

## Semana 5

# Modelamiento de Bases de Datos

Formato de respuesta

Nombre estudiante:	Antonio González
Asignatura: Modelamiento de bases de datos	Carrera: Desarrollo de aplicaciones
Profesor: Lourdes Townsend	Fecha: 15/09/2025

## 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.

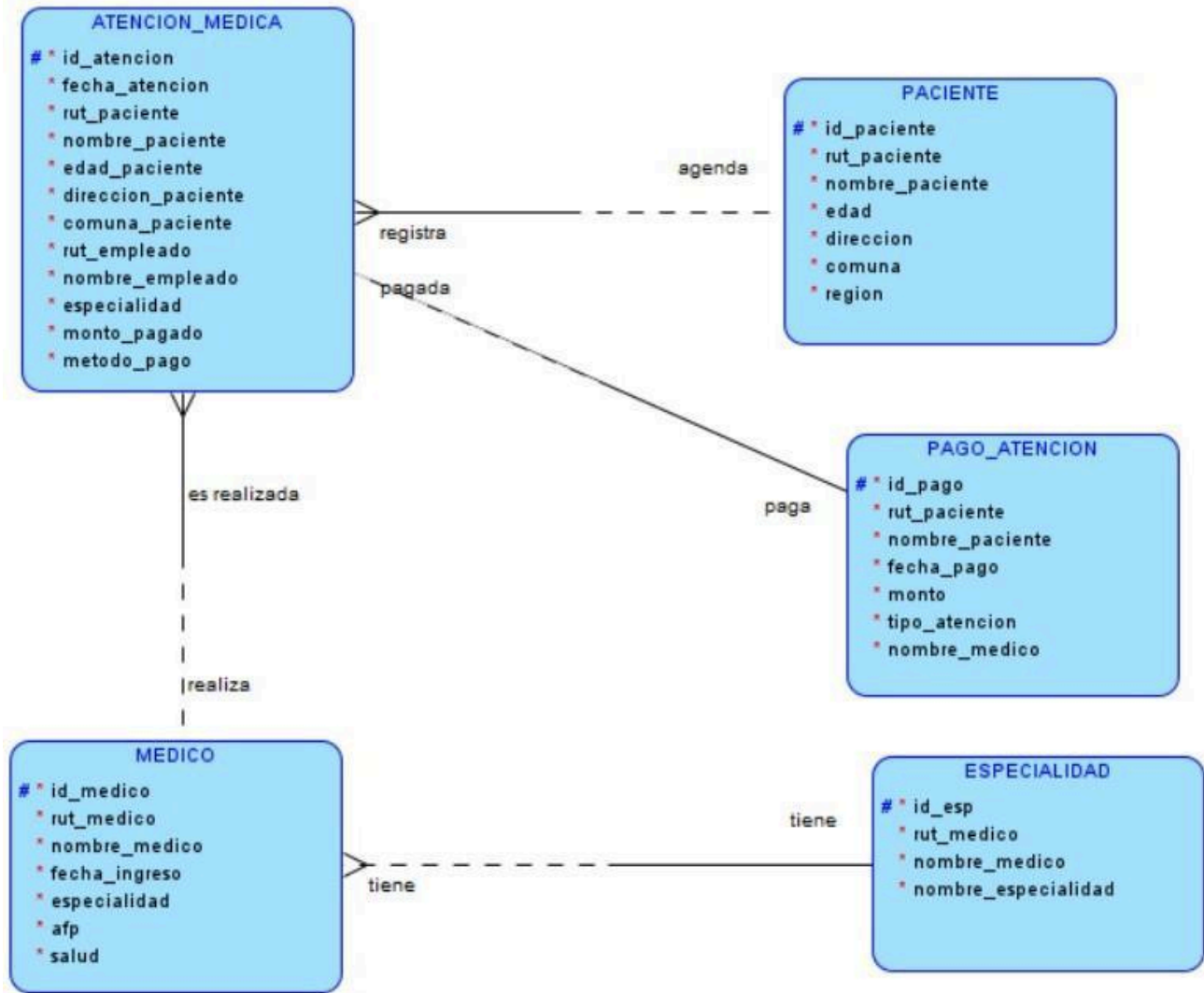
Actualmente, la información se gestiona en una hoja de cálculo donde cada fila representa una atención completa. Esta estructura plana provoca duplicación de datos, errores de ingreso y dificultades en el seguimiento clínico.

**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/sql/)

**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*

BONO FONASA		Monto pagado:
CENTRO MÉDICO UNIVERSITARIO		<b>\$12.700</b>
Folio:	830142229	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 un Bono de Fonasa.

## Figura 3

*Vista de Usuario: Agendamiento de hora*

### AGENDA TU HORA

Ingresa tus datos y selecciona la atención solicitada:

Nombre:	<input type="text"/>
Rut:	<input type="text" value="18555555-3"/>
Fecha de nacimiento:	<input type="text" value="27-11-1968"/>
Especialidad:	<input type="text" value="Ginecología"/> ▼
Tipo de atención:	<input type="text" value="Preventiva"/> ▼
Fecha cita:	<input type="text" value="22-04-2024"/>

*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/>

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

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

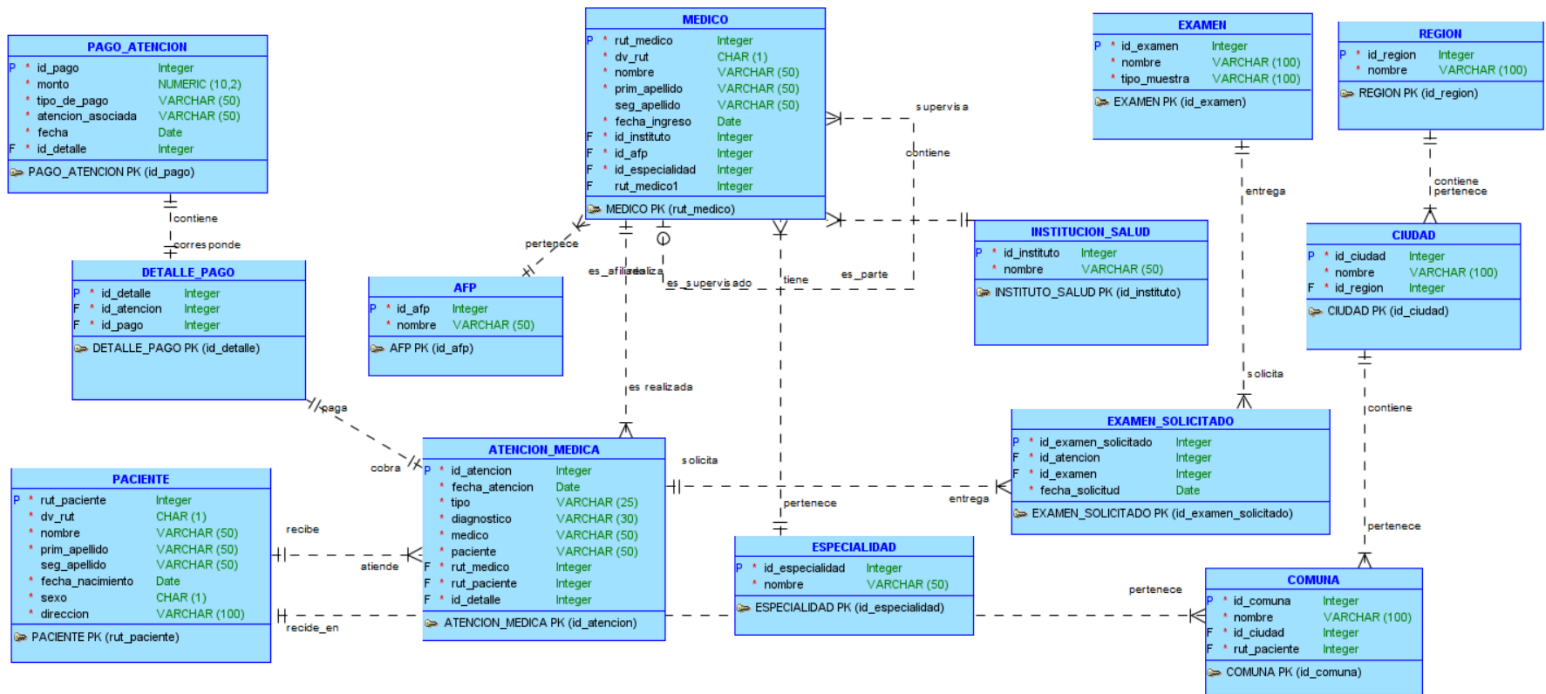
### Ejemplos:

Modelo MER-E Normalizado en notación Barker	Modelo Relacional Normalizado	Script DDL
		<pre> Editor de Archivo DDL - Oracle Database 11g Relational_1  100 CREATE TABLE paciente ( 101     rut                NUMBER(9) NOT NULL, 102     dv                 CHAR(1) NOT NULL, 103     pri_nombre         VARCHAR2(25) NOT NULL, 104     pri_apellido        VARCHAR2(25) NOT NULL, 105     seg_apellido        VARCHAR2(25) NOT NULL, 106     fecha_nac          DATE NOT NULL, 107     telefono           NUMBER(9) NOT NULL, 108     ciudad_cod_ciudad  NUMBER(3) NOT NULL 109 ); 110 111 ALTER TABLE paciente 112     ADD CONSTRAINT dv_ck CHECK ( dv IN ( '0', '1', 113   '5', '6', 114   'k' ) ); 115 116 ALTER TABLE paciente ADD CONSTRAINT paciente_pk PRI 117 118 CREATE TABLE receta ( 119     id_receta           NUMBER(9) NOT NULL, 120     observacion         VARCHAR2(30), 121     digitador_id_digitador NUMBER(5) NOT NULL, 122     boleta_n_boleta     NUMBER(5) NOT NULL, 123     diagnostico_cod_diag NUMBER(4) NOT NULL, 124     medico_id_medico    NUMBER(6) NOT NULL, 125     paciente_rut        NUMBER(9) NOT NULL, 126     tipo_receta_tipo_receta_id NUMBER NOT NULL 127 ); </pre>



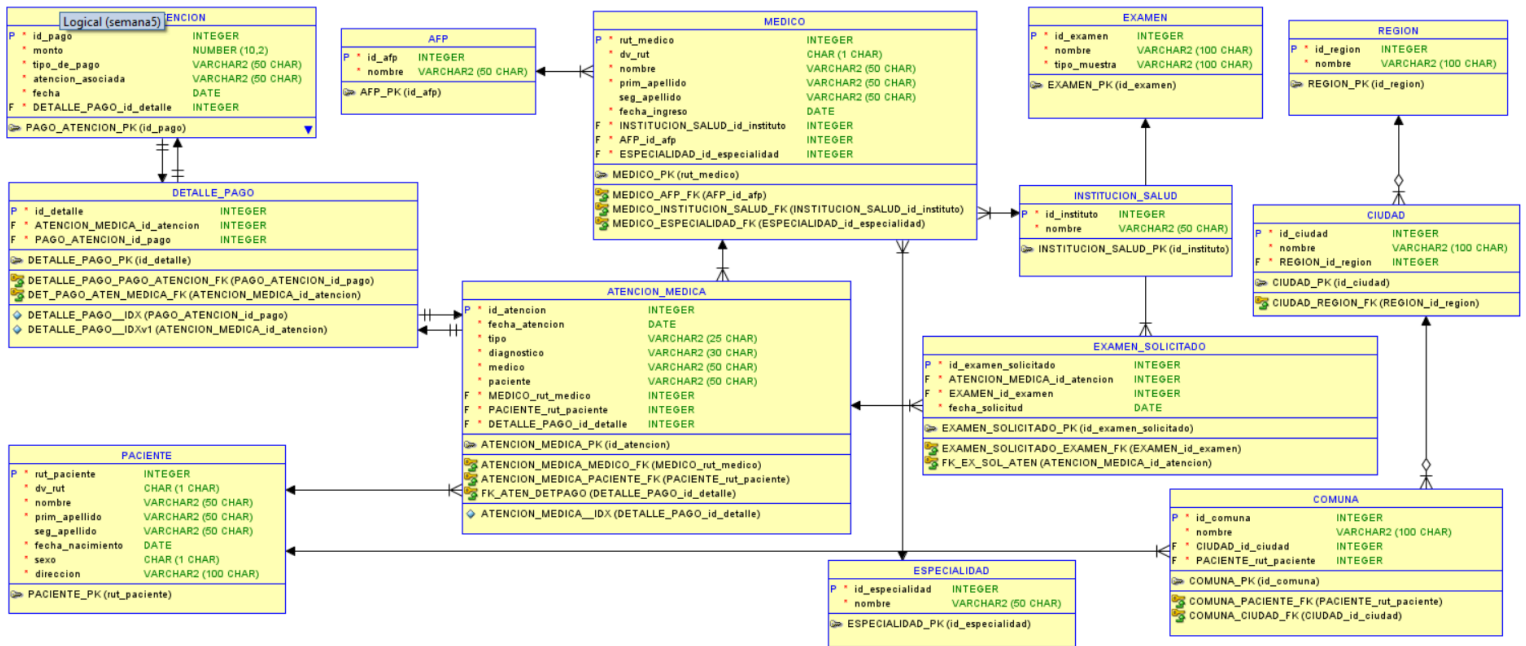
Adjunta tus evidencias en esta parte:

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





## Modelo Relacional (MR) Normalizado:



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

```
-- Generado por Oracle SQL Developer Data Modeler 24.3.1.351.0831
-- en:      2025-09-15 13:51:48 AEST
-- sitio:   Oracle Database 21c
-- tipo:    Oracle Database 21c
```

```
DROP TABLE AFP CASCADE CONSTRAINTS
;
```

```
DROP TABLE ATENCION_MEDICA CASCADE CONSTRAINTS
;
```

```
DROP TABLE CIUDAD CASCADE CONSTRAINTS
;
```

```
DROP TABLE COMUNA CASCADE CONSTRAINTS
;
```

```
DROP TABLE DETALLE_PAGO CASCADE CONSTRAINTS
;
```

```
DROP TABLE ESPECIALIDAD CASCADE CONSTRAINTS
;
```

```
DROP TABLE EXAMEN CASCADE CONSTRAINTS
;
```

```
DROP TABLE EXAMEN_SOLICITADO CASCADE CONSTRAINTS
;
```

```
DROP TABLE INSTITUCION_SALUD CASCADE CONSTRAINTS
;
```

```
DROP TABLE MEDICO CASCADE CONSTRAINTS
;
```

```
DROP TABLE PACIENTE CASCADE CONSTRAINTS
;
```

```
DROP TABLE PAGO_ATENCION CASCADE CONSTRAINTS
;
```

```
DROP TABLE REGION CASCADE CONSTRAINTS
;
```

```
-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
```

```
-- predefined type, no DDL - XMLTYPE
```

```
CREATE TABLE AFP
(
  id_afp INTEGER NOT NULL ,
  nombre VARCHAR2 (50 CHAR) NOT NULL
)
;
```

```
ALTER TABLE AFP
```

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

```
ADD CONSTRAINT AFP_PK PRIMARY KEY ( id_afp );

CREATE TABLE ATENCION_MEDICA
(
    id_atencion          INTEGER NOT NULL ,
    fecha_atencion       DATE  NOT NULL ,
    tipo                 VARCHAR2 (25 CHAR) NOT NULL ,
    diagnostico          VARCHAR2 (30 CHAR) NOT NULL ,
    medico               VARCHAR2 (50 CHAR) NOT NULL ,
    paciente             VARCHAR2 (50 CHAR) NOT NULL ,
    MEDICO_rut_medico    INTEGER NOT NULL ,
    PACIENTE_rut_paciente INTEGER NOT NULL ,
    DETALLE_PAGO_id_detalle INTEGER NOT NULL
)
;
CREATE UNIQUE INDEX ATENCION_MEDICA__IDX ON ATENCION_MEDICA
(
    DETALLE_PAGO_id_detalle ASC
)
;

ALTER TABLE ATENCION_MEDICA
    ADD CONSTRAINT ATENCION_MEDICA_PK PRIMARY KEY ( id_atencion );

CREATE TABLE CIUDAD
(
    id_ciudad           INTEGER NOT NULL ,
    nombre              VARCHAR2 (100 CHAR) NOT NULL ,
    REGION_id_region    INTEGER NOT NULL
)
;

ALTER TABLE CIUDAD
    ADD CONSTRAINT CIUDAD_PK PRIMARY KEY ( id_ciudad );

CREATE TABLE COMUNA
(
    id_comuna           INTEGER NOT NULL ,
    nombre              VARCHAR2 (100 CHAR) NOT NULL ,
    CIUDAD_id_ciudad    INTEGER NOT NULL ,
    PACIENTE_rut_paciente INTEGER NOT NULL
)
;

ALTER TABLE COMUNA
    ADD CONSTRAINT COMUNA_PK PRIMARY KEY ( id_comuna );

CREATE TABLE DETALLE_PAGO
(
    id_detalle          INTEGER NOT NULL ,
    ATENCION_MEDICA_id_atencion INTEGER NOT NULL ,
    PAGO_ATENCION_id_pago  INTEGER NOT NULL
)
;
CREATE UNIQUE INDEX DETALLE_PAGO__IDX ON DETALLE_PAGO
(
    PAGO_ATENCION_id_pago ASC
)
```

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

```
)
;
CREATE UNIQUE INDEX DETALLE_PAGO__IDXv1 ON DETALLE_PAGO
(
    ATENCION_MEDICA_id_atencion ASC
)
;

ALTER TABLE DETALLE_PAGO
    ADD CONSTRAINT DETALLE_PAGO_PK PRIMARY KEY ( id_detalle );

CREATE TABLE ESPECIALIDAD
(
    id_especialidad INTEGER NOT NULL ,
    nombre          VARCHAR2 (50 CHAR) NOT NULL
)
;

ALTER TABLE ESPECIALIDAD
    ADD CONSTRAINT ESPECIALIDAD_PK PRIMARY KEY ( id_especialidad );

CREATE TABLE EXAMEN
(
    id_examen  INTEGER NOT NULL ,
    nombre     VARCHAR2 (100 CHAR) NOT NULL ,
    tipo_muestra VARCHAR2 (100 CHAR) NOT NULL
)
;

ALTER TABLE EXAMEN
    ADD CONSTRAINT EXAMEN_PK PRIMARY KEY ( id_examen );

CREATE TABLE EXAMEN_SOLICITADO
(
    id_examen_solicitado  INTEGER NOT NULL ,
    ATENCION_MEDICA_id_atencion INTEGER NOT NULL ,
    EXAMEN_id_examen      INTEGER NOT NULL ,
    fecha_solicitud       DATE NOT NULL
)
;

ALTER TABLE EXAMEN_SOLICITADO
    ADD CONSTRAINT EXAMEN_SOLICITADO_PK PRIMARY KEY ( id_examen_solicitado );

CREATE TABLE INSTITUCION_SALUD
(
    id_instituto INTEGER NOT NULL ,
    nombre       VARCHAR2 (50 CHAR) NOT NULL
)
;

ALTER TABLE INSTITUCION_SALUD
    ADD CONSTRAINT INSTITUCION_SALUD_PK PRIMARY KEY ( id_instituto );

CREATE TABLE MEDICO
(
    rut_medico          INTEGER NOT NULL ,
```

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

```
dv_rut          CHAR (1 CHAR) NOT NULL ,
nombre          VARCHAR2 (50 CHAR) NOT NULL ,
prim_apellido   VARCHAR2 (50 CHAR) NOT NULL ,
seg_apellido    VARCHAR2 (50 CHAR) ,
fecha_ingreso   DATE NOT NULL ,
INSTITUCION_SALUD_id_instituto INTEGER NOT NULL ,
AFP_id_afp      INTEGER NOT NULL ,
ESPECIALIDAD_id_especialidad  INTEGER NOT NULL
)
;

ALTER TABLE MEDICO
  ADD CONSTRAINT MEDICO_PK PRIMARY KEY ( rut_medico ) ;

CREATE TABLE PACIENTE
(
  rut_paciente   INTEGER NOT NULL ,
  dv_rut         CHAR (1 CHAR) NOT NULL ,
  nombre         VARCHAR2 (50 CHAR) NOT NULL ,
  prim_apellido  VARCHAR2 (50 CHAR) NOT NULL ,
  seg_apellido   VARCHAR2 (50 CHAR) ,
  fecha_nacimiento DATE NOT NULL ,
  sexo          CHAR (1 CHAR) NOT NULL ,
  direccion      VARCHAR2 (100 CHAR) NOT NULL
)
;

ALTER TABLE PACIENTE
  ADD CONSTRAINT PACIENTE_PK PRIMARY KEY ( rut_paciente ) ;

CREATE TABLE PAGO_ATENCION
(
  id_pago        INTEGER NOT NULL ,
  monto          NUMBER (10,2) NOT NULL ,
  tipo_de_pago   VARCHAR2 (50 CHAR) NOT NULL ,
  atencion_asociada  VARCHAR2 (50 CHAR) NOT NULL ,
  fecha          DATE NOT NULL ,
  DETALLE_PAGO_id_detalle INTEGER NOT NULL
)
;

CREATE UNIQUE INDEX PAGO_ATENCION__IDX ON PAGO_ATENCION
(
  DETALLE_PAGO_id_detalle ASC
)
;

ALTER TABLE PAGO_ATENCION
  ADD CONSTRAINT PAGO_ATENCION_PK PRIMARY KEY ( id_pago ) ;

CREATE TABLE REGION
(
  id_region      INTEGER NOT NULL ,
  nombre        VARCHAR2 (100 CHAR) NOT NULL
)
;

ALTER TABLE REGION
```

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

```
ADD CONSTRAINT REGION_PK PRIMARY KEY ( id_region ) ;
```

```
ALTER TABLE ATENCION_MEDICA
ADD CONSTRAINT ATENCION_MEDICA_MEDICO_FK FOREIGN KEY
(
    MEDICO_rut_medico
)
REFERENCES MEDICO
(
    rut_medico
)
;
```

```
ALTER TABLE ATENCION_MEDICA
ADD CONSTRAINT ATENCION_MEDICA_PACIENTE_FK FOREIGN KEY
(
    PACIENTE_rut_paciente
)
REFERENCES PACIENTE
(
    rut_paciente
)
;
```

```
ALTER TABLE CIUDAD
ADD CONSTRAINT CIUDAD_REGION_FK FOREIGN KEY
(
    REGION_id_region
)
REFERENCES REGION
(
    id_region
)
;
```

```
ALTER TABLE COMUNA
ADD CONSTRAINT COMUNA_CIUDAD_FK FOREIGN KEY
(
    CIUDAD_id_ciudad
)
REFERENCES CIUDAD
(
    id_ciudad
)
;
```

```
ALTER TABLE COMUNA
ADD CONSTRAINT COMUNA_PACIENTE_FK FOREIGN KEY
(
    PACIENTE_rut_paciente
)
REFERENCES PACIENTE
(
    rut_paciente
)
;
```

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

```
ALTER TABLE DETALLE_PAGO
  ADD CONSTRAINT DET_PAGO_ATEN_MEDICA_FK FOREIGN KEY
  (
    ATENCION_MEDICA_id_atencion
  )
  REFERENCES ATENCION_MEDICA
  (
    id_atencion
  )
;

ALTER TABLE DETALLE_PAGO
  ADD CONSTRAINT DETALLE_PAGO_PAGO_ATENCION_FK FOREIGN KEY
  (
    PAGO_ATENCION_id_pago
  )
  REFERENCES PAGO_ATENCION
  (
    id_pago
  )
;

ALTER TABLE EXAMEN_SOLICITADO
  ADD CONSTRAINT EXAMEN_SOLICITADO_EXAMEN_FK FOREIGN KEY
  (
    EXAMEN_id_examen
  )
  REFERENCES EXAMEN
  (
    id_examen
  )
;

ALTER TABLE ATENCION_MEDICA
  ADD CONSTRAINT FK_ATEN_DETPAGO FOREIGN KEY
  (
    DETALLE_PAGO_id_detalle
  )
  REFERENCES DETALLE_PAGO
  (
    id_detalle
  )
;

ALTER TABLE EXAMEN_SOLICITADO
  ADD CONSTRAINT FK_EX_SOL_ATEN FOREIGN KEY
  (
    ATENCION_MEDICA_id_atencion
  )
  REFERENCES ATENCION_MEDICA
  (
    id_atencion
  )
;

ALTER TABLE MEDICO
  ADD CONSTRAINT MEDICO_AFP_FK FOREIGN KEY
```



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

```
(
  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 MEDICO
  ADD CONSTRAINT MEDICO_INSTITUCION_SALUD_FK FOREIGN KEY
  (
    INSTITUCION_SALUD_id_instituto
  )
  REFERENCES INSTITUCION_SALUD
  (
    id_instituto
  )
;

ALTER TABLE PAGO_ATENCION
  ADD CONSTRAINT PAGO_ATENCION_DETALLE_PAGO_FK FOREIGN KEY
  (
    DETALLE_PAGO_id_detalle
  )
  REFERENCES DETALLE_PAGO
  (
    id_detalle
  )
;

CREATE OR REPLACE TRIGGER FKNTM_CIUADAD
BEFORE UPDATE OF REGION_id_region
ON CIUDAD
BEGIN
  raise_application_error(-20225,'Non Transferable FK constraint  on table CIUDAD is violated');
END;
/

CREATE OR REPLACE TRIGGER FKNTM_COMUNA
BEFORE UPDATE OF CIUDAD_id_ciudad
ON COMUNA
BEGIN
  raise_application_error(-20225,'Non Transferable FK constraint  on table COMUNA is violated');
END;
/
```

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

-- Informe de Resumen de Oracle SQL Developer Data Modeler:

```
--
-- CREATE TABLE          13
-- CREATE INDEX           4
-- ALTER TABLE          27
-- CREATE VIEW            0
-- ALTER VIEW            0
-- CREATE PACKAGE          0
-- CREATE PACKAGE BODY    0
-- CREATE PROCEDURE       0
-- CREATE FUNCTION        0
-- CREATE TRIGGER         2
-- 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                 0
-- WARNINGS               0
```

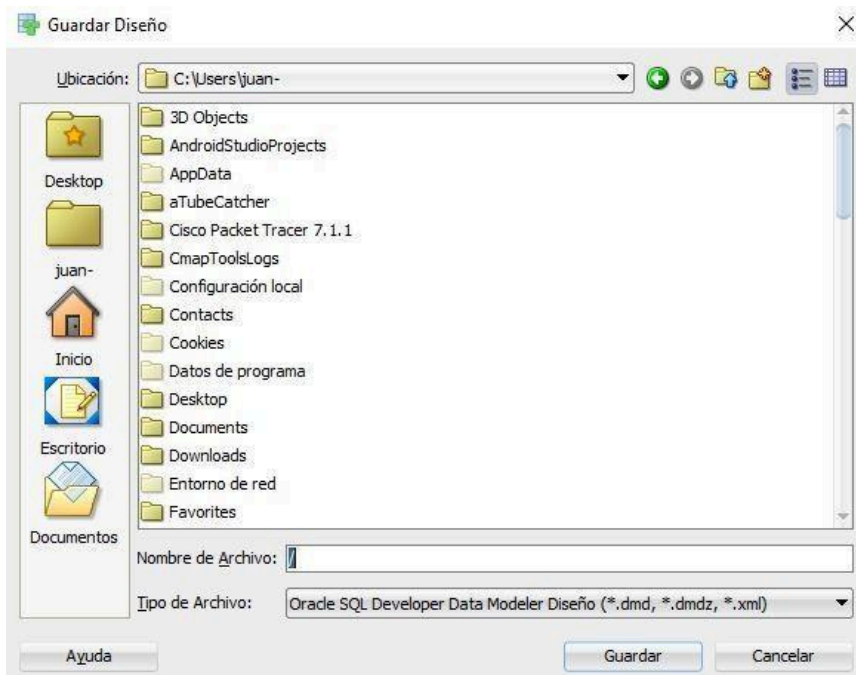
## Paso 2:

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)

**Figura 5**

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

Nombre	Tipo
Modelo_Base	Carpeta de archivos
Modelo_Base.dmd	Archivo DMD

**Figura 6**

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

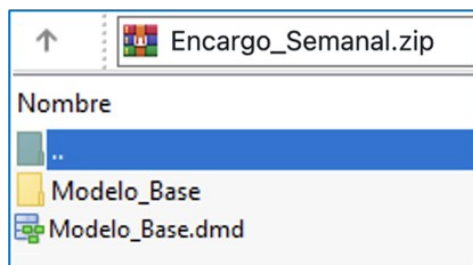
Nombre	Fecha de modificación	Tipo
businessinfo	03/10/2024 15:55	Carpeta de archivos
datatypes	03/10/2024 15:55	Carpeta de archivos
files	03/10/2024 15:55	Carpeta de archivos
logical	03/10/2024 15:55	Carpeta de archivos
mapping	03/10/2024 15:55	Carpeta de archivos
pm	03/10/2024 15:55	Carpeta de archivos
rdbms	03/10/2024 15:55	Carpeta de archivos
rel	03/10/2024 15:55	Carpeta de archivos
ChangeRequests.local	03/10/2024 15:55	Archivo LOCAL
dl_settings.xml	03/10/2024 15:55	xmlfile

*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)

**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*



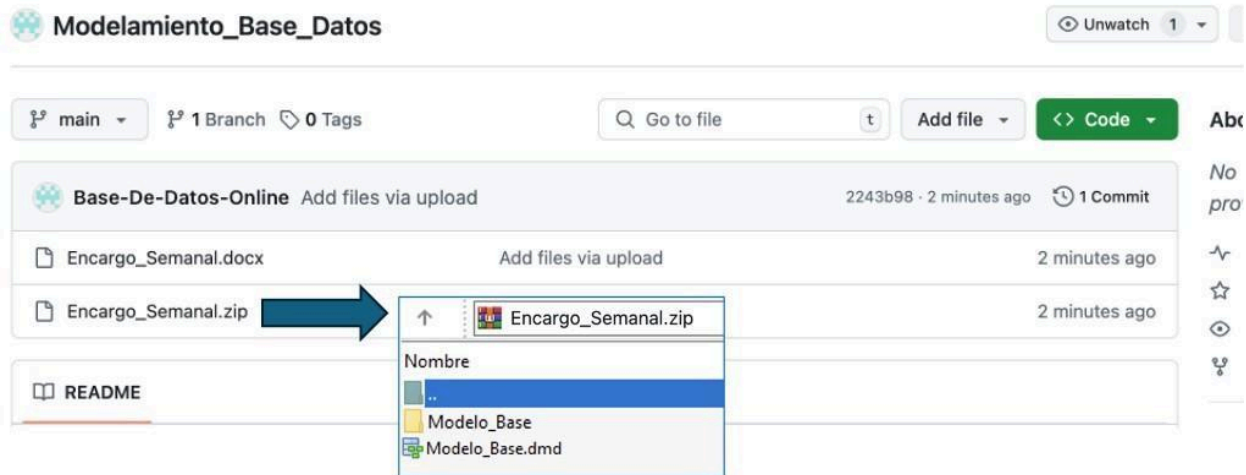
*Nota.* La figura muestra la estructura interna del archivo comprimido Encargo\_Semanal.zip. Oracle. (s.f.). *Oracle SQL Developer Data Modeler* [Software]. [Oracle.com](https://www.oracle.com)

**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*



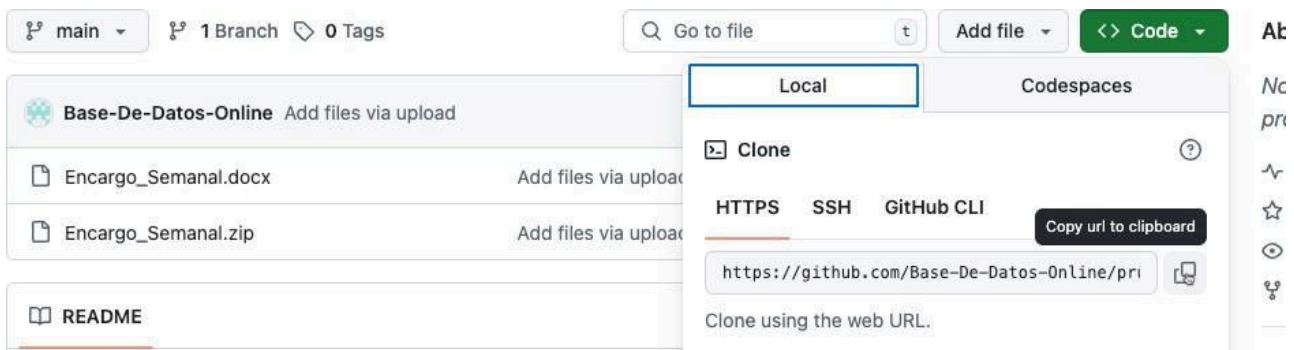
*Nota.* Ejemplo genérico de archivos cargados en el repositorio GitHub. GitHub (s.f.). *GitHub*.

<https://github.com/>

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

**Figura 9**

*Enlace de proyecto GitHub*



*Nota.* Ejemplo genérico de donde se extrae un enlace en GitHub. GitHub (s.f.). *GitHub*.

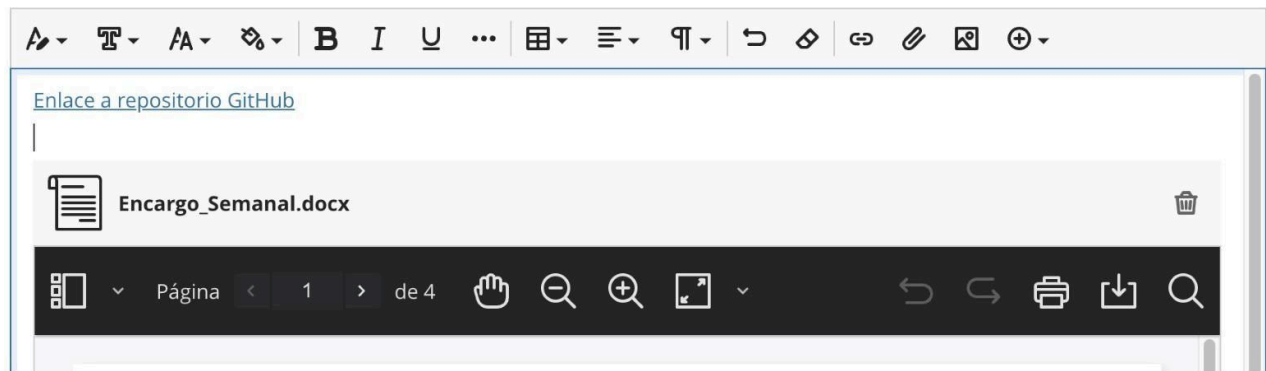
<https://github.com/>

**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*

### Entrega





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.