

## Semana 5

# Modelamiento de Bases de Datos

Formato de respuesta

<b>Nombre estudiante:</b> Jorge Gallardo Heck	
<b>Asignatura:</b> Modelamiento de Base de Datos	<b>Carrera:</b> Analista Programador Computacional
<b>Profesor:</b> Rodrigo Opazo Salazar	<b>Fecha:</b> 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.

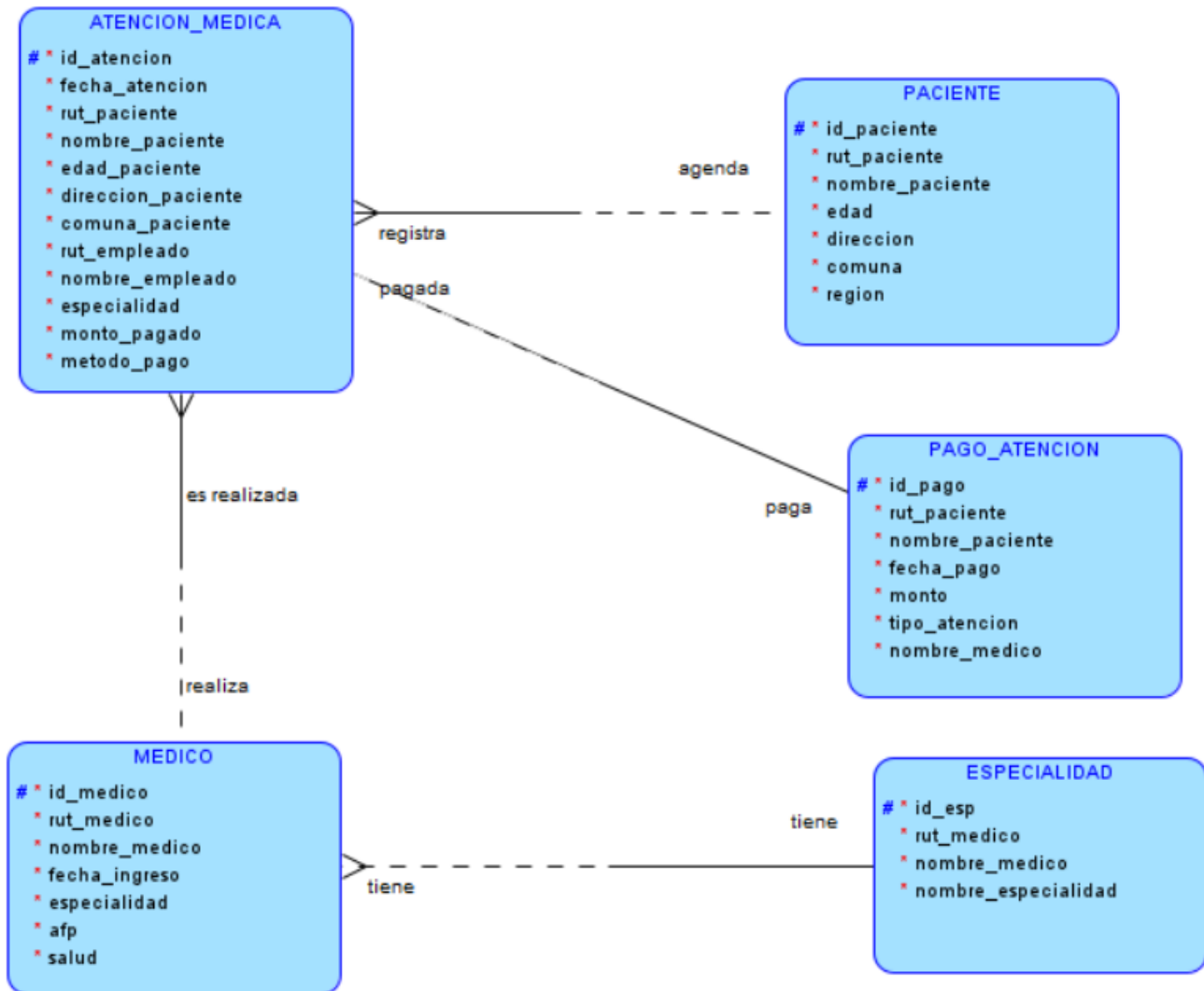
Actualmente, la información se gestiona en una hoja de cálculo donde cada fila representa una atención completa. Esta estructura plana provoca duplicación de datos, errores de ingreso y dificultades en el seguimiento clínico.

### **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-developer/)

**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:

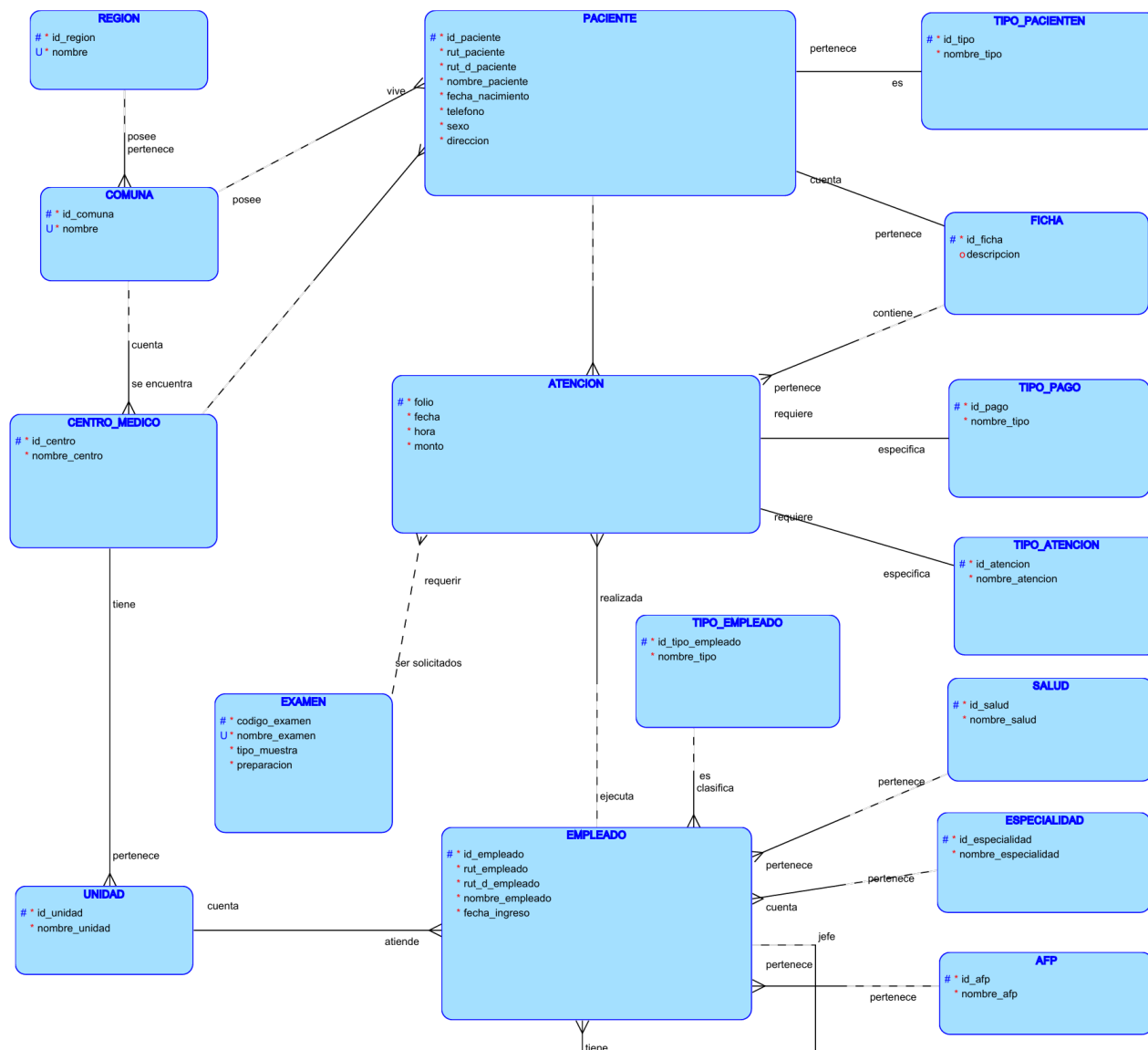
- Modelo Entidad-Relación-Extendido (MER-E) Normalizado solicitado en notación Barker.
- Modelo Relacional (MR) Normalizado.
- Script DDL generado con la herramienta Oracle Data Modeler.

Ejemplos:

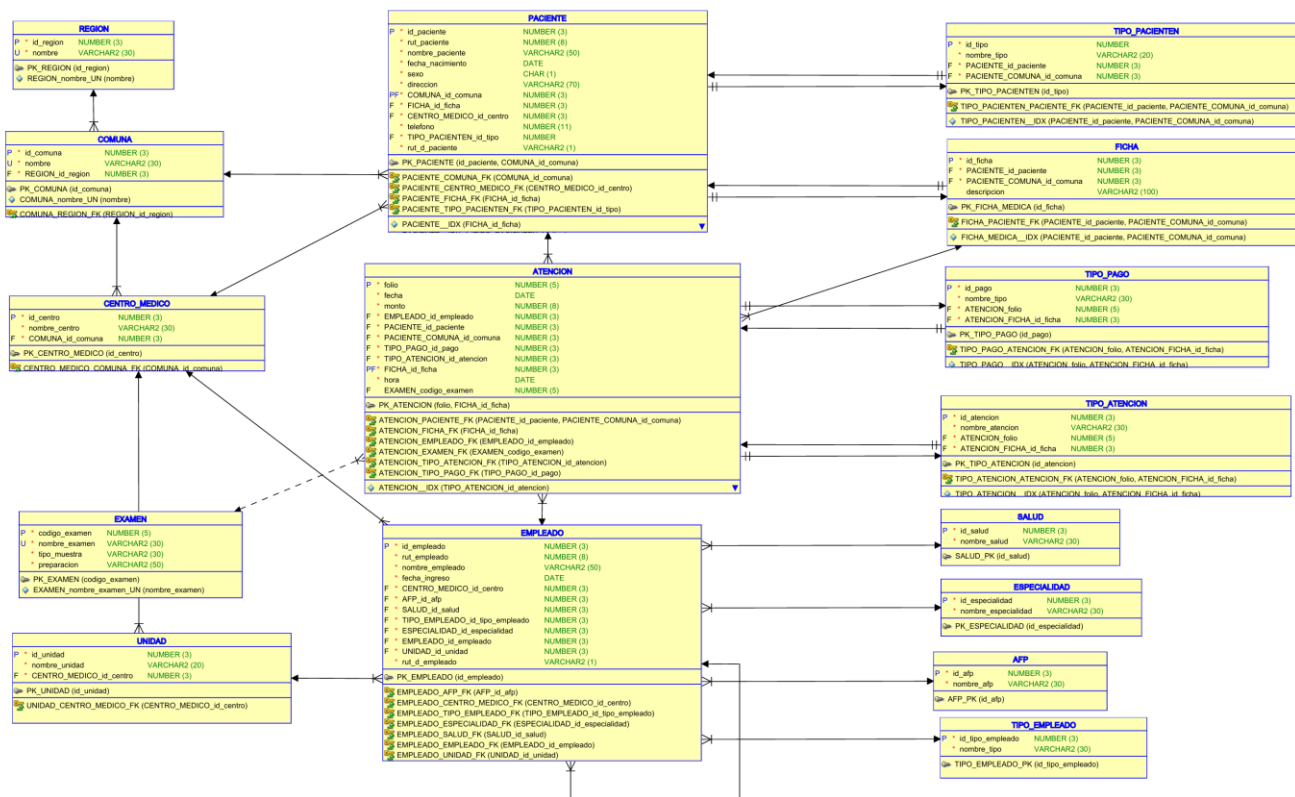
Modelo MER-E Normalizado en notación Barker	Modelo Relacional Normalizado	Script DDL
		<pre> Editor de Archivo DDL - Oracle Database 11g Relational_1  100 CREATE TABLE paciente ( 101     rut          NUMBER(8) NOT NULL, 102     dv           CHAR(1) NOT NULL, 103     pri_nombre   VARCHAR2(25) NOT NULL, 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(8) NOT NULL, 120     observacion   VARCHAR2(30), 121     digitador_id_digitador NUMBER(5) NOT NULL, 122     boleta_n_boleta NUMBER(8) NOT NULL, 123     diagnostico_cod_diag NUMBER(4) NOT NULL, 124     medico_id_medico NUMBER(6) NOT NULL, 125     paciente_rut   NUMBER(8) 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:







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

```
Exp2_S5_Jorge_Gallardo_DDL: Bloc de notas
Archivo Edición Formato Ver Ayuda
-- Generado por Oracle SQL Developer Data Modeler 24.3.1.351.0831
-- en: 2025-09-11 15:24:22 CLST
-- sitio: Oracle Database 11g
-- tipo: Oracle Database 11g

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

CREATE TABLE AFP
(
    id_afp NUMBER (3) NOT NULL ,
    nombre_afp VARCHAR2 (30) NOT NULL
)
;

ALTER TABLE AFP
ADD CONSTRAINT AFP_PK PRIMARY KEY ( id_afp ) ;

CREATE TABLE ATENCION
(
    folio NUMBER (5) NOT NULL ,
    fecha DATE NOT NULL ,
    monto NUMBER (8) NOT NULL ,
    EMPLEADO_id_empleado NUMBER (3) NOT NULL ,
    PACIENTE_id_paciente NUMBER (3) NOT NULL ,
    PACIENTE_COMUNA_id_comuna NUMBER (3) NOT NULL ,
    TIPO_PAGO_id_pago NUMBER (3) NOT NULL ,
    TIPO_ATENCION_id_atencion NUMBER (3) NOT NULL ,
    FICHA_id_ficha NUMBER (3) NOT NULL ,
    hora DATE NOT NULL ,
    EXAMEN_codigo_examen NUMBER (5)
)
;

Linea 1, columna 1 100% Windows (CRLF) UTF-8
23°C 16:02 11-09-2025
```

```
Exp2_S5_Jorge_Gallardo_DDL: Bloc de notas
Archivo Edición Formato Ver Ayuda
ALTER TABLE ATENCION
ADD CONSTRAINT CK_ATENCION_MONTO
CHECK (monto>0)
;

CREATE UNIQUE INDEX ATENCION__IDX ON ATENCION
(
    TIPO_ATENCION_id_atencion ASC
)
;

CREATE UNIQUE INDEX ATENCION__IDXv1 ON ATENCION
(
    TIPO_PAGO_id_pago ASC
)
;

ALTER TABLE ATENCION
ADD CONSTRAINT PK_ATENCION PRIMARY KEY ( folio, FICHA_id_ficha ) ;

CREATE TABLE CENTRO_MEDICO
(
    id_centro NUMBER (3) NOT NULL ,
    nombre_centro VARCHAR2 (30) NOT NULL ,
    COMUNA_id_comuna NUMBER (3) NOT NULL
)
;

ALTER TABLE CENTRO_MEDICO
ADD CONSTRAINT PK_CENTRO_MEDICO PRIMARY KEY ( id_centro ) ;

CREATE TABLE COMUNA
(
    id_comuna NUMBER (3) NOT NULL ,
    nombre VARCHAR2 (30) NOT NULL ,
    REGION_id_region NUMBER (3) NOT NULL
)
;

Linea 1, columna 1 100% Windows (CRLF) UTF-8
23°C 16:03 11-09-2025
```

Exp2\_S5\_Jorge\_Gallardo\_DDL: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
ALTER TABLE COMUNA
ADD CONSTRAINT PK_COMUNA PRIMARY KEY ( id_comuna );

ALTER TABLE COMUNA
ADD CONSTRAINT COMUNA_nombre_UN UNIQUE ( nombre );

CREATE TABLE EMPLEADO
(
    id_empleado          NUMBER (3) NOT NULL ,
    rut_empleado         NUMBER (8) NOT NULL ,
    nombre_empleado      VARCHAR2 (50) NOT NULL ,
    fecha_ingreso        DATE NOT NULL ,
    CENTRO_MEDICO_id_centro NUMBER (3) NOT NULL ,
    AFP_id_afp           NUMBER (3) NOT NULL ,
    SALUD_id_salud       NUMBER (3) NOT NULL ,
    TIPO_EMPLEADO_id_tipo_empleado NUMBER (3) NOT NULL ,
    ESPECIALIDAD_id_especialidad NUMBER (3) NOT NULL ,
    EMPLEADO_id_empleado NUMBER (3) NOT NULL ,
    UNIDAD_id_unidad     NUMBER (3) NOT NULL ,
    rut_d_empleado       VARCHAR2 (1) NOT NULL
);

ALTER TABLE EMPLEADO
ADD CONSTRAINT CK_EMPLEADO_RUT_D
CHECK (rut_d_empleado IN ('"0"', '"1"', '"2"', '"3"', '"4"', '"5"', '"6"', '"7"', '"8"', '"9"', '"K"', '"k"'))
;

ALTER TABLE EMPLEADO
ADD CONSTRAINT PK_EMPLEADO PRIMARY KEY ( id_empleado );

CREATE TABLE ESPECIALIDAD
(
    id_especialidad      NUMBER (3) NOT NULL ,
```

Exp2\_S5\_Jorge\_Gallardo\_DDL: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
CREATE TABLE ESPECIALIDAD
(
    id_especialidad      NUMBER (3) NOT NULL ,
    nombre_especialidad VARCHAR2 (30) NOT NULL
);

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

CREATE TABLE EXAMEN
(
    codigo_examen NUMBER (5) NOT NULL ,
    nombre_examen VARCHAR2 (30) NOT NULL ,
    tipo_muestra  VARCHAR2 (30) NOT NULL ,
    preparacion   VARCHAR2 (50) NOT NULL
);

ALTER TABLE EXAMEN
ADD CONSTRAINT PK_EXAMEN PRIMARY KEY ( codigo_examen );

ALTER TABLE EXAMEN
ADD CONSTRAINT EXAMEN_nombre_examen_UN UNIQUE ( nombre_examen );

CREATE TABLE FICHA
(
    id_ficha          NUMBER (3) NOT NULL ,
    PACIENTE_id_paciente NUMBER (3) NOT NULL ,
    PACIENTE_COMUNA_id_comuna NUMBER (3) NOT NULL ,
    descripcion       VARCHAR2 (100)
);

CREATE UNIQUE INDEX FICHA_MEDICA__IDX ON FICHA
(
```

Exp2\_S5\_Jorge\_Gallardo\_DDL: Bloc de notas

Archivo Edición Formato Ver Ayuda

```

;
CREATE UNIQUE INDEX FICHA_MEDICA__IDX ON FICHA
(
    PACIENTE_id_paciente ASC ,
    PACIENTE_COMUNA_id_comuna ASC
)
;

ALTER TABLE FICHA
ADD CONSTRAINT PK_FICHA_MEDICA PRIMARY KEY ( id_ficha ) ;

CREATE TABLE PACIENTE
(
    id_paciente          NUMBER (3) NOT NULL ,
    rut_paciente         NUMBER (8) NOT NULL ,
    nombre_paciente     VARCHAR2 (50) NOT NULL ,
    fecha_nacimiento    DATE NOT NULL ,
    sexo                CHAR (1) NOT NULL ,
    direccion            VARCHAR2 (70) NOT NULL ,
    COMUNA_id_comuna     NUMBER (3) NOT NULL ,
    FICHA_id_ficha       NUMBER (3) NOT NULL ,
    CENTRO_MEDICO_id_centro NUMBER (3) NOT NULL ,
    telefono            NUMBER (11) NOT NULL ,
    TIPO_PACIENTEN_id_tipo NUMBER NOT NULL ,
    rut_d_paciente       VARCHAR2 (1) NOT NULL
)
;

ALTER TABLE PACIENTE
ADD CONSTRAINT CK_PACIENTE_RU_D
CHECK (rut_d_paciente IN ('"0"', '"1"', '"2"', '"3"', '"4"', '"5"', '"6"', '"7"', '"8"', '"9"', '"K"', '"k"'))
;
CREATE UNIQUE INDEX PACIENTE__IDX ON PACIENTE
(
    FICHA_id_ficha ASC
)
;

```

Exp2\_S5\_Jorge\_Gallardo\_DDL: Bloc de notas

Archivo Edición Formato Ver Ayuda

```

CREATE UNIQUE INDEX PACIENTE__IDX ON PACIENTE
(
    FICHA_id_ficha ASC
)
;
CREATE UNIQUE INDEX PACIENTE__IDXv1 ON PACIENTE
(
    TIPO_PACIENTEN_id_tipo ASC
)
;

ALTER TABLE PACIENTE
ADD CONSTRAINT PK_PACIENTE PRIMARY KEY ( id_paciente, COMUNA_id_comuna ) ;

CREATE TABLE REGION
(
    id_region NUMBER (3) NOT NULL ,
    nombre    VARCHAR2 (30) NOT NULL
)
;

ALTER TABLE REGION
ADD CONSTRAINT PK_REGION PRIMARY KEY ( id_region ) ;

ALTER TABLE REGION
ADD CONSTRAINT REGION_nombre_UN UNIQUE ( nombre ) ;

CREATE TABLE SALUD
(
    id_salud          NUMBER (3) NOT NULL ,
    nombre_salud     VARCHAR2 (30) NOT NULL
)
;

ALTER TABLE SALUD

```

```

Exp2_S5_Jorge_Gallardo_DDL: Bloc de notas
Archivo Edición Formato Ver Ayuda
ALTER TABLE SALUD
ADD CONSTRAINT SALUD_PK PRIMARY KEY ( id_salud );

CREATE TABLE TIPO_ATENCION
(
    id_atencion          NUMBER (3) NOT NULL ,
    nombre_atencion      VARCHAR2 (30) NOT NULL ,
    ATENCION_folio       NUMBER (5) NOT NULL ,
    ATENCION_FICHA_id_ficha NUMBER (3) NOT NULL
)
;
CREATE UNIQUE INDEX TIPO_ATENCION__IDX ON TIPO_ATENCION
(
    ATENCION_folio ASC ,
    ATENCION_FICHA_id_ficha ASC
)
;

ALTER TABLE TIPO_ATENCION
ADD CONSTRAINT PK_TIPO_ATENCION PRIMARY KEY ( id_atencion );

CREATE TABLE TIPO_EMPLEADO
(
    id_tipo_empleado NUMBER (3) NOT NULL ,
    nombre_tipo      VARCHAR2 (30) NOT NULL
)
;

ALTER TABLE TIPO_EMPLEADO
ADD CONSTRAINT TIPO_EMPLEADO_PK PRIMARY KEY ( id_tipo_empleado );

CREATE TABLE TIPO_PACIENTEN
(
    id_tipo          NUMBER NOT NULL ,
    nombre_tipo      VARCHAR2 (20) NOT NULL ,
    PACIENTE_id_paciente NUMBER (3) NOT NULL ,
    PACIENTE_COMUNA_id_comuna NUMBER (3) NOT NULL
)
;
CREATE UNIQUE INDEX TIPO_PACIENTEN__IDX ON TIPO_PACIENTEN
(
    PACIENTE_id_paciente ASC ,
    PACIENTE_COMUNA_id_comuna ASC
)
;

ALTER TABLE TIPO_PACIENTEN
ADD CONSTRAINT PK_TIPO_PACIENTEN PRIMARY KEY ( id_tipo );

CREATE TABLE TIPO_PAGO
(
    id_pago          NUMBER (3) NOT NULL ,
    nombre_tipo      VARCHAR2 (30) NOT NULL ,
    ATENCION_folio   NUMBER (5) NOT NULL ,
    ATENCION_FICHA_id_ficha NUMBER (3) NOT NULL
)
;
CREATE UNIQUE INDEX TIPO_PAGO__IDX ON TIPO_PAGO
(
    ATENCION_folio ASC ,
    ATENCION_FICHA_id_ficha ASC
)
;

ALTER TABLE TIPO_PAGO
ADD CONSTRAINT PK_TIPO_PAGO PRIMARY KEY ( id_pago );

```

```

Exp2_S5_Jorge_Gallardo_DDL: Bloc de notas
Archivo Edición Formato Ver Ayuda
CREATE TABLE TIPO_PACIENTEN
(
    id_tipo          NUMBER NOT NULL ,
    nombre_tipo      VARCHAR2 (20) NOT NULL ,
    PACIENTE_id_paciente NUMBER (3) NOT NULL ,
    PACIENTE_COMUNA_id_comuna NUMBER (3) NOT NULL
)
;
CREATE UNIQUE INDEX TIPO_PACIENTEN__IDX ON TIPO_PACIENTEN
(
    PACIENTE_id_paciente ASC ,
    PACIENTE_COMUNA_id_comuna ASC
)
;

ALTER TABLE TIPO_PACIENTEN
ADD CONSTRAINT PK_TIPO_PACIENTEN PRIMARY KEY ( id_tipo );

CREATE TABLE TIPO_PAGO
(
    id_pago          NUMBER (3) NOT NULL ,
    nombre_tipo      VARCHAR2 (30) NOT NULL ,
    ATENCION_folio   NUMBER (5) NOT NULL ,
    ATENCION_FICHA_id_ficha NUMBER (3) NOT NULL
)
;
CREATE UNIQUE INDEX TIPO_PAGO__IDX ON TIPO_PAGO
(
    ATENCION_folio ASC ,
    ATENCION_FICHA_id_ficha ASC
)
;

ALTER TABLE TIPO_PAGO
ADD CONSTRAINT PK_TIPO_PAGO PRIMARY KEY ( id_pago );

```

```

Exp2_S5_Jorge_Gallardo_DDL: Bloc de notas
Archivo Edición Formato Ver Ayuda
ALTER TABLE TIPO_PAGO
ADD CONSTRAINT PK_TIPO_PAGO PRIMARY KEY ( id_pago );

CREATE TABLE UNIDAD
(
    id_unidad          NUMBER (3) NOT NULL ,
    nombre_unidad      VARCHAR2 (20) NOT NULL ,
    CENTRO_MEDICO_id_centro NUMBER (3) NOT NULL
)
;

ALTER TABLE UNIDAD
ADD CONSTRAINT PK_UNIDAD PRIMARY KEY ( id_unidad );

ALTER TABLE ATENCION
ADD CONSTRAINT ATENCION_EMPLEADO_FK FOREIGN KEY
(
    EMPLEADO_id_empleado
)
REFERENCES EMPLEADO
(
    id_empleado
)
;

ALTER TABLE ATENCION
ADD CONSTRAINT ATENCION_EXAMEN_FK FOREIGN KEY
(
    EXAMEN_codigo_examen
)
REFERENCES EXAMEN
(
    codigo_examen
)
;

```

Linea 1, columna 1 100% Windows (CRLF) UTF-8 23°C 16:04 11-09-2025

```

Exp2_S5_Jorge_Gallardo_DDL: Bloc de notas
Archivo Edición Formato Ver Ayuda

ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_AFP_FK FOREIGN KEY
(
    AFP_id_afp
)
REFERENCES AFP
(
    id_afp
)
;

ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_CENTRO_MEDICO_FK FOREIGN KEY
(
    CENTRO_MEDICO_id_centro
)
REFERENCES CENTRO_MEDICO
(
    id_centro
)
;

ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_EMPLEADO_FK FOREIGN KEY
(
    EMPLEADO_id_empleado
)
REFERENCES EMPLEADO
(
    id_empleado
)
;

ALTER TABLE EMPLEADO

```

Linea 1, columna 1 100% Windows (CRLF) UTF-8 23°C 16:09 11-09-2025

Exp2\_S5\_Jorge\_Gallardo\_DDL: Bloc de notas  
Archivo Edición Formato Ver Ayuda

```
ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_ESPECIALIDAD_FK FOREIGN KEY
(
    ESPECIALIDAD_id_especialidad
)
REFERENCES ESPECIALIDAD
(
    id_especialidad
)
;

ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_SALUD_FK FOREIGN KEY
(
    SALUD_id_salud
)
REFERENCES SALUD
(
    id_salud
)
;

ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_TIPO_EMPLEADO_FK FOREIGN KEY
(
    TIPO_EMPLEADO_id_tipo_empleado
)
REFERENCES TIPO_EMPLEADO
(
    id_tipo_empleado
)
;

ALTER TABLE EMPLEADO
```

Exp2\_S5\_Jorge\_Gallardo\_DDL: Bloc de notas  
Archivo Edición Formato Ver Ayuda

```
ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_ESPECIALIDAD_FK FOREIGN KEY
(
    ESPECIALIDAD_id_especialidad
)
REFERENCES ESPECIALIDAD
(
    id_especialidad
)
;

ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_SALUD_FK FOREIGN KEY
(
    SALUD_id_salud
)
REFERENCES SALUD
(
    id_salud
)
;

ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_TIPO_EMPLEADO_FK FOREIGN KEY
(
    TIPO_EMPLEADO_id_tipo_empleado
)
REFERENCES TIPO_EMPLEADO
(
    id_tipo_empleado
)
;

ALTER TABLE EMPLEADO
```

Exp2\_S5\_Jorge\_Gallardo\_DDL: Bloc de notas  
Archivo Edición Formato Ver Ayuda

```
ALTER TABLE EMPLEADO
ADD CONSTRAINT EMPLEADO_UNIDAD_FK FOREIGN KEY
(
    UNIDAD_id_unidad
)
REFERENCES UNIDAD
(
    id_unidad
)
;

ALTER TABLE FICHA
ADD CONSTRAINT FICHA_PACIENTE_FK FOREIGN KEY
(
    PACIENTE_id_paciente,
    PACIENTE_COMUNA_id_comuna
)
REFERENCES PACIENTE
(
    id_paciente,
    COMUNA_id_comuna
)
;

ALTER TABLE PACIENTE
ADD CONSTRAINT PACIENTE_CENTRO_MEDICO_FK FOREIGN KEY
(
    CENTRO_MEDICO_id_centro
)
REFERENCES CENTRO_MEDICO
(
    id_centro
)
;
```

Exp2\_S5\_Jorge\_Gallardo\_DDL: Bloc de notas  
Archivo Edición Formato Ver Ayuda

```
ALTER TABLE PACIENTE
ADD CONSTRAINT PACIENTE_COMUNA_FK FOREIGN KEY
(
    COMUNA_id_comuna
)
REFERENCES COMUNA
(
    id_comuna
)
;

ALTER TABLE PACIENTE
ADD CONSTRAINT PACIENTE_FICHA_FK FOREIGN KEY
(
    FICHA_id_ficha
)
REFERENCES FICHA
(
    id_ficha
)
;

ALTER TABLE PACIENTE
ADD CONSTRAINT PACIENTE_TIPO_PACIENTEN_FK FOREIGN KEY
(
    TIPO_PACIENTEN_id_tipo
)
REFERENCES TIPO_PACIENTEN
(
    id_tipo
)
;

ALTER TABLE TIPO_ATENCION
ADD CONSTRAINT TIPO_ATENCION_ATENCION_FK FOREIGN KEY
```



```

Exp2_S5_Jorge_Gallardo_DDL: Bloc de notas
Archivo Edición Formato Ver Ayuda
ALTER TABLE TIPO_ATENCION
ADD CONSTRAINT TIPO_ATENCION_ATENCION_FK FOREIGN KEY
(
    ATENCION_folio,
    ATENCION_FICHA_id_ficha
)
REFERENCES ATENCION
(
    folio,
    FICHA_id_ficha
)
;

ALTER TABLE TIPO_PACIENTEN
ADD CONSTRAINT TIPO_PACIENTEN_PACIENTE_FK FOREIGN KEY
(
    PACIENTE_id_paciente,
    PACIENTE_COMUNA_id_comuna
)
REFERENCES PACIENTE
(
    id_paciente,
    COMUNA_id_comuna
)
;

ALTER TABLE TIPO_PAGO
ADD CONSTRAINT TIPO_PAGO_ATENCION_FK FOREIGN KEY
(
    ATENCION_folio,
    ATENCION_FICHA_id_ficha
)
REFERENCES ATENCION
(
    folio,
    FICHA_id_ficha
)
;

```

```

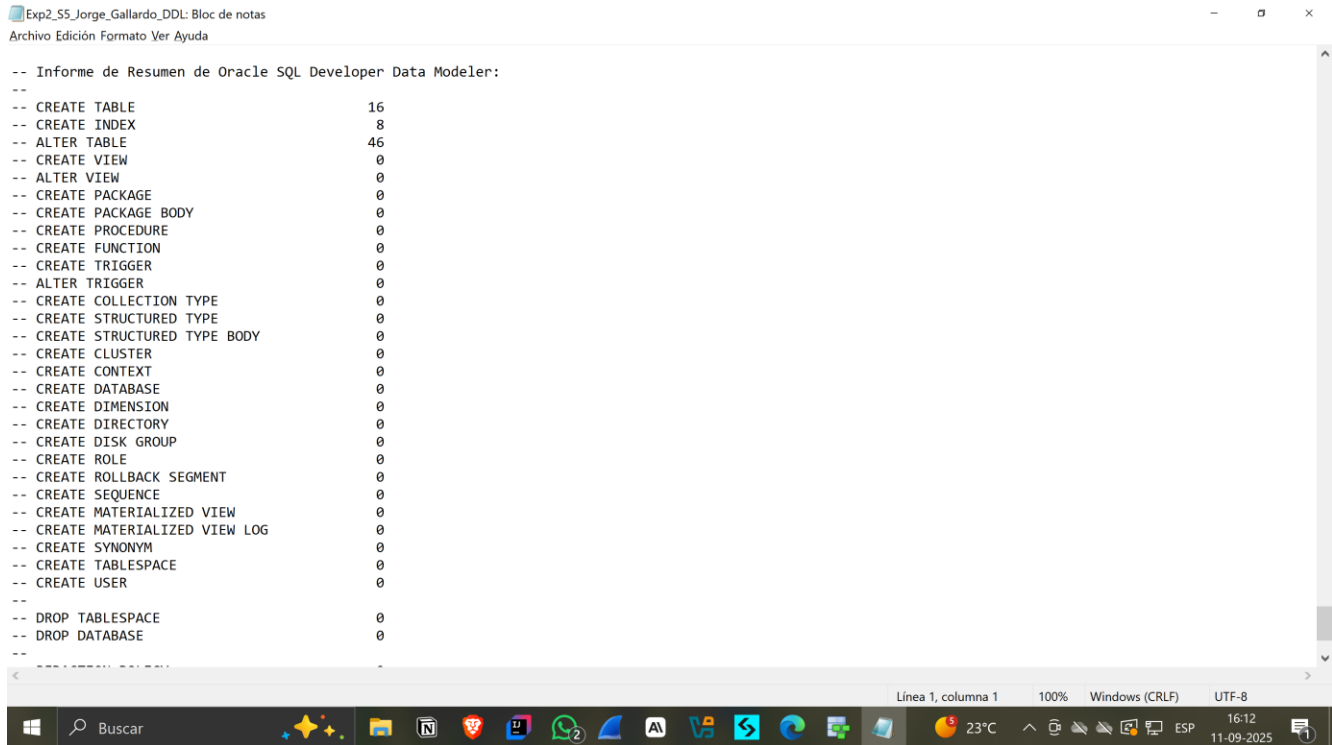
Exp2_S5_Jorge_Gallardo_DDL: Bloc de notas
Archivo Edición Formato Ver Ayuda
;

ALTER TABLE TIPO_PAGO
ADD CONSTRAINT TIPO_PAGO_ATENCION_FK FOREIGN KEY
(
    ATENCION_folio,
    ATENCION_FICHA_id_ficha
)
REFERENCES ATENCION
(
    folio,
    FICHA_id_ficha
)
;

ALTER TABLE UNIDAD
ADD CONSTRAINT UNIDAD_CENTRO_MEDICO_FK FOREIGN KEY
(
    CENTRO_MEDICO_id_centro
)
REFERENCES CENTRO_MEDICO
(
    id_centro
)
;

-- Informe de Resumen de Oracle SQL Developer Data Modeler:
--
-- CREATE TABLE          16
-- CREATE INDEX            8
-- ALTER TABLE           46
-- CREATE VIEW              0
-- ALTER VIEW              0
-- CREATE SEQUENCE         0

```



Exp2\_S5\_Jorge\_Gallardo\_DDL: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
-- Informe de Resumen de Oracle SQL Developer Data Modeler:
--
-- CREATE TABLE                      16
-- CREATE INDEX                      8
-- ALTER TABLE                     46
-- CREATE VIEW                       0
-- ALTER VIEW                       0
-- CREATE PACKAGE                    0
-- CREATE PACKAGE BODY               0
-- CREATE PROCEDURE                  0
-- CREATE FUNCTION                   0
-- CREATE TRIGGER                    0
-- 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
--
```

Linea 1, columna 1 100% Windows (CRLF) UTF-8

23°C 16:12 11-09-2025

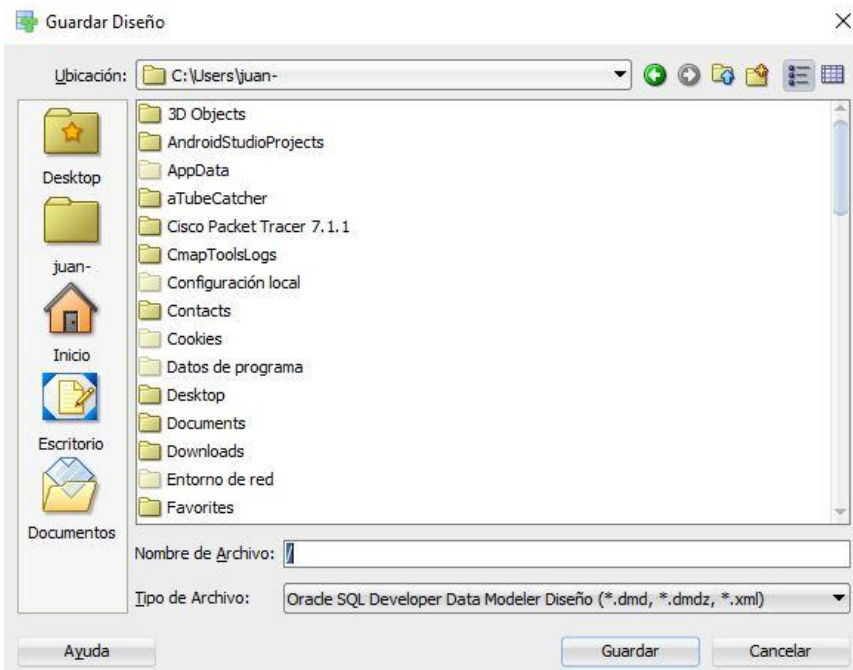
## Paso 2:

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)*

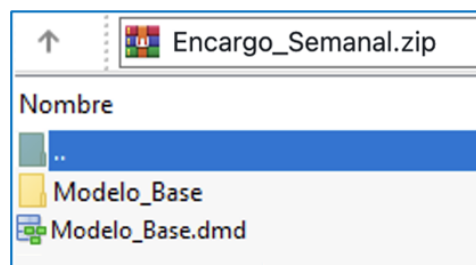
Nombre	Fecha de modificación	Tipo
businessinfo	03/10/2024 15:55	Carpeta de archivos
datatypes	03/10/2024 15:55	Carpeta de archivos
files	03/10/2024 15:55	Carpeta de archivos
logical	03/10/2024 15:55	Carpeta de archivos
mapping	03/10/2024 15:55	Carpeta de archivos
pm	03/10/2024 15:55	Carpeta de archivos
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*



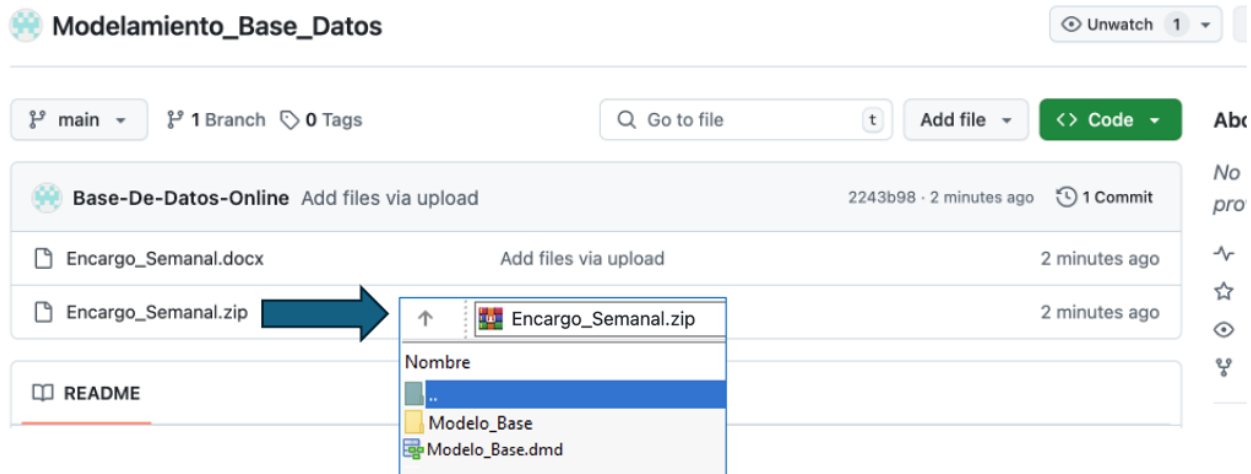
*Nota.* La figura muestra la estructura interna del archivo comprimido Encargo\_Semanal.zip. Oracle. (s.f.). *Oracle SQL Developer Data Modeler* [Software]. [Oracle.com](https://www.oracle.com)

**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*



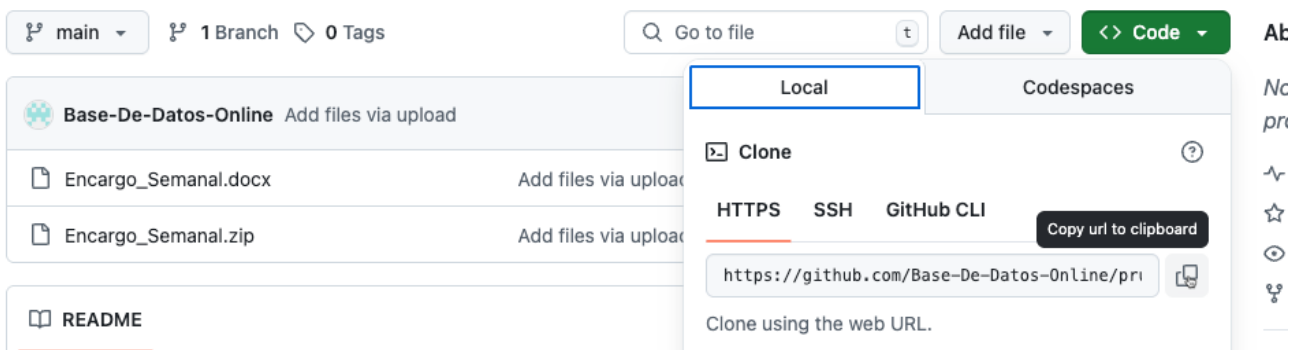
*Nota.* Ejemplo genérico de archivos cargados en el repositorio GitHub. GitHub (s.f.). *GitHub*.

<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*.

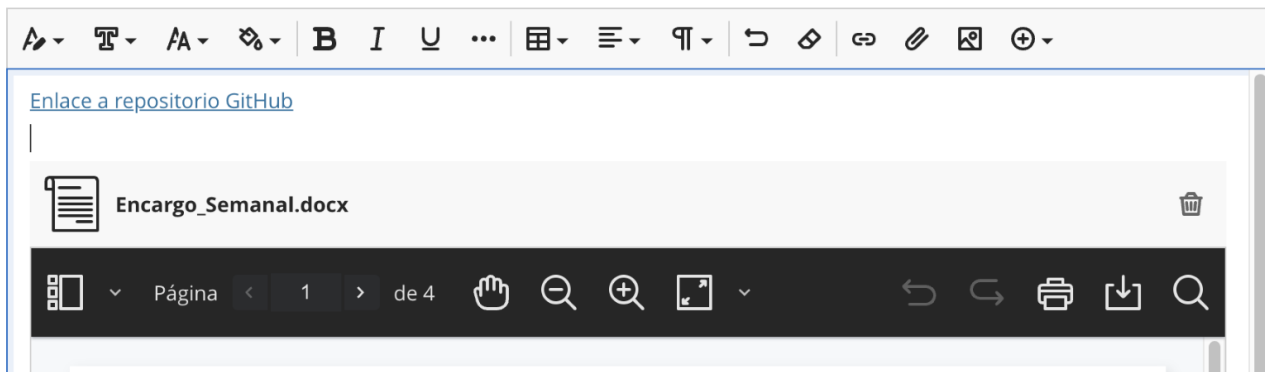
<https://github.com/>

**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.