

Semana 5

Modelamiento de Bases de Datos

Formato de respuesta

| **Nombre estudiante: Antonio Muñoz Navarrete** |  |
| --- | --- |
| **Asignatura: Modelamiento Base de Datos** | **Carrera: Analista Programador**  **Computacional Online** |
| **Profesor:** **Rodrigo Opazo Salazar** | **Fecha:** **13-09-2025** |

# 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/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:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Modelo Relacional (MR) Normalizado:**

Diagrama

El contenido generado por IA puede ser incorrecto.

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

**-- Generado por Oracle SQL Developer Data Modeler 23.1.0.087.0806**

**-- en: 2025-09-13 23:16:57 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 NOT NULL,**

**nombre\_afp VARCHAR2(30) NOT NULL**

**);**

**ALTER TABLE afp ADD CONSTRAINT afp\_pk PRIMARY KEY ( id\_afp );**

**ALTER TABLE afp ADD CONSTRAINT afp\_nombre\_afp\_un UNIQUE ( nombre\_afp );**

**CREATE TABLE atencion\_medica (**

**id\_atencion NUMBER NOT NULL,**

**fecha\_atencion DATE NOT NULL,**

**medico\_id\_medico NUMBER,**

**paciente\_id\_paciente NUMBER,**

**solicitud\_id\_solicitud NUMBER NOT NULL**

**);**

**CREATE UNIQUE INDEX atencion\_medica\_\_idx ON**

**atencion\_medica (**

**solicitud\_id\_solicitud**

**ASC );**

**ALTER TABLE atencion\_medica ADD CONSTRAINT atencion\_medica\_pk PRIMARY KEY ( id\_atencion );**

**CREATE TABLE comuna (**

**id\_comuna NUMBER NOT NULL,**

**nombre\_comuna VARCHAR2(30) NOT NULL,**

**region\_id\_region NUMBER NOT NULL**

**);**

**ALTER TABLE comuna ADD CONSTRAINT comuna\_pk PRIMARY KEY ( id\_comuna );**

**ALTER TABLE comuna ADD CONSTRAINT comuna\_nombre\_comuna\_un UNIQUE ( nombre\_comuna );**

**CREATE TABLE especialidad (**

**id\_especialidad NUMBER NOT NULL,**

**nomb\_esp VARCHAR2(30) NOT NULL**

**);**

**ALTER TABLE especialidad ADD CONSTRAINT especialidad\_nomb\_esp\_un UNIQUE ( nomb\_esp );**

**CREATE TABLE examen (**

**id\_examen NUMBER NOT NULL,**

**nombre\_examen VARCHAR2(30) NOT NULL,**

**solicitud\_id\_solicitud NUMBER NOT NULL**

**);**

**ALTER TABLE examen ADD CONSTRAINT examen\_pk PRIMARY KEY ( id\_examen );**

**ALTER TABLE examen ADD CONSTRAINT examen\_nombre\_examen\_un UNIQUE ( nombre\_examen );**

**CREATE TABLE fonasa (**

**id\_inst NUMBER NOT NULL**

**);**

**ALTER TABLE fonasa ADD CONSTRAINT fonasa\_pk PRIMARY KEY ( id\_inst );**

**CREATE TABLE general (**

**id\_atencion NUMBER NOT NULL**

**);**

**ALTER TABLE general ADD CONSTRAINT general\_pk PRIMARY KEY ( id\_atencion );**

**CREATE TABLE institucion\_salud (**

**id\_inst NUMBER NOT NULL,**

**nomb\_inst VARCHAR2(30) NOT NULL**

**);**

**ALTER TABLE institucion\_salud ADD CONSTRAINT institucion\_salud\_pk PRIMARY KEY ( id\_inst );**

**ALTER TABLE institucion\_salud ADD CONSTRAINT institucion\_salud\_nomb\_inst\_un UNIQUE ( nomb\_inst );**

**CREATE TABLE isapre (**

**id\_inst NUMBER NOT NULL**

**);**

**ALTER TABLE isapre ADD CONSTRAINT isapre\_pk PRIMARY KEY ( id\_inst );**

**CREATE TABLE medico (**

**id\_medico NUMBER NOT NULL,**

**rut\_medico VARCHAR2(15) NOT NULL,**

**nombre\_medico VARCHAR2(30) NOT NULL,**

**fecha\_ingreso DATE NOT NULL,**

**institucion\_salud\_id\_inst NUMBER,**

**afp\_id\_afp NUMBER,**

**medico\_id\_medico NUMBER,**

**especialidad\_nomb\_esp VARCHAR2(30)**

**);**

**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 paciente (**

**id\_paciente NUMBER NOT NULL,**

**rut\_paciente VARCHAR2(15) NOT NULL,**

**nombre\_paciente VARCHAR2(30) NOT NULL,**

**fecha\_naciemiento\_paciente DATE NOT NULL,**

**direccion\_paciente VARCHAR2(50) NOT NULL,**

**comuna\_id\_comuna NUMBER,**

**tipo\_paciente VARCHAR2(15) NOT NULL,**

**email\_paciente VARCHAR2(50) NOT NULL**

**);**

**ALTER TABLE paciente**

**ADD CONSTRAINT ck\_paciente\_tipo\_paciente CHECK ( tipo\_paciente IN ( '"Estudiante"', '"Externo"', '"Funcionario"' ) );**

**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 paciente\_femenino (**

**id\_paciente NUMBER NOT NULL**

**);**

**ALTER TABLE paciente\_femenino ADD CONSTRAINT paciente\_femenino\_pk PRIMARY KEY ( id\_paciente );**

**CREATE TABLE paciente\_masculino (**

**id\_paciente NUMBER NOT NULL**

**);**

**ALTER TABLE paciente\_masculino ADD CONSTRAINT paciente\_masculino\_pk PRIMARY KEY ( id\_paciente );**

**CREATE TABLE pago (**

**id\_pago NUMBER NOT NULL,**

**monto NUMBER NOT NULL,**

**fecha\_pago DATE NOT NULL,**

**atencion\_medica\_id\_atencion NUMBER NOT NULL**

**);**

**ALTER TABLE pago ADD CONSTRAINT ck\_pago\_monto CHECK ( monto >= 0 );**

**ALTER TABLE pago ADD CONSTRAINT pago\_pk PRIMARY KEY ( id\_pago );**

**CREATE TABLE pago\_convenio (**

**id\_pago NUMBER NOT NULL,**

**id\_convenio NUMBER NOT NULL,**

**nomb\_conv VARCHAR2(30) NOT NULL**

**);**

**ALTER TABLE pago\_convenio ADD CONSTRAINT pago\_convenio\_pk PRIMARY KEY ( id\_pago );**

**ALTER TABLE pago\_convenio ADD CONSTRAINT pago\_convenio\_pkv1 UNIQUE ( id\_convenio );**

**ALTER TABLE pago\_convenio ADD CONSTRAINT pago\_convenio\_nomb\_conv\_un UNIQUE ( nomb\_conv );**

**CREATE TABLE pago\_credito (**

**id\_pago NUMBER NOT NULL,**

**cuotas NUMBER NOT NULL**

**);**

**ALTER TABLE pago\_credito ADD CONSTRAINT ck\_pago\_credito\_cuotas CHECK ( cuotas > 0 );**

**ALTER TABLE pago\_credito ADD CONSTRAINT pago\_credito\_pk PRIMARY KEY ( id\_pago );**

**CREATE TABLE pago\_debito (**

**id\_pago NUMBER NOT NULL**

**);**

**ALTER TABLE pago\_debito ADD CONSTRAINT pago\_debito\_pk PRIMARY KEY ( id\_pago );**

**CREATE TABLE pago\_efectivo (**

**id\_pago NUMBER NOT NULL**

**);**

**ALTER TABLE pago\_efectivo ADD CONSTRAINT pago\_efectivo\_pk PRIMARY KEY ( id\_pago );**

**CREATE TABLE pago\_tarjeta (**

**id\_pago NUMBER NOT NULL,**

**id\_tarjeta NUMBER NOT NULL,**

**nombre\_tarjeta VARCHAR2(30) NOT NULL**

**);**

**ALTER TABLE pago\_tarjeta ADD CONSTRAINT pago\_tarjeta\_pk PRIMARY KEY ( id\_pago );**

**ALTER TABLE pago\_tarjeta ADD CONSTRAINT pago\_tarjeta\_pkv1 UNIQUE ( id\_tarjeta );**

**CREATE TABLE preparacion (**

**id\_preparacion NUMBER NOT NULL,**

**nomb\_prep VARCHAR2(30) NOT NULL,**

**examen\_id\_examen NUMBER**

**);**

**ALTER TABLE preparacion ADD CONSTRAINT preparacion\_pk PRIMARY KEY ( id\_preparacion );**

**ALTER TABLE preparacion ADD CONSTRAINT preparacion\_nomb\_prep\_un UNIQUE ( nomb\_prep );**

**CREATE TABLE preventiva (**

**id\_atencion NUMBER NOT NULL**

**);**

**ALTER TABLE preventiva ADD CONSTRAINT preventiva\_pk PRIMARY KEY ( id\_atencion );**

**CREATE TABLE region (**

**id\_region NUMBER NOT NULL,**

**nombre\_region VARCHAR2(30) NOT NULL**

**);**

**ALTER TABLE region ADD CONSTRAINT region\_pk PRIMARY KEY ( id\_region );**

**ALTER TABLE region ADD CONSTRAINT region\_nombre\_region\_un UNIQUE ( nombre\_region );**

**CREATE TABLE solicitud (**

**id\_solicitud NUMBER NOT NULL**

**);**

**ALTER TABLE solicitud ADD CONSTRAINT solicitud\_pk PRIMARY KEY ( id\_solicitud );**

**CREATE TABLE urgencia (**

**id\_atencion NUMBER NOT NULL**

**);**

**ALTER TABLE urgencia ADD CONSTRAINT urgencia\_pk PRIMARY KEY ( 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 atencion\_medica**

**ADD CONSTRAINT atencion\_medica\_solicitud\_fk FOREIGN KEY ( solicitud\_id\_solicitud )**

**REFERENCES solicitud ( id\_solicitud );**

**ALTER TABLE comuna**

**ADD CONSTRAINT comuna\_region\_fk FOREIGN KEY ( region\_id\_region )**

**REFERENCES region ( id\_region );**

**ALTER TABLE examen**

**ADD CONSTRAINT examen\_solicitud\_fk FOREIGN KEY ( solicitud\_id\_solicitud )**

**REFERENCES solicitud ( id\_solicitud );**

**ALTER TABLE fonasa**

**ADD CONSTRAINT fonasa\_institucion\_salud\_fk FOREIGN KEY ( id\_inst )**

**REFERENCES institucion\_salud ( id\_inst );**

**ALTER TABLE general**

**ADD CONSTRAINT general\_atencion\_medica\_fk FOREIGN KEY ( id\_atencion )**

**REFERENCES atencion\_medica ( id\_atencion );**

**ALTER TABLE isapre**

**ADD CONSTRAINT isapre\_institucion\_salud\_fk FOREIGN KEY ( id\_inst )**

**REFERENCES institucion\_salud ( id\_inst );**

**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\_nomb\_esp )**

**REFERENCES especialidad ( nomb\_esp );**

**ALTER TABLE medico**

**ADD CONSTRAINT medico\_institucion\_salud\_fk FOREIGN KEY ( institucion\_salud\_id\_inst )**

**REFERENCES institucion\_salud ( id\_inst );**

**ALTER TABLE medico**

**ADD CONSTRAINT medico\_medico\_fk FOREIGN KEY ( medico\_id\_medico )**

**REFERENCES medico ( id\_medico );**

**ALTER TABLE paciente**

**ADD CONSTRAINT paciente\_comuna\_fk FOREIGN KEY ( comuna\_id\_comuna )**

**REFERENCES comuna ( id\_comuna );**

**ALTER TABLE paciente\_femenino**

**ADD CONSTRAINT paciente\_femenino\_paciente\_fk FOREIGN KEY ( id\_paciente )**

**REFERENCES paciente ( id\_paciente );**

**ALTER TABLE paciente\_masculino**

**ADD CONSTRAINT paciente\_masculino\_paciente\_fk FOREIGN KEY ( id\_paciente )**

**REFERENCES paciente ( id\_paciente );**

**ALTER TABLE pago**

**ADD CONSTRAINT pago\_atencion\_medica\_fk FOREIGN KEY ( atencion\_medica\_id\_atencion )**

**REFERENCES atencion\_medica ( id\_atencion );**

**ALTER TABLE pago\_convenio**

**ADD CONSTRAINT pago\_convenio\_pago\_fk FOREIGN KEY ( id\_pago )**

**REFERENCES pago ( id\_pago );**

**ALTER TABLE pago\_credito**

**ADD CONSTRAINT pago\_credito\_pago\_tarjeta\_fk FOREIGN KEY ( id\_pago )**

**REFERENCES pago\_tarjeta ( id\_pago );**

**ALTER TABLE pago\_debito**

**ADD CONSTRAINT pago\_debito\_pago\_tarjeta\_fk FOREIGN KEY ( id\_pago )**

**REFERENCES pago\_tarjeta ( id\_pago );**

**ALTER TABLE pago\_efectivo**

**ADD CONSTRAINT pago\_efectivo\_pago\_fk FOREIGN KEY ( id\_pago )**

**REFERENCES pago ( id\_pago );**

**ALTER TABLE pago\_tarjeta**

**ADD CONSTRAINT pago\_tarjeta\_pago\_fk FOREIGN KEY ( id\_pago )**

**REFERENCES pago ( id\_pago );**

**ALTER TABLE preparacion**

**ADD CONSTRAINT preparacion\_examen\_fk FOREIGN KEY ( examen\_id\_examen )**

**REFERENCES examen ( id\_examen );**

**ALTER TABLE preventiva**

**ADD CONSTRAINT preventiva\_atencion\_medica\_fk FOREIGN KEY ( id\_atencion )**

**REFERENCES atencion\_medica ( id\_atencion );**

**ALTER TABLE urgencia**

**ADD CONSTRAINT urgencia\_atencion\_medica\_fk FOREIGN KEY ( id\_atencion )**

**REFERENCES atencion\_medica ( id\_atencion );**

**-- 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**

**-- ERROR: No Discriminator Column found in Arc FKArc\_2 - constraint trigger for Arc cannot be generated**

**-- ERROR: No Discriminator Column found in Arc FKArc\_2 - constraint trigger for Arc cannot be generated**

**-- ERROR: No Discriminator Column found in Arc FKArc\_3 - constraint trigger for Arc cannot be generated**

**-- ERROR: No Discriminator Column found in Arc FKArc\_3 - constraint trigger for Arc cannot be generated**

**-- ERROR: No Discriminator Column found in Arc FKArc\_4 - constraint trigger for Arc cannot be generated**

**-- ERROR: No Discriminator Column found in Arc FKArc\_4 - constraint trigger for Arc cannot be generated**

**-- ERROR: No Discriminator Column found in Arc FKArc\_5 - constraint trigger for Arc cannot be generated**

**-- ERROR: No Discriminator Column found in Arc FKArc\_5 - constraint trigger for Arc cannot be generated**

**-- ERROR: No Discriminator Column found in Arc FKArc\_5 - constraint trigger for Arc cannot be generated**

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

**--**

**-- CREATE TABLE 24**

**-- CREATE INDEX 1**

**-- ALTER TABLE 62**

**-- 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**

**--**

**-- REDACTION POLICY 0**

**--**

**-- ORDS DROP SCHEMA 0**

**-- ORDS ENABLE SCHEMA 0**

**-- ORDS ENABLE OBJECT 0**

**--**

**-- ERRORS 12**

**-- 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/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.