencion INTEGER NOT NULL ,

EXAMENES\_LABORATORIO\_id\_examen INTEGER NOT NULL

)

;

ALTER TABLE SOLICITUD\_EXAMEN

ADD CONSTRAINT SOLICITUD\_EXAMEN\_PK PRIMARY KEY ( id\_solicitud ) ;

CREATE TABLE TARJETA

(

id\_pago INTEGER NOT NULL ,

tipo CHAR (50 CHAR) NOT NULL ,

numero\_cuotas INTEGER

)

;

ALTER TABLE TARJETA

ADD CONSTRAINT TARJETA\_PK PRIMARY KEY ( id\_pago ) ;

CREATE TABLE TRANSFERENCIA

(

id\_pago INTEGER NOT NULL ,

id\_transferencia INTEGER NOT NULL ,

rut\_origen CHAR (13 CHAR) NOT NULL ,

rut\_destino CHAR (13 CHAR) NOT NULL ,

cuenta\_origen INTEGER NOT NULL ,

cuenta\_destino INTEGER NOT NULL

)

;

ALTER TABLE TRANSFERENCIA

ADD CONSTRAINT TRANSFERENCIA\_PK PRIMARY KEY ( id\_pago ) ;

ALTER TABLE TRANSFERENCIA

ADD CONSTRAINT TRANSFERENCIA\_PKv1 UNIQUE ( id\_transferencia ) ;

ALTER TABLE ATENCION\_MEDICO

ADD CONSTRAINT ATENCION\_MEDICA\_FK FOREIGN KEY

(

ATENCION\_MEDICA\_id\_atencion

)

REFERENCES ATENCION\_MEDICA

(

id\_atencion

)

;

ALTER TABLE ATENCION\_DIAGNOSTICO

ADD CONSTRAINT ATENCION\_MEDICA\_FKv1 FOREIGN KEY

(

ATENCION\_MEDICA\_id\_atencion

)

REFERENCES ATENCION\_MEDICA

(

id\_atencion

)

;

ALTER TABLE ATENCION\_MEDICA

ADD CONSTRAINT ATENCION\_MEDICA\_PACIENTE\_FK FOREIGN KEY

(

PACIENTE\_rut,

PACIENTE\_id\_paciente

)

REFERENCES PACIENTE

(

rut,

id\_paciente

)

;

ALTER TABLE ATENCION\_MEDICA

ADD CONSTRAINT ATENCION\_MEDICA\_PAGO\_FK FOREIGN KEY

(

PAGO\_id\_pago

)

REFERENCES PAGO

(

id\_pago

)

;

ALTER TABLE COMUNA

ADD CONSTRAINT COMUNA\_REGION\_FK FOREIGN KEY

(

REGION\_id\_region

)

REFERENCES REGION

(

id\_region

)

;

ALTER TABLE ATENCION\_DIAGNOSTICO

ADD CONSTRAINT DIAGNOSTICO\_FK FOREIGN KEY

(

DIAGNOSTICO\_id\_diagnostico

)

REFERENCES DIAGNOSTICO

(

id\_diagnostico

)

;

ALTER TABLE DIRECCION

ADD CONSTRAINT DIRECCION\_COMUNA\_FK FOREIGN KEY

(

COMUNA\_id\_comuna

)

REFERENCES COMUNA

(

id\_comuna

)

;

ALTER TABLE EFECTIVO

ADD CONSTRAINT EFECTIVO\_PAGO\_FK FOREIGN KEY

(

id\_pago

)

REFERENCES PAGO

(

id\_pago

)

;

ALTER TABLE MEDICO

ADD CONSTRAINT MEDICO\_AFP\_FK FOREIGN KEY

(

AFP\_id\_afp

)

REFERENCES AFP

(

id\_afp

)

;

ALTER TABLE MEDICO

ADD CONSTRAINT MEDICO\_ESPECIALIDAD\_FK FOREIGN KEY

(

ESPECIALIDAD\_id\_especialidad

)

REFERENCES ESPECIALIDAD

(

id\_especialidad

)

;

ALTER TABLE ATENCION\_MEDICO

ADD CONSTRAINT MEDICO\_FK FOREIGN KEY

(

MEDICO\_rut,

MEDICO\_id\_medico

)

REFERENCES MEDICO

(

rut,

id\_medico

)

;

ALTER TABLE MEDICO

ADD CONSTRAINT MEDICO\_INSTITUCION\_SALUD\_FK FOREIGN KEY

(

INSTITUCION\_SALUD\_id\_institucion

)

REFERENCES INSTITUCION\_SALUD

(

id\_institucion

)

;

ALTER TABLE MEDICO

ADD CONSTRAINT MEDICO\_MEDICO\_FK FOREIGN KEY

(

MEDICO\_rut,

MEDICO\_id\_medico

)

REFERENCES MEDICO

(

rut,

id\_medico

)

;

ALTER TABLE PAGO

ADD CONSTRAINT PAGO\_ATENCION\_MEDICA\_FK FOREIGN KEY

(

ATENCION\_MEDICA\_id\_atencion

)

REFERENCES ATENCION\_MEDICA

(

id\_atencion

)

;

ALTER TABLE Relation\_6

ADD CONSTRAINT Relation\_6\_DIRECCION\_FK FOREIGN KEY

(

DIRECCION\_id\_direccion

)

REFERENCES DIRECCION

(

id\_direccion

)

;

ALTER TABLE Relation\_6

ADD CONSTRAINT Relation\_6\_PACIENTE\_FK FOREIGN KEY

(

PACIENTE\_rut,

PACIENTE\_id\_paciente

)

REFERENCES PACIENTE

(

rut,

id\_paciente

)

;

-- ERROR: FK name length exceeds maximum allowed length(30)

ALTER TABLE SOLICITUD\_EXAMEN

ADD CONSTRAINT SOLICITUD\_EXAMEN\_ATENCION\_MEDICA\_FK FOREIGN KEY

(

ATENCION\_MEDICA\_id\_atencion

)

REFERENCES ATENCION\_MEDICA

(

id\_atencion

)

;

-- ERROR: FK name length exceeds maximum allowed length(30)

ALTER TABLE SOLICITUD\_EXAMEN

ADD CONSTRAINT SOLICITUD\_EXAMEN\_EXAMENES\_LABORATORIO\_FK FOREIGN KEY

(

EXAMENES\_LABORATORIO\_id\_examen

)

REFERENCES EXAMENES\_LABORATORIO

(

id\_examen

)

;

ALTER TABLE TARJETA

ADD CONSTRAINT TARJETA\_PAGO\_FK FOREIGN KEY

(

id\_pago

)

REFERENCES PAGO

(

id\_pago

)

;

ALTER TABLE TRANSFERENCIA

ADD CONSTRAINT TRANSFERENCIA\_PAGO\_FK FOREIGN KEY

(

id\_pago

)

REFERENCES PAGO

(

id\_pago

)

;

CREATE OR REPLACE TRIGGER FKNTM\_ATENCION\_DIAGNOSTICO

BEFORE UPDATE OF ATENCION\_MEDICA\_id\_atencion, DIAGNOSTICO\_id\_diagnostico

ON ATENCION\_DIAGNOSTICO

BEGIN

raise\_application\_error(-20225,'Non Transferable FK constraint on table ATENCION\_DIAGNOSTICO is violated');

END;

/

CREATE OR REPLACE TRIGGER FKNTM\_ATENCION\_MEDICA

BEFORE UPDATE OF PACIENTE\_rut, PACIENTE\_id\_paciente, PAGO\_id\_pago

ON ATENCION\_MEDICA

BEGIN

raise\_application\_error(-20225,'Non Transferable FK constraint on table ATENCION\_MEDICA is violated');

END;

/

CREATE OR REPLACE TRIGGER FKNTM\_ATENCION\_MEDICO

BEFORE UPDATE OF ATENCION\_MEDICA\_id\_atencion, MEDICO\_rut, MEDICO\_id\_medico

ON ATENCION\_MEDICO

BEGIN

raise\_application\_error(-20225,'Non Transferable FK constraint on table ATENCION\_MEDICO is violated');

END;

/

CREATE OR REPLACE TRIGGER FKNTM\_COMUNA

BEFORE UPDATE OF REGION\_id\_region

ON COMUNA

BEGIN

raise\_application\_error(-20225,'Non Transferable FK constraint on table COMUNA is violated');

END;

/

CREATE OR REPLACE TRIGGER FKNTM\_DIRECCION

BEFORE UPDATE OF COMUNA\_id\_comuna

ON DIRECCION

BEGIN

raise\_application\_error(-20225,'Non Transferable FK constraint on table DIRECCION is violated');

END;

/

CREATE OR REPLACE TRIGGER FKNTO\_MEDICO

BEFORE UPDATE OF MEDICO\_rut, MEDICO\_id\_medico

ON MEDICO

FOR EACH ROW

BEGIN

IF :old.MEDICO\_rut IS NOT NULL THEN

raise\_application\_error(-20225,'Non Transferable FK constraint MEDICO\_MEDICO\_FK on table MEDICO is violated');

END IF;

IF :old.MEDICO\_id\_medico IS NOT NULL THEN

raise\_application\_error(-20225,'Non Transferable FK constraint MEDICO\_MEDICO\_FK on table MEDICO is violated');

END IF;

END;

/

CREATE OR REPLACE TRIGGER FKNTM\_MEDICO

BEFORE UPDATE OF AFP\_id\_afp, INSTITUCION\_SALUD\_id\_institucion, ESPECIALIDAD\_id\_especialidad

ON MEDICO

BEGIN

raise\_application\_error(-20225,'Non Transferable FK constraint on table MEDICO is violated');

END;

/

CREATE OR REPLACE TRIGGER FKNTM\_PAGO

BEFORE UPDATE OF ATENCION\_MEDICA\_id\_atencion

ON PAGO

BEGIN

raise\_application\_error(-20225,'Non Transferable FK constraint on table PAGO is violated');

END;

/

CREATE OR REPLACE TRIGGER FKNTM\_Relation\_6

BEFORE UPDATE OF DIRECCION\_id\_direccion, PACIENTE\_rut, PACIENTE\_id\_paciente

ON Relation\_6

BEGIN

raise\_application\_error(-20225,'Non Transferable FK constraint on table Relation\_6 is violated');

END;

/

CREATE OR REPLACE TRIGGER FKNTM\_SOLICITUD\_EXAMEN

BEFORE UPDATE OF ATENCION\_MEDICA\_id\_atencion, EXAMENES\_LABORATORIO\_id\_examen

ON SOLICITUD\_EXAMEN

BEGIN

raise\_application\_error(-20225,'Non Transferable FK constraint on table SOLICITUD\_EXAMEN is violated');

END;

/

-- ERROR: No Discriminator Column found in Arc FKArc\_1 - constraint trigger for Arc cannot be generated

-- ERROR: No Discriminator Column found in Arc FKArc\_1 - constraint trigger for Arc cannot be generated

-- ERROR: No Discriminator Column found in Arc FKArc\_1 - constraint trigger for Arc cannot be generated

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE 19

-- CREATE INDEX 2

-- ALTER TABLE 40

-- CREATE VIEW 0

-- ALTER VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 10

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 0

-- CREATE MATERIALIZED VIEW 0

-- CREATE MATERIALIZED VIEW LOG 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 0

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

--

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

--

-- ERRORS 6

-- WARNINGS 3

**Paso 2:**

Además, tendrás que descargar el resultado y generar un archivo DMD y una subcarpeta.

Para ello, tendrás que hacer clic en la opción “Guardar como” del menú “Archivo”, lo que desplegará el submenú que se ilustra en la siguiente imagen:

**Figura 4**

*Cómo guardar Diseño en un archivo .dmd*

  
*Nota.* Ejemplo de diseño guardado como archivo .dmd con sus respectivas subcarpetas. Oracle. (s.f.). *Oracle SQL Developer Data Modeler* [Software]. [Oracle.com](https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/)

**Figura 5**

*Ejemplo de archivos de diseño generado con SQL Developer Data Modeler*

**Figura 6**

*Contenido estándar de la subcarpeta generada del ejemplo (Modelo\_Base)*

  
*Nota.* La figura muestra el contenido detallado de la carpeta de recursos asociada a un modelo de Oracle Data Modeler. Oracle. (s.f.). *Oracle SQL Developer Data Modeler* [Software]. [Oracle.com](https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/)

**Paso 3:** Una vez generado el archivo .dmd y su subcarpeta correspondiente, todo este contenido debe comprimirse en un solo archivo ZIP o RAR.

**Figura 7**

*Contenido del archivo comprimido*



*Nota.* La figura muestra la estructura interna del archivo comprimido Encargo\_Semanal.zip. Oracle. (s.f.). *Oracle SQL Developer Data Modeler* [Software]. [Oracle.com](https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/)

**Paso 4:** Este documento Word deberás subirlo al repositorio GitHub (sin comprimir). Si no has creado tu cuenta aún, puedes hacerlo a través del siguiente enlace:

<https://github.com/>

**Figura 8**

*Repositorio en GitHub*Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

*Nota.* Ejemplo genérico de archivos cargados en el repositorio GitHub. GitHub (s.f.). *GitHub.* [https://github.com/](https://github.com/%E2%80%AF%E2%80%AF)

**Paso 5:** Posteriormente, desde el repositorio, deberás generar un enlace de tu proyecto:

**Figura 9**

*Enlace de proyecto GitHub Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.*

*Nota.* Ejemplo genérico de donde se extrae un enlace en GitHub. GitHub (s.f.). *GitHub.* <https://github.com/>

**Paso 6:** Finalmente, deberás subir al AVA este documento Word sin comprimir (con las capturas Barker y Bachman o Ingeniería de la información) y el enlace de tu repositorio GitHub en la sección “Entrega”.

**Figura 10**

*Visualización de entrega en el AVA*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente



Reservados todos los derechos Fundación Instituto Profesional Duoc UC. No se permite copiar, reproducir, reeditar, descargar, publicar, emitir, difundir, de forma total o parcial la presente obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de Fundación Instituto Profesional Duoc UC La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.