

Semana 9

Modelamiento de Bases de Datos (PRY2204)

Formato de respuesta

| **Nombre estudiante:** | Carlos Alberto Barría Valdevenito |
| --- | --- |
| **Asignatura: Modelamiento BBDD** | **Carrera: Desarrollo de Aplicaciones** |
| **Profesor: Josue Nahum Oteiza Soto** | **Fecha: 6 Octubre 2024** |

# Descripción de la actividad

La Evaluación Final Transversal de la asignatura Modelamiento de Bases de Datos, consiste en la elaboración de un encargo de manera individual donde deberás entregar, a partir de un caso, el desarrollo desde el comienzo con el Modelo conceptual, modelo lógico y finalizar con un modelo relacional para brindar solución a los requerimientos del cliente. El producto de esta evaluación se dividirá en dos partes. Para el desarrollo y cumplimiento de la parte I, tendrás que entregar el modelo desarrollado en todas sus etapas con las herramientas Oracle SQL Developer y DBMS Base de Datos Oracle (en Oracle CLOUD) para la creación y poblamiento de las tablas del Modelo Relacional (MR) y adjuntar el Script de implementación.

En la parte II, deberás realizar la presentación de tu trabajo final mediante la grabación de un video. En ella, deberás mostrar y explicar detalladamente el proceso de desarrollo de tu proyecto, cómo funciona tu implementación y las características principales de lo realizado.

## Instrucciones específicas

Para el desarrollo de la EFT, tendrás que analizar el siguiente caso estudio:

“El Ministerio de las Culturas, las Artes y el Patrimonio, una entidad del Estado de Chile, desempeña un papel crucial en el diseño, formulación e implementación de políticas, planes y programas orientados al desarrollo cultural, artístico y patrimonial. Su misión es promover una evolución armónica y equitativa en estas áreas a lo largo de todo el territorio nacional.

Recientemente, se ha observado una disminución en el interés de la población juvenil por participar y preservar las tradiciones culturales chilenas, tales como la danza, la música folclórica, el desarrollo artístico indígena y la interpretación de instrumentos nativos. Esta tendencia amenaza con debilitar la rica herencia cultural del país y disminuir la diversidad de sus expresiones artísticas.

Ante esta situación, el Ministerio ha decidido tomar medidas proactivas para revitalizar el interés y la participación de los jóvenes en estas tradiciones culturales. La estrategia incluye potenciar los proyectos concursables disponibles para las academias de arte en cada región, con especial énfasis en las disciplinas de música y danza. El objetivo es mejorar la difusión de estas artes, generar campañas y actividades en los colegios, y trabajar para aumentar la calidad e infraestructura relacionada, todo con el fin de fortalecer la cultura como parte integral del desarrollo humano.

Se espera que, a través de estos esfuerzos, se logre no solo un aumento en la participación juvenil en las actividades culturales sino también una mejora en la calidad de vida y la salud mental de los jóvenes, especialmente en tiempos desafiantes.

La convocatoria lanzada por el Ministerio tiene como fin proporcionar financiamiento a proyectos que promuevan la valorización, promoción y fomento del arte popular, originario o de las culturas tradicionales de Chile. Las academias de artes interesadas en participar deben completar y enviar formularios oficiales con información detallada sobre diversos aspectos de su funcionamiento, incluyendo costos de personal, locaciones, insumos, permisos necesarios, seguridad, producción, información general de la academia, montajistas, entre otros.

Como especialista en datos del área de tecnologías de la información, se te ha encomendado la tarea de diseñar e implementar una base de datos que permita registrar y gestionar eficientemente toda la información relevante enviada por las Academias de Artes en los formularios oficiales. Este sistema de base de datos debe ser capaz de adaptarse a las necesidades dinámicas del Ministerio y las academias, facilitando la toma de decisiones y el seguimiento del impacto de los proyectos financiados.

**Las reglas de negocio y especificaciones técnicas son las siguientes:**

* Inicialmente, el sistema se centrará en las academias de Danza, Canto, e Intérprete de instrumentos musicales, con planes de expandirse a otras artes en el futuro.
* Las municipalidades pueden ofrecer espacios a múltiples academias, que a su vez, pueden impartir diversos cursos en diferentes turnos dentro del mismo recinto.
* Los profesores contratados pueden ser de planta o a honorarios, con diferencias significativas en términos de remuneración y obligaciones legales.
* Es vital registrar los turnos de trabajo de los profesores, dado que pueden laborar en más de una academia.
* Cada academia debe tener un director responsable, cuya posición puede cambiar con el tiempo.
* Para participar en los proyectos concursables, cada academia de arte deberá completar el formulario de postulación, proporcionando detalles vitales para su identificación y la evaluación de su elegibilidad para financiamiento:

**Figura 1**

*Ejemplo de formulario de postulación*

Tabla

Descripción generada automáticamente

Este formulario sirve como un apoyo visual para ilustrar la información que las academias de artes proporcionarían durante un proceso de postulación real (no se requiere completarlo como parte de esta actividad).

* Para mejorar el rendimiento de la base de datos, se considerará la desnormalización para incorporar la cantidad de profesores por academia.

Para la implementación, se requiere la creación de cuatro tablas principales en la primera etapa, considerando tanto tablas fuertes como débiles, la asignación adecuada de tipos de datos, restricciones (PK, FK, UN, NOT NULL) y el poblamiento inicial de datos para demostrar la funcionalidad del sistema”.

**Ahora que has leído el caso, deberás desarrollar lo requerido a través de los siguientes pasos:**

**Paso 1 – Diseño del Modelo conceptual**

Inicia interpretando el caso del Ministerio para diseñar un Modelo Entidad-Relación (MER). Este modelo debe representar las entidades, sus relaciones y atributos, basado en el análisis del escenario dado, donde el Ministerio busca impulsar la participación juvenil en tradiciones culturales.

**Paso 2 – Transformación del Modelo conceptual al Modelo lógico**

Transforma el MER en un Modelo Relacional Normalizado (MR), especificando cómo se organizarán los datos en la base de datos. Esto incluye la definición de claves primarias, claves foráneas y restricciones de integridad.

**Paso 3 – Desarrollo del Modelo Relacional**

Desarrolla el Modelo Relacional, considerando aspectos como la cantidad de profesores por academia.

**Paso 4 – Implementación de la base de datos**

* Utiliza Oracle SQL Developer y DBMS Base de Datos en Oracle CLOUD para llevar a cabo la creación de las tablas del Modelo Relacional (MR).

Podrás realizar la descarga de SQL a través del siguiente enlace:

<https://www.oracle.com/database/sqldeveloper/technologies/download/>

* Implementa las tablas utilizando sentencias SQL de creación (CREATE TABLE), define las relaciones entre tablas con sentencias de clave foránea (FOREIGN KEY), y llena las tablas con datos de prueba mediante sentencias de inserción (INSERT INTO).

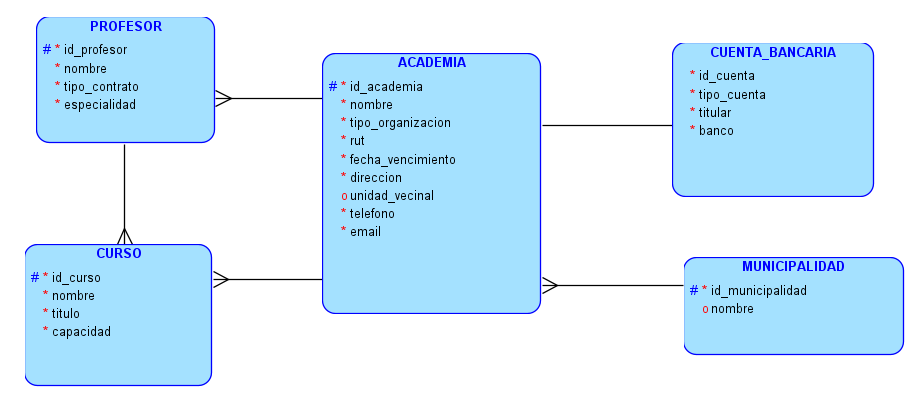
**Paso 5 – Script de implementación**

Adjunta el script de SQL utilizado para la creación y poblamiento de las tablas, asegurando que esté claro, comentado y bien organizado. Este script debe incluir:

* La construcción de cuatro tablas del modelo relacional normalizado, especificando dos tablas fuertes y dos débiles.
* Ejecución secuencial de comandos DDL en el orden adecuado.
* Asignación de tipos de datos adecuados a cada columna.
* Incorporación de restricciones de claves primarias (PK), claves foráneas (FK), únicas (UN), y not null, con nombres representativos.
* Poblamiento inicial de las tablas con al menos dos filas por tabla, utilizando objetos secuencia para asignar las llaves primarias.

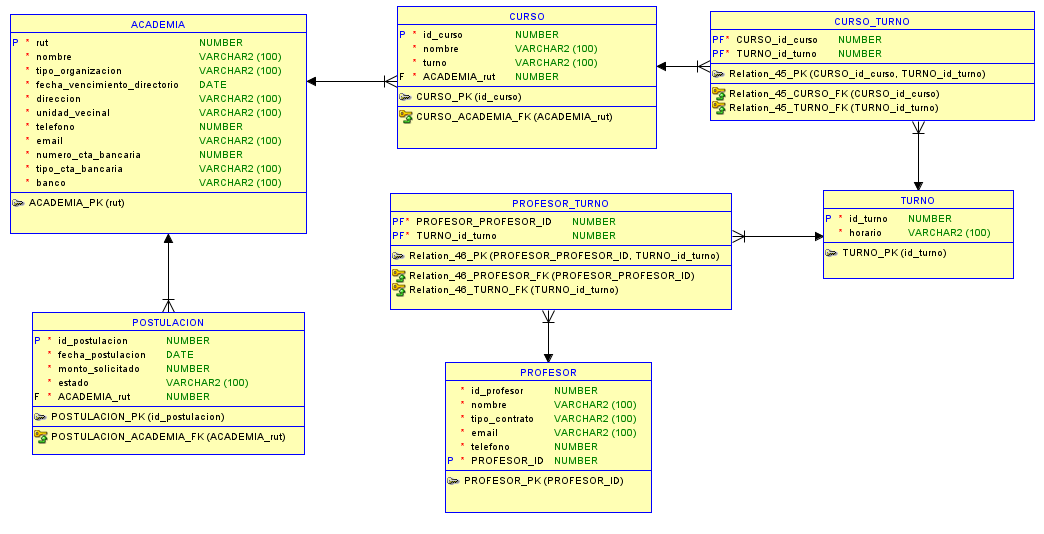
Cuando tengas lista tu solución, adjunta las siguientes evidencias:

* Modelo Entidad Relación (MER)



**(Este espacio es aproximado; ajústalo según tus necesidades)**

* Modelo Relacional Normalizado



* Script con las sentencias SQL para la creación y poblamiento de las tablas.

Script Creación

CREATE TABLE academia (

rut NUMBER NOT NULL,

nombre VARCHAR2(100) NOT NULL,

tipo\_organizacion VARCHAR2(100) NOT NULL,

fecha\_vencimiento\_directorio DATE NOT NULL,

direccion VARCHAR2(100) NOT NULL,

unidad\_vecinal VARCHAR2(100) NOT NULL,

telefono NUMBER NOT NULL,

email VARCHAR2(100) NOT NULL,

numero\_cta\_bancaria NUMBER NOT NULL,

tipo\_cta\_bancaria VARCHAR2(100) NOT NULL,

banco VARCHAR2(100) NOT NULL

);

ALTER TABLE academia ADD CONSTRAINT academia\_pk PRIMARY KEY ( rut );

CREATE TABLE curso (

id\_curso NUMBER NOT NULL,

nombre VARCHAR2(100) NOT NULL,

turno VARCHAR2(100) NOT NULL,

academia\_rut NUMBER NOT NULL

);

ALTER TABLE curso ADD CONSTRAINT curso\_pk PRIMARY KEY ( id\_curso );

CREATE TABLE curso\_turno (

curso\_id\_curso NUMBER NOT NULL,

turno\_id\_turno NUMBER NOT NULL

);

ALTER TABLE curso\_turno ADD CONSTRAINT relation\_45\_pk PRIMARY KEY ( curso\_id\_curso,

turno\_id\_turno );

CREATE TABLE postulacion (

id\_postulacion NUMBER NOT NULL,

fecha\_postulacion DATE NOT NULL,

monto\_solicitado NUMBER NOT NULL,

estado VARCHAR2(100) NOT NULL,

academia\_rut NUMBER NOT NULL

);

ALTER TABLE postulacion ADD CONSTRAINT postulacion\_pk PRIMARY KEY ( id\_postulacion );

CREATE TABLE profesor (

id\_profesor NUMBER NOT NULL,

nombre VARCHAR2(100) NOT NULL,

tipo\_contrato VARCHAR2(100) NOT NULL,

email VARCHAR2(100) NOT NULL,

telefono NUMBER NOT NULL,

profesor\_id NUMBER NOT NULL

);

ALTER TABLE profesor ADD CONSTRAINT profesor\_pk PRIMARY KEY ( profesor\_id );

CREATE TABLE profesor\_turno (

profesor\_profesor\_id NUMBER NOT NULL,

turno\_id\_turno NUMBER NOT NULL

);

ALTER TABLE profesor\_turno ADD CONSTRAINT relation\_46\_pk PRIMARY KEY ( profesor\_profesor\_id,

turno\_id\_turno );

CREATE TABLE turno (

id\_turno NUMBER NOT NULL,

horario VARCHAR2(100) NOT NULL

);

ALTER TABLE turno ADD CONSTRAINT turno\_pk PRIMARY KEY ( id\_turno );

ALTER TABLE curso

ADD CONSTRAINT curso\_academia\_fk FOREIGN KEY ( academia\_rut )

REFERENCES academia ( rut );

ALTER TABLE postulacion

ADD CONSTRAINT postulacion\_academia\_fk FOREIGN KEY ( academia\_rut )

REFERENCES academia ( rut );

ALTER TABLE curso\_turno

ADD CONSTRAINT relation\_45\_curso\_fk FOREIGN KEY ( curso\_id\_curso )

REFERENCES curso ( id\_curso );

ALTER TABLE curso\_turno

ADD CONSTRAINT relation\_45\_turno\_fk FOREIGN KEY ( turno\_id\_turno )

REFERENCES turno ( id\_turno );

ALTER TABLE profesor\_turno

ADD CONSTRAINT relation\_46\_profesor\_fk FOREIGN KEY ( profesor\_profesor\_id )

REFERENCES profesor ( profesor\_id );

ALTER TABLE profesor\_turno

ADD CONSTRAINT relation\_46\_turno\_fk FOREIGN KEY ( turno\_id\_turno )

REFERENCES turno ( id\_turno );

CREATE SEQUENCE curso\_id\_curso\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER curso\_id\_curso\_trg BEFORE

INSERT ON curso

FOR EACH ROW

WHEN ( new.id\_curso IS NULL )

BEGIN

:new.id\_curso := curso\_id\_curso\_seq.nextval;

END;

/

CREATE SEQUENCE postulacion\_id\_postulacion\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER postulacion\_id\_postulacion\_trg BEFORE

INSERT ON postulacion

FOR EACH ROW

WHEN ( new.id\_postulacion IS NULL )

BEGIN

:new.id\_postulacion := postulacion\_id\_postulacion\_seq.nextval;

END;

/

CREATE SEQUENCE profesor\_profesor\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER profesor\_profesor\_id\_trg BEFORE

INSERT ON profesor

FOR EACH ROW

WHEN ( new.profesor\_id IS NULL )

BEGIN

:new.profesor\_id := profesor\_profesor\_id\_seq.nextval;

END;

/

**(Este espacio es aproximado; ajústalo según tus necesidades)**

Población de Tablas

**-- Poblar ACADEMIA**

**INSERT INTO academia (rut, nombre, tipo\_organizacion, fecha\_vencimiento\_directorio, direccion, unidad\_vecinal, telefono, email, numero\_cta\_bancaria, tipo\_cta\_bancaria, banco) VALUES (12345678, 'Academia de Ciencias', 'Privada', TO\_DATE('2025-12-31', 'YYYY-MM-DD'), 'Av. Libertador 123', 'Unidad 1', 987654321, 'contacto@ciencias.cl', 1234567890, 'Corriente', 'Banco Estado');**

**INSERT INTO academia (rut, nombre, tipo\_organizacion, fecha\_vencimiento\_directorio, direccion, unidad\_vecinal, telefono, email, numero\_cta\_bancaria, tipo\_cta\_bancaria, banco) VALUES (87654321, 'Academia de Artes', 'Pública', TO\_DATE('2024-11-30', 'YYYY-MM-DD'), 'Calle Cultura 456', 'Unidad 2', 123456789, 'info@artes.cl', 9876543210, 'Ahorro', 'Banco Chile');**

**-- Poblando CURSO**

**INSERT INTO curso (id\_curso, nombre, turno, academia\_rut) VALUES (1, 'Matemáticas Avanzadas', 'Mañana', 12345678);**

**INSERT INTO curso (id\_curso, nombre, turno, academia\_rut) VALUES (2, 'Historia del Arte', 'Tarde', 87654321);**

**-- Poblar TURNO**

**INSERT INTO turno (id\_turno, horario) VALUES (1, '08:00 - 10:00');**

**INSERT INTO turno (id\_turno, horario) VALUES (2, '10:00 - 12:00');**

**-- Poblar CURSO\_TURNO**

**INSERT INTO curso\_turno (curso\_id\_curso, turno\_id\_turno) VALUES (1, 1);**

**INSERT INTO curso\_turno (curso\_id\_curso, turno\_id\_turno) VALUES (1, 2);**

**-- Poblar POSTULACION**

**INSERT INTO postulacion (id\_postulacion, fecha\_postulacion, monto\_solicitado, estado, academia\_rut) VALUES (1, TO\_DATE('2024-01-15', 'YYYY-MM-DD'), 500000, 'Pendiente', 12345678);**

**INSERT INTO postulacion (id\_postulacion, fecha\_postulacion, monto\_solicitado, estado, academia\_rut) VALUES (2, TO\_DATE('2024-02-20', 'YYYY-MM-DD'), 300000, 'Aprobado', 87654321);**

**-- Poblar PROFESOR**

**INSERT INTO profesor (id\_profesor, nombre, tipo\_contrato, email, telefono, profesor\_id) VALUES (1, 'Juan Pérez', 'Tiempo Completo', 'juan.perez@academia.cl', 987654321, 1);**

**INSERT INTO profesor (id\_profesor, nombre, tipo\_contrato, email, telefono, profesor\_id) VALUES (2, 'María López', 'Medio Tiempo', 'maria.lopez@academia.cl', 123456789, 2);**

**-- Poblar PROFESOR\_TURNO**

**INSERT INTO profesor\_turno (profesor\_profesor\_id, turno\_id\_turno) VALUES (1, 1);**

**INSERT INTO profesor\_turno (profesor\_profesor\_id, turno\_id\_turno) VALUES (2, 2);**

Además, tendrás que descargar el resultado. Para ello, deberás hacer clic en la opción Guardar como... del menú Archivo, esto despliega el submenú que se ilustra en la siguiente figura:

**Figura 2**

*Cómo guardar un archivo en SQL*

   
Nota. Ejemplo de guardado de archivo SQL. Oracle. (s.f.). *Oracle Data Modeler* [Software]. Oracle. <https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/>

**Paso 6:** todos los archivos descargados desde SQL deberás subirlos al repositorio GitHub. Si no has creado tu cuenta aún, puedes hacerlo a través del siguiente enlace:

<https://github.com/>

Una vez subido el archivo a GitHub, deberás descargar el archivo comprimido .java desde tu repositorio, tal como se muestra en la imagen:

Figura 3

Archivo .raw en GitHub

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente   
Nota. Descarga de archivo desde repositorio GitHub. GitHub (s.f.). GitHub. <https://github.com/>

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

Figura 4

Enlace de proyecto GitHub

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente   
Nota. Ejemplo de dónde se extrae un enlace en GitHub. GitHub (s.f.). GitHub. <https://github.com/>

**Paso 7: Presentación**

Para este paso, tendrás que realizar una presentación en video de tu proyecto para el Teatro Moro, con una duración máxima de 5 minutos. En este video, esperamos que compartas el proceso completo de construcción de tu aplicación, incluyendo el diseño, las estrategias de implementación que utilizaste, las dificultades que enfrentaste y cómo las superaste, así como las características únicas de tu proyecto. Adjunta el video al momento de subir tu actividad.

Puedes descargar el instructivo de Kaltura en el siguiente enlace: <https://ava.duoc.cl/bbcswebdav/xid-3242677_1>

**Paso 8:** para finalizar, sube el archivo comprimido en formato .zip o .rar junto con el enlace generado en el espacio dispuesto en el AVA.

Formulario Postulación

3. El archivo comprimido debe contener:

• Modelo Entidad Relación (MER)

• Modelo Relacional Normalizado (MR)

• Modelo Relacional Desnormalizado

Script con las sentencias SQL para la creación de las tablas y poblamiento de las tablas

**Presentación**

Para este paso, tendrás que realizar una presentación en video de tu proyecto para el Teatro Moro, con una duración máxima de 5 minutos. En este video, esperamos que compartas el proceso completo de construcción de tu aplicación, incluyendo el diseño, las estrategias de implementación que utilizaste, las dificultades que enfrentaste y cómo las superaste, así como las características únicas de tu proyecto.

Adjunta el video al momento de subir tu actividad.

https://github.com/krlosBarria/MBBDD\_EFT

Puedes descargar el instructivo de Kaltura en el siguiente enlace:

<https://ava.duoc.cl/bbcswebdav/xid-3577107_1>



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.