

## Semana 5

# Modelamiento de Bases de Datos

Formato de respuesta

<b>Nombre estudiante: Eduardo Stegmaier</b>	
<b>Asignatura: Modelamiento de Bases de Datos</b>	<b>Carrera: Analista Programador Computacional</b>
<b>Profesor: Manuel Abarca</b>	<b>Fecha: 15/09/25</b>

## 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.

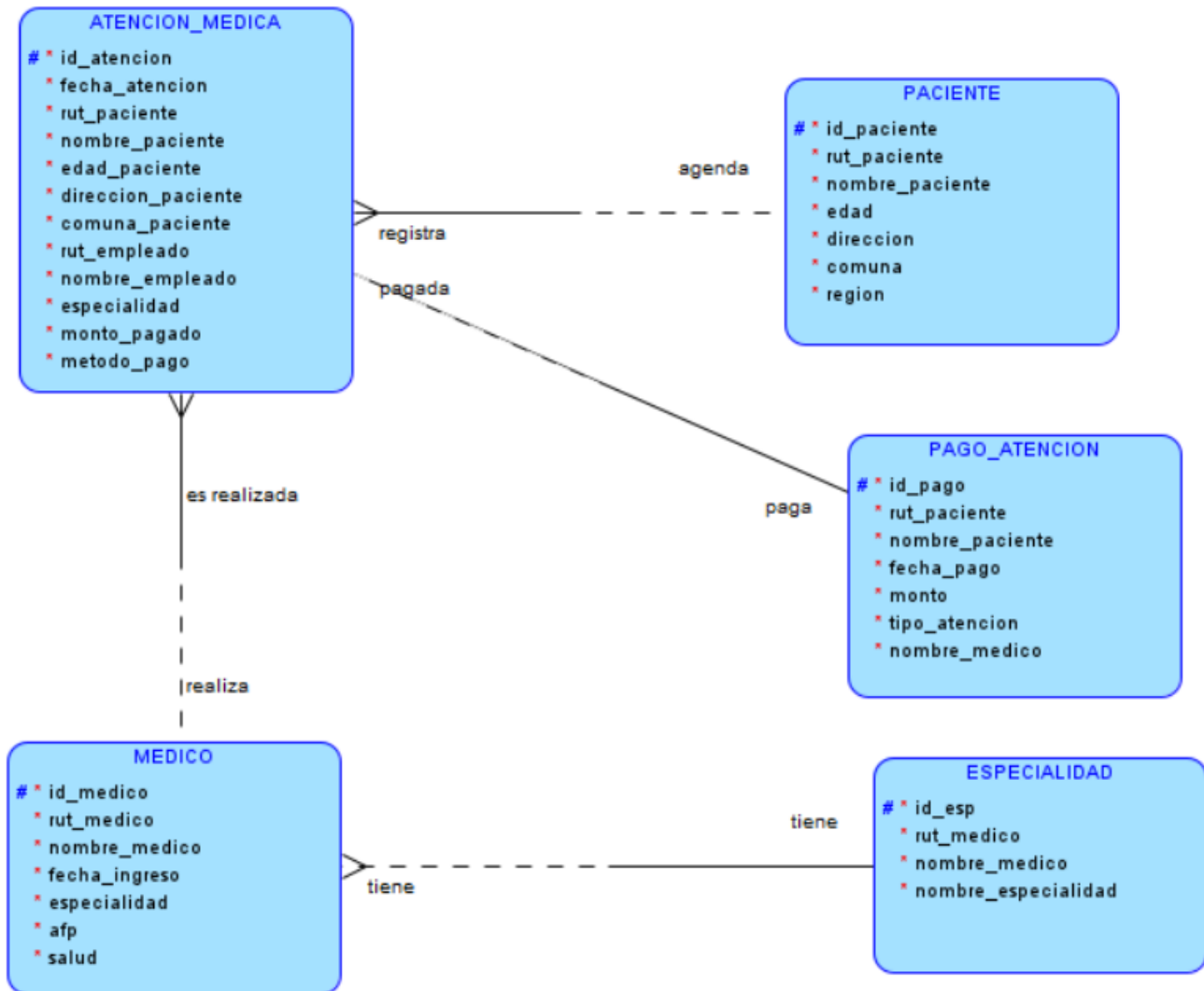
Actualmente, la información se gestiona en una hoja de cálculo donde cada fila representa una atención completa. Esta estructura plana provoca duplicación de datos, errores de ingreso y dificultades en el seguimiento clínico.

### **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:

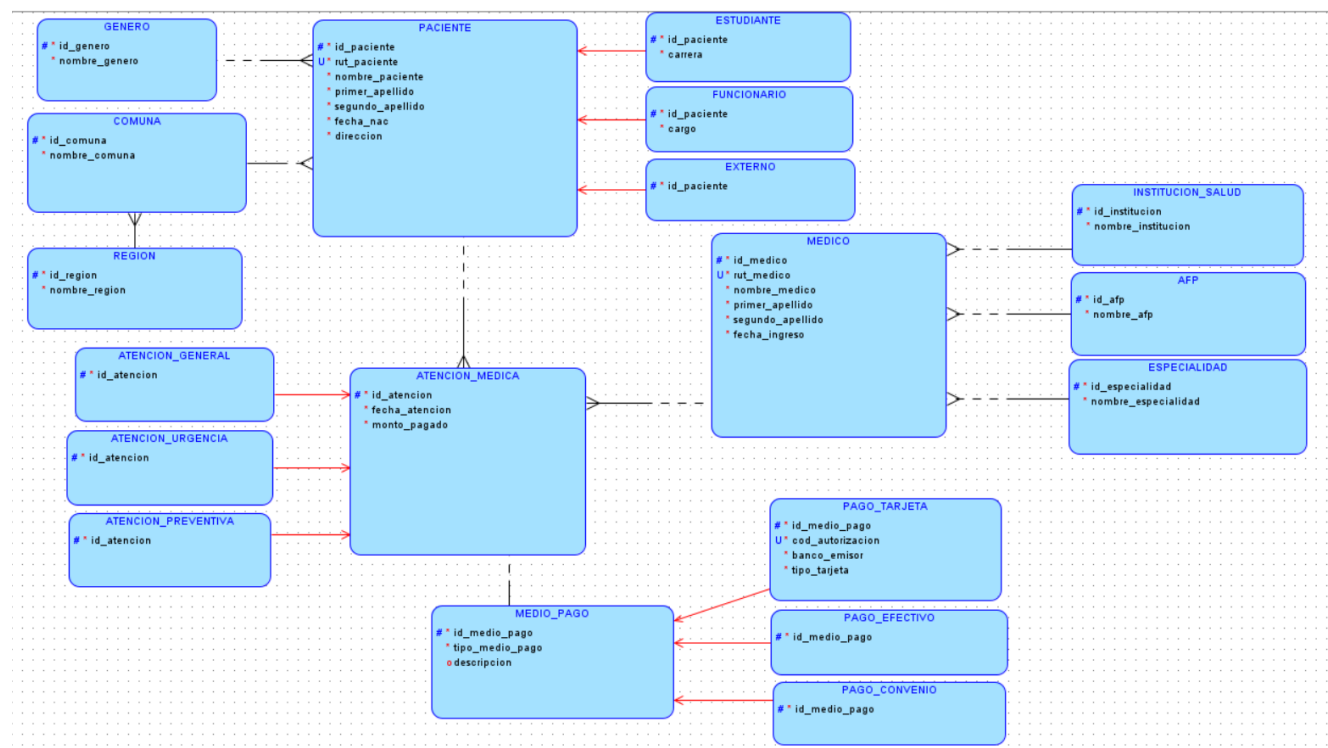
- Modelo Entidad-Relación-Extendido (MER-E) Normalizado solicitado en notación Barker.
- Modelo Relacional (MR) Normalizado.
- Script DDL generado con la herramienta Oracle Data Modeler.

Ejemplos:

Modelo MER-E Normalizado en notación Barker	Modelo Relacional Normalizado	Script DDL
		<pre> 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 117 118 119 CREATE TABLE receta ( 120     id_receta     NUMBER(8) NOT NULL, 121     observacion   VARCHAR2(30), 122     digitador_id_digitador NUMBER(5) NOT NULL, 123     boleta_n_boleta NUMBER(5) NOT NULL, 124     diagnostico_cod_diag NUMBER(4) NOT NULL, 125     medico_id_medico NUMBER(6) NOT NULL, 126     paciente_rut  NUMBER(8) NOT NULL, 127     tipo_receta_tipo_receta_id NUMBER NOT NULL 128 );         </pre>

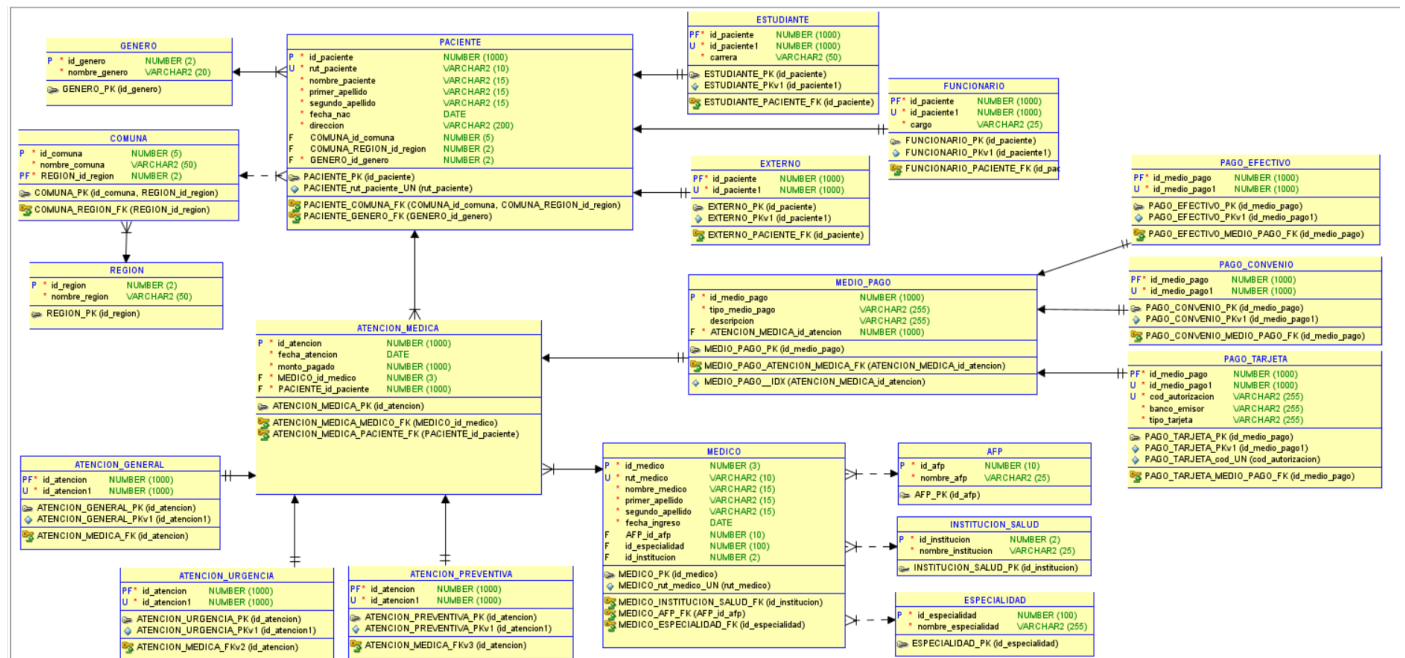
Adjunta tus evidencias en esta parte:

### 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 17:21:10 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 (10)  NOT NULL ,
    nombre_afp VARCHAR2 (25)  NOT NULL
)
;

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

CREATE TABLE ATENCION_GENERAL
(
    id_atencion  NUMBER (1000) NOT NULL ,
    id_atencion1 NUMBER (1000) NOT NULL
)
;

ALTER TABLE ATENCION_GENERAL
    ADD CONSTRAINT ATENCION_GENERAL_PK PRIMARY KEY ( id_atencion ) ;

ALTER TABLE ATENCION_GENERAL
    ADD CONSTRAINT ATENCION_GENERAL_PKv1 UNIQUE ( id_atencion1 ) ;

CREATE TABLE ATENCION_MEDICA
(
    id_atencion      NUMBER (1000) NOT NULL ,
    fecha_atencion   DATE NOT NULL ,
    monto_pagado      NUMBER (1000) NOT NULL ,
    MEDICO_id_medico  NUMBER (3)  NOT NULL ,
    PACIENTE_id_paciente NUMBER (1000) NOT NULL
)
;

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

CREATE TABLE ATENCION_PREVENTIVA
(
    id_atencion  NUMBER (1000) NOT NULL ,
    id_atencion1 NUMBER (1000) NOT NULL
)
;

ALTER TABLE ATENCION_PREVENTIVA
```

```

        ADD CONSTRAINT ATENCION_PREVENTIVA_PK PRIMARY KEY ( id_atencion ) ;

ALTER TABLE ATENCION_PREVENTIVA
    ADD CONSTRAINT ATENCION_PREVENTIVA_PKv1 UNIQUE ( id_atencion1 ) ;

CREATE TABLE ATENCION_URGENCIA
(
    id_atencion NUMBER (1000) NOT NULL ,
    id_atencion1 NUMBER (1000) NOT NULL
)
;

ALTER TABLE ATENCION_URGENCIA
    ADD CONSTRAINT ATENCION_URGENCIA_PK PRIMARY KEY ( id_atencion ) ;

ALTER TABLE ATENCION_URGENCIA
    ADD CONSTRAINT ATENCION_URGENCIA_PKv1 UNIQUE ( id_atencion1 ) ;

CREATE TABLE COMUNA
(
    id_comuna NUMBER (5) NOT NULL ,
    nombre_comuna VARCHAR2 (50) NOT NULL ,
    REGION_id_region NUMBER (2) NOT NULL
)
;

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

CREATE TABLE ESPECIALIDAD
(
    id_especialidad NUMBER (100) NOT NULL ,
    nombre_especialidad VARCHAR2 (255) NOT NULL
)
;

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

CREATE TABLE ESTUDIANTE
(
    id_paciente NUMBER (1000) NOT NULL ,
    id_paciente1 NUMBER (1000) NOT NULL ,
    carrera VARCHAR2 (50) NOT NULL
)
;

ALTER TABLE ESTUDIANTE
    ADD CONSTRAINT ESTUDIANTE_PK PRIMARY KEY ( id_paciente ) ;

ALTER TABLE ESTUDIANTE
    ADD CONSTRAINT ESTUDIANTE_PKv1 UNIQUE ( id_paciente1 ) ;

CREATE TABLE EXTERNO

```

```

(
    id_paciente NUMBER (1000) NOT NULL ,
    id_pacientel NUMBER (1000) NOT NULL
)
;

ALTER TABLE EXTERNO
    ADD CONSTRAINT EXTERNO_PK PRIMARY KEY ( id_paciente ) ;

ALTER TABLE EXTERNO
    ADD CONSTRAINT EXTERNO_PKv1 UNIQUE ( id_pacientel ) ;

CREATE TABLE FUNCIONARIO
(
    id_paciente NUMBER (1000) NOT NULL ,
    id_pacientel NUMBER (1000) NOT NULL ,
    cargo VARCHAR2 (25) NOT NULL
)
;

ALTER TABLE FUNCIONARIO
    ADD CONSTRAINT FUNCIONARIO_PK PRIMARY KEY ( id_paciente ) ;

ALTER TABLE FUNCIONARIO
    ADD CONSTRAINT FUNCIONARIO_PKv1 UNIQUE ( id_pacientel ) ;

CREATE TABLE GENERO
(
    id_genero NUMBER (2) NOT NULL ,
    nombre_genero VARCHAR2 (20) NOT NULL
)
;

ALTER TABLE GENERO
    ADD CONSTRAINT GENERO_PK PRIMARY KEY ( id_genero ) ;

CREATE TABLE INSTITUCION_SALUD
(
    id_institucion NUMBER (2) NOT NULL ,
    nombre_institucion VARCHAR2 (25) NOT NULL
)
;

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

CREATE TABLE MEDICO
(
    id_medico NUMBER (3) NOT NULL ,
    rut_medico VARCHAR2 (10) NOT NULL ,
    nombre_medico VARCHAR2 (15) NOT NULL ,
    primer_apellido VARCHAR2 (15) NOT NULL ,
    segundo_apellido VARCHAR2 (15) NOT NULL ,

```

```

        fecha_ingreso    DATE NOT NULL ,
        AFP_id_afp        NUMBER (10) ,
        id_especialidad   NUMBER (100) ,
        id_institucion    NUMBER (2)
    )
;

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

ALTER TABLE MEDICO
    ADD CONSTRAINT MEDICO_rut_medico_UN UNIQUE ( rut_medico ) ;

CREATE TABLE MEDIO_PAGO
(
    id_medio_pago          NUMBER (1000) NOT NULL ,
    tipo_medio_pago        VARCHAR2 (255) NOT NULL ,
    descripcion            VARCHAR2 (255) ,
    ATENCION_MEDICA_id_atencion NUMBER (1000) NOT NULL
)
;

CREATE UNIQUE INDEX MEDIO_PAGO__IDX ON MEDIO_PAGO
(
    ATENCION_MEDICA_id_atencion ASC
)
;

ALTER TABLE MEDIO_PAGO
    ADD CONSTRAINT MEDIO_PAGO_PK PRIMARY KEY ( id_medio_pago ) ;

CREATE TABLE PACIENTE
(
    id_paciente            NUMBER (1000) NOT NULL ,
    rut_paciente           VARCHAR2 (10) NOT NULL ,
    nombre_paciente        VARCHAR2 (15) NOT NULL ,
    primer_apellido        VARCHAR2 (15) NOT NULL ,
    segundo_apellido       VARCHAR2 (15) NOT NULL ,
    fecha_nac              DATE NOT NULL ,
    direccion              VARCHAR2 (200) NOT NULL ,
    COMUNA_id_comuna       NUMBER (5) ,
    COMUNA_REGION_id_region NUMBER (2) ,
    GENERO_id_genero       NUMBER (2) NOT NULL
)
;

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

ALTER TABLE PACIENTE
    ADD CONSTRAINT PACIENTE_rut_paciente_UN UNIQUE ( rut_paciente ) ;

CREATE TABLE PAGO_CONVENIO
(

```

```

        id_medio_pago NUMBER (1000) NOT NULL ,
        id_medio_pago1 NUMBER (1000) NOT NULL
    )
;

ALTER TABLE PAGO_CONVENIO
    ADD CONSTRAINT PAGO_CONVENIO_PK PRIMARY KEY ( id_medio_pago ) ;

ALTER TABLE PAGO_CONVENIO
    ADD CONSTRAINT PAGO_CONVENIO_PKv1 UNIQUE ( id_medio_pago1 ) ;

CREATE TABLE PAGO_EFECTIVO
(
    id_medio_pago NUMBER (1000) NOT NULL ,
    id_medio_pago1 NUMBER (1000) NOT NULL
)
;

ALTER TABLE PAGO_EFECTIVO
    ADD CONSTRAINT PAGO_EFECTIVO_PK PRIMARY KEY ( id_medio_pago ) ;

ALTER TABLE PAGO_EFECTIVO
    ADD CONSTRAINT PAGO_EFECTIVO_PKv1 UNIQUE ( id_medio_pago1 ) ;

CREATE TABLE PAGO_TARJETA
(
    id_medio_pago NUMBER (1000) NOT NULL ,
    id_medio_pago1 NUMBER (1000) NOT NULL ,
    cod_autorizacion VARCHAR2 (255) NOT NULL ,
    banco_emisor VARCHAR2 (255) NOT NULL ,
    tipo_tarjeta VARCHAR2 (255) NOT NULL
)
;

ALTER TABLE PAGO_TARJETA
    ADD CONSTRAINT PAGO_TARJETA_PK PRIMARY KEY ( id_medio_pago ) ;

ALTER TABLE PAGO_TARJETA
    ADD CONSTRAINT PAGO_TARJETA_cod_UN UNIQUE ( cod_autorizacion ) ;

ALTER TABLE PAGO_TARJETA
    ADD CONSTRAINT PAGO_TARJETA_PKv1 UNIQUE ( id_medio_pago1 ) ;

CREATE TABLE REGION
(
    id_region NUMBER (2) NOT NULL ,
    nombre_region VARCHAR2 (50) NOT NULL
)
;

ALTER TABLE REGION
    ADD CONSTRAINT REGION_PK PRIMARY KEY ( id_region ) ;
ALTER TABLE ATENCION_GENERAL

```

```

ADD CONSTRAINT ATENCION_MEDICA_FK FOREIGN KEY
(
    id_atencion
)
REFERENCES ATENCION_MEDICA
(
    id_atencion
)
;

ALTER TABLE ATENCION_URGENCIA
ADD CONSTRAINT ATENCION_MEDICA_FKv2 FOREIGN KEY
(
    id_atencion
)
REFERENCES ATENCION_MEDICA
(
    id_atencion
)
;

ALTER TABLE ATENCION_PREVENTIVA
ADD CONSTRAINT ATENCION_MEDICA_FKv3 FOREIGN KEY
(
    id_atencion
)
REFERENCES ATENCION_MEDICA
(
    id_atencion
)
;

ALTER TABLE ATENCION_MEDICA
ADD CONSTRAINT ATENCION_MEDICA_MEDICO_FK FOREIGN KEY
(
    MEDICO_id_medico
)
REFERENCES MEDICO
(
    id_medico
)
;

ALTER TABLE ATENCION_MEDICA
ADD CONSTRAINT ATENCION_MEDICA_PACIENTE_FK FOREIGN KEY
(
    PACIENTE_id_paciente
)
REFERENCES PACIENTE
(
    id_paciente
)
;

```

```

ALTER TABLE COMUNA
  ADD CONSTRAINT COMUNA_REGION_FK FOREIGN KEY
  (
    REGION_id_region
  )
  REFERENCES REGION
  (
    id_region
  )
;

ALTER TABLE ESTUDIANTE
  ADD CONSTRAINT ESTUDIANTE_PACIENTE_FK FOREIGN KEY
  (
    id_paciente
  )
  REFERENCES PACIENTE
  (
    id_paciente
  )
;

ALTER TABLE EXTERNO
  ADD CONSTRAINT EXTERNO_PACIENTE_FK FOREIGN KEY
  (
    id_paciente
  )
  REFERENCES PACIENTE
  (
    id_paciente
  )
;

ALTER TABLE FUNCIONARIO
  ADD CONSTRAINT FUNCIONARIO_PACIENTE_FK FOREIGN KEY
  (
    id_paciente
  )
  REFERENCES PACIENTE
  (
    id_paciente
  )
;

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
    (
      id_especialidad
    )
  REFERENCES ESPECIALIDAD
    (
      id_especialidad
    )
;

ALTER TABLE MEDICO
  ADD CONSTRAINT MEDICO_INSTITUCION_SALUD_FK FOREIGN KEY
    (
      id_institucion
    )
  REFERENCES INSTITUCION_SALUD
    (
      id_institucion
    )
;

ALTER TABLE MEDIO_PAGO
  ADD CONSTRAINT MEDIO_PAGO_ATENCION_MEDICA_FK FOREIGN KEY
    (
      ATENCION_MEDICA_id_atencion
    )
  REFERENCES ATENCION_MEDICA
    (
      id_atencion
    )
;

ALTER TABLE PACIENTE
  ADD CONSTRAINT PACIENTE_COMUNA_FK FOREIGN KEY
    (
      COMUNA_id_comuna,
      COMUNA_REGION_id_region
    )
  REFERENCES COMUNA
    (
      id_comuna,
      REGION_id_region
    )
;

ALTER TABLE PACIENTE
  ADD CONSTRAINT PACIENTE_GENERO_FK FOREIGN KEY
    (
      GENERO_id_genero
    )

```

```

REFERENCES GENERO
(
    id_genero
)
;

ALTER TABLE PAGO_CONVENIO
ADD CONSTRAINT PAGO_CONVENIO_MEDIO_PAGO_FK FOREIGN KEY
(
    id_medio_pago
)
REFERENCES MEDIO_PAGO
(
    id_medio_pago
)
;

ALTER TABLE PAGO_EFECTIVO
ADD CONSTRAINT PAGO_EFECTIVO_MEDIO_PAGO_FK FOREIGN KEY
(
    id_medio_pago
)
REFERENCES MEDIO_PAGO
(
    id_medio_pago
)
;

ALTER TABLE PAGO_TARJETA
ADD CONSTRAINT PAGO_TARJETA_MEDIO_PAGO_FK FOREIGN KEY
(
    id_medio_pago
)
REFERENCES MEDIO_PAGO
(
    id_medio_pago
)
;

```

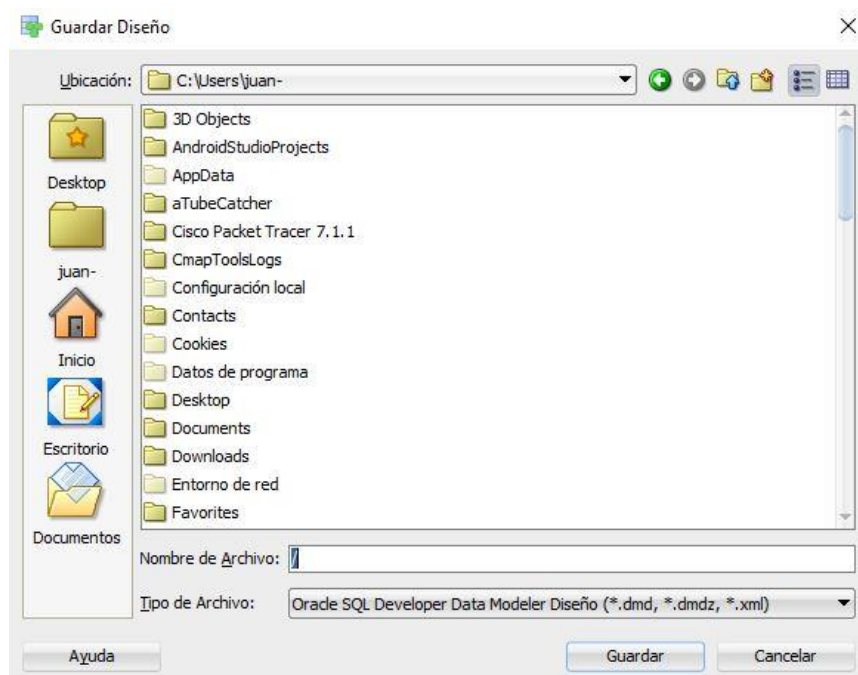
## Paso 2:

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)*

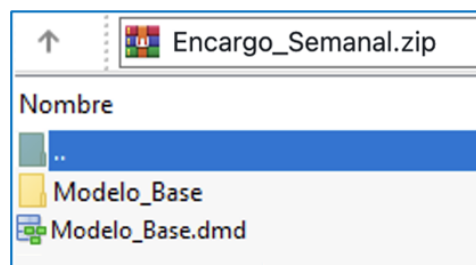
Nombre	Fecha de modificación	Tipo
businessinfo	03/10/2024 15:55	Carpeta de archivos
datatypes	03/10/2024 15:55	Carpeta de archivos
files	03/10/2024 15:55	Carpeta de archivos
logical	03/10/2024 15:55	Carpeta de archivos
mapping	03/10/2024 15:55	Carpeta de archivos
pm	03/10/2024 15:55	Carpeta de archivos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



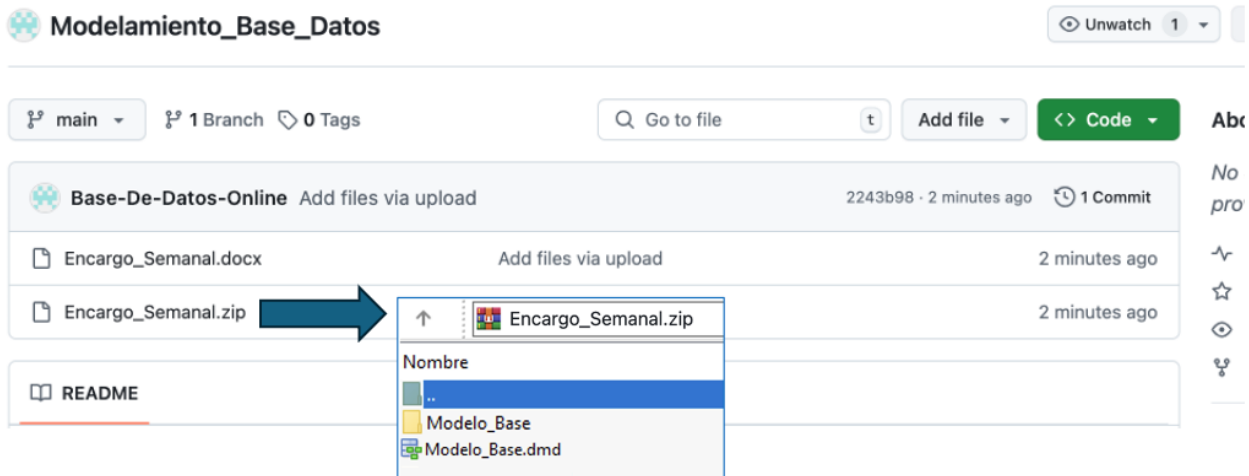
*Nota.* La figura muestra la estructura interna del archivo comprimido Encargo\_Semanal.zip. Oracle. (s.f.). *Oracle SQL Developer Data Modeler* [Software]. [Oracle.com](https://www.oracle.com)

**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*



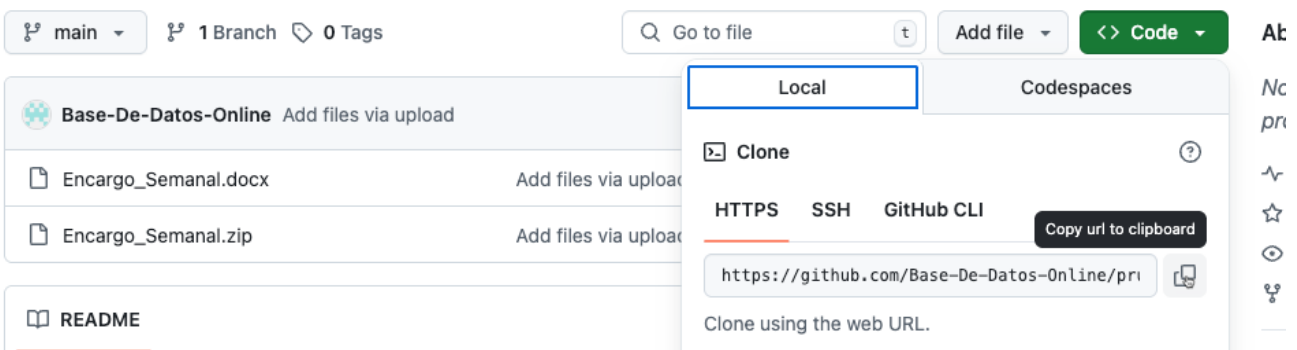
*Nota.* Ejemplo genérico de archivos cargados en el repositorio GitHub. GitHub (s.f.). *GitHub*.

<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*.

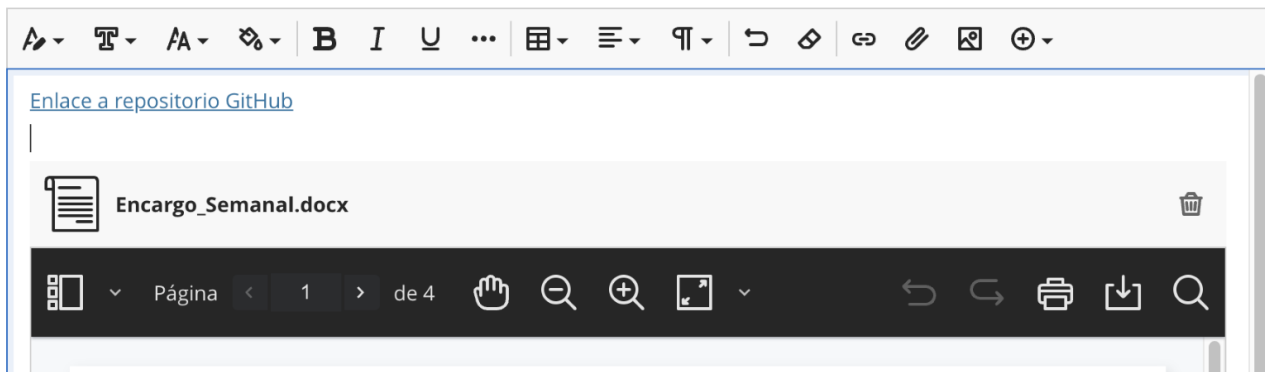
<https://github.com/>

**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.