**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**

**FACULTAD DE CIENCIAS FISICO MATEMATICAS**

**CIUDAD UNIVERSITARIA**

**2do Avance Proyecto**

**Base de datos multimedia**

**Alumno:** Ricardo Alberto Grimaldo Estévez

**Matricula:** 1868520

**Grupo:** 005

**Carrera:** Licenciatura en Multimedia y Animación Digital

**Semestre: Enero – Junio 2022**

# Índice

Página

[Índice 2](#_Toc130667946)

[Modelo Relacional Completo 3](#_Toc130667947)

[Script Creación Tablas 4](#_Toc130667948)

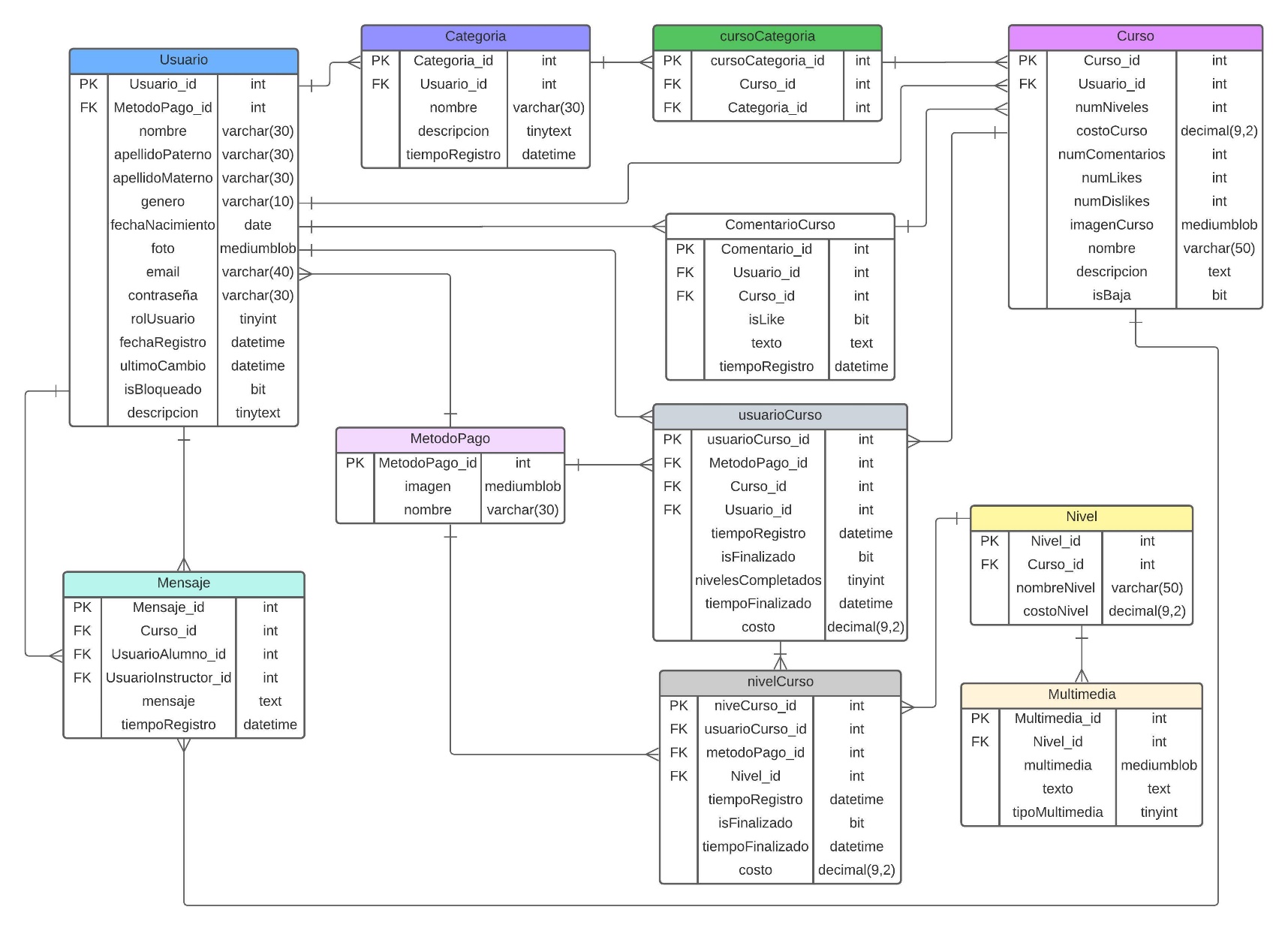
[Diagrama de MySQL Workbench 14](#_Toc130667949)

[Script Stored Procedure Gestión de Usuarios 16](#_Toc130667950)

[SCRIPT DATA DICTIONARY 20](#_Toc130667951)

[Clase conexión con la base de datos PHP 24](#_Toc130667952)

# Modelo Relacional Completo



# Script Creación Tablas

CREATE DATABASE BDMM\_PROYECTO;

USE BDMM\_PROYECTO;

-- TABLA DE METODOS DE PAGO --

DROP TABLE IF EXISTS MetodoPago;

CREATE TABLE MetodoPago(

MetodoPago\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de los Metodos de pago',

nombreMetodo VARCHAR(30) NOT NULL COMMENT'Nombre del metodo de pago',

imagenMetodo MEDIUMBLOB NOT NULL COMMENT'Imagen del metodo de pago',

CONSTRAINT PK\_MetodoPago

PRIMARY KEY (MetodoPago\_id)

);

-- TABLA DE USUARIOS --

DROP TABLE IF EXISTS Usuario;

CREATE TABLE Usuario(

Usuario\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria Tabla Usuario',

MetodoPago\_id INT COMMENT'Clave Foránea de los Metodos de pago',

correo VARCHAR(40) NOT NULL UNIQUE COMMENT'Correo electrónico del usuario',

userPassword VARCHAR(30) NOT NULL COMMENT'Contraseña del usuario',

rolUsuario TINYINT NOT NULL COMMENT'No. que identifica el rol del usuario 1:Admin, 2:Instructor, 3:Alumno',

fotoPerfil MEDIUMBLOB COMMENT'Foto de perfil tipo avatar',

descripcion TINYTEXT COMMENT'descripcion del usuario',

nombre VARCHAR(30) NOT NULL COMMENT'Nombre completo del usuario',

apellidoMaterno VARCHAR(30) NOT NULL COMMENT'Apellido materno del usuario',

apellidoPaterno VARCHAR(30) NOT NULL COMMENT'Apellido paterno del usuario',

fechaNacimiento DATE NOT NULL COMMENT'Fecha de nacimiento del usuario',

sexo VARCHAR(10) NOT NULL COMMENT'Género del usuario',

fechaRegistro DATETIME NOT NULL COMMENT'Fecha en la que se dio de alta el usuario',

ultimoCambio DATETIME NOT NULL COMMENT'Fecha ultima de modificacion del usuario',

esBloqueado BIT DEFAULT 0 COMMENT'Bandera que indica si esta bloqueado',

CONSTRAINT PK\_Usuario

PRIMARY KEY (Usuario\_id),

CONSTRAINT FK\_Usuario\_MetodoPago

FOREIGN KEY (MetodoPago\_id) REFERENCES MetodoPago(MetodoPago\_id)

);

-- TABLA DE CURSO --

DROP TABLE IF EXISTS Curso;

CREATE TABLE Curso(

Curso\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de los cursos',

Usuario\_id INT NOT NULL COMMENT'Clave Foránea del usuario que registro el curso',

noNiveles INT NOT NULL COMMENT'número de niveles del curso',

costoCurso DECIMAL(9,2) UNSIGNED NOT NULL COMMENT'Costo del curso',

noComentarios INT COMMENT'numero de comentarios del curso',

noLikes INT COMMENT'numero de reseñas buenas del curso',

noDislikes INT COMMENT'numero de reseñas malas del curso',

imagenCurso MEDIUMBLOB NOT NULL COMMENT'Imagen del curso',

nombre VARCHAR(50) NOT NULL COMMENT'Nombre del curso',

descripcion TEXT NOT NULL COMMENT'Nombre de usuario',

isBaja BIT DEFAULT 0 COMMENT'Bandera que indica si esta bloqueado',

CONSTRAINT PK\_Curso

PRIMARY KEY (Curso\_id),

CONSTRAINT FK\_Curso\_Usuario

FOREIGN KEY (Usuario\_id) REFERENCES Usuario(Usuario\_id)

);

-- TABLA DE CATEGORIA --

DROP TABLE IF EXISTS Categoria;

CREATE TABLE Categoria(

Categoria\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de las categorias',

Usuario\_id INT NOT NULL COMMENT'Clave Foránea del usuario que registro la categoria',

nombre VARCHAR(30) NOT NULL COMMENT'Nombre de la categoria',

descripcion TINYTEXT NOT NULL COMMENT'Descripcion de la categoria',

tiempoRegistro DATETIME NOT NULL COMMENT'Tiempo de registro de la categoria',

CONSTRAINT PK\_Categoria

PRIMARY KEY (Categoria\_id),

CONSTRAINT FK\_Categoria\_Usuario

FOREIGN KEY (Usuario\_id) REFERENCES Usuario(Usuario\_id)

);

-- TABLA DE COMENTARIO CURSO (LIKE Y DISLIKE)--

DROP TABLE IF EXISTS ComentarioCurso;

CREATE TABLE ComentarioCurso(

ComentarioCurso\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de los comentarios del curso(reseñas)',

Usuario\_id INT NOT NULL COMMENT'Clave Foránea del usuario que registro el comentario del curso',

Curso\_id INT NOT NULL COMMENT'Clave Foránea del curso que registro el comentario del curso',

isLike BIT NOT NULL COMMENT'Bandera que indica si es una reseña positiva o negativa',

texto TEXT NOT NULL COMMENT'Texto sobre la reseña del curso',

tiempoRegistro DATETIME NOT NULL COMMENT'Tiempo de registro de la reseña',

CONSTRAINT PK\_ComentarioCurso

PRIMARY KEY (ComentarioCurso\_id),

CONSTRAINT FK\_ComentarioCurso\_Usuario

FOREIGN KEY (Usuario\_id) REFERENCES Usuario(Usuario\_id),

CONSTRAINT FK\_ComentarioCurso\_Curso

FOREIGN KEY (Curso\_id) REFERENCES Curso(Curso\_id)

);

-- TABLA DE MENSAJES--

DROP TABLE IF EXISTS Mensaje;

CREATE TABLE Mensaje(

Mensaje\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de los mensajes',

UsuarioInstructor\_id INT NOT NULL COMMENT'Clave Foránea del usuario instructor del mensaje',

UsuarioAlumno\_id INT NOT NULL COMMENT'Clave Foránea del usuario alumno del mensaje',

Curso\_id INT NOT NULL COMMENT'Clave Foránea del curso del mensaje',

texto TEXT NOT NULL COMMENT'Texto sobre la reseña del curso',

tiempoRegistro DATETIME NOT NULL COMMENT'Tiempo de registro del mensaje',

CONSTRAINT PK\_Mensaje

PRIMARY KEY (Mensaje\_id),

CONSTRAINT FK\_Mensaje\_UsuarioInstructor

FOREIGN KEY (UsuarioInstructor\_id) REFERENCES Usuario(Usuario\_id),

CONSTRAINT FK\_Mensaje\_UsuarioAlumno

FOREIGN KEY (UsuarioAlumno\_id) REFERENCES Usuario(Usuario\_id),

CONSTRAINT FK\_Mensaje\_Curso

FOREIGN KEY (Curso\_id) REFERENCES Curso(Curso\_id)

);

-- TABLA DE CURSO CATEGORIA--

DROP TABLE IF EXISTS CursoCategoria;

CREATE TABLE CursoCategoria(

CursoCategoria\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de las categorias de los cursos',

Curso\_id INT NOT NULL COMMENT'Clave Foránea del curso',

Categoria\_id INT NOT NULL COMMENT'Clave Foránea de la categoria',

CONSTRAINT PK\_CursoCategoria

PRIMARY KEY (CursoCategoria\_id),

CONSTRAINT FK\_CursoCategoria\_Categoria

FOREIGN KEY (Categoria\_id) REFERENCES Categoria(Categoria\_id),

CONSTRAINT FK\_CursoCategoria\_Curso

FOREIGN KEY (Curso\_id) REFERENCES Curso(Curso\_id)

);

-- TABLA DE NIVEL--

DROP TABLE IF EXISTS Nivel;

CREATE TABLE Nivel(

Nivel\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de los niveles',

Curso\_id INT NOT NULL COMMENT'Clave Foránea del curso',

nombre VARCHAR(50) NOT NULL COMMENT'Nombre del nivel',

costoNivel DECIMAL(9,2) UNSIGNED NOT NULL COMMENT'Costo del nivel',

CONSTRAINT PK\_Nivel

PRIMARY KEY (Nivel\_id),

CONSTRAINT FK\_Nivel\_Curso

FOREIGN KEY (Curso\_id) REFERENCES Curso(Curso\_id)

);

-- TABLA DE MULTIMEDIA--

DROP TABLE IF EXISTS Multimedia;

CREATE TABLE Multimedia(

Multimedia\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de la multimedia',

Nivel\_id INT NOT NULL COMMENT'Clave Foránea del nivel',

multimedia MEDIUMBLOB COMMENT'Multimedia',

texto TEXT COMMENT'Texto',

tipoMultimedia TINYINT NOT NULL COMMENT'No. que identifica el tipo de multimedia del nivel 1:texto o link, 2:imagen, 3:video, 4:PDF',

CONSTRAINT PK\_Multimedia

PRIMARY KEY (Multimedia\_id),

CONSTRAINT FK\_Multimedia\_Nivel

FOREIGN KEY (Nivel\_id) REFERENCES Nivel(Nivel\_id)

);

-- TABLA DE usuarioCurso--

DROP TABLE IF EXISTS usuarioCurso;

CREATE TABLE usuarioCurso(

usuarioCurso\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de usuarioCurso',

MetodoPago\_id INT NOT NULL COMMENT'Clave Foránea del metodo de pago del usuario',

Curso\_id INT NOT NULL COMMENT'Clave Foránea del curso',

Usuario\_id INT NOT NULL COMMENT'Clave Foránea del usuario',

tiempoRegistro DATETIME NOT NULL COMMENT'Tiempo de registro del usuario al curso',

isFinalizado BIT NOT NULL COMMENT'Bandera que indica si el usuario termino el curso',

nivelesCompletados TINYINT DEFAULT 0 COMMENT'Cantidad de niveles completados del curso por el estudiante',

tiempoCompletado DATETIME NOT NULL COMMENT'Tiempo de finalización del usuario al curso',

costoCurso DECIMAL(9,2) UNSIGNED NOT NULL COMMENT'Costo del curso en ese momento',

CONSTRAINT PK\_usuarioCurso

PRIMARY KEY (usuarioCurso\_id),

CONSTRAINT FK\_usuarioCurso\_MetodoPago

FOREIGN KEY (MetodoPago\_id) REFERENCES MetodoPago(MetodoPago\_id),

CONSTRAINT FK\_usuarioCurso\_Curso

FOREIGN KEY (Curso\_id) REFERENCES Curso(Curso\_id),

CONSTRAINT FK\_usuarioCurso\_Usuario

FOREIGN KEY (Usuario\_id) REFERENCES Usuario(Usuario\_id)

);

-- TABLA DE usuarioNivel--

DROP TABLE IF EXISTS nivelCurso;

CREATE TABLE nivelCurso(

nivelCurso\_id INT AUTO\_INCREMENT NOT NULL COMMENT'Clave Primaria de usuarioNivel',

MetodoPago\_id INT NOT NULL COMMENT'Clave Foránea del metodo de pago del usuario',

usuarioCurso\_id INT NOT NULL COMMENT'Clave Foránea del usuarioCurso',

Nivel\_id INT NOT NULL COMMENT'Clave Foránea del nivel',

tiempoRegistro DATETIME NOT NULL COMMENT'Tiempo de registro del usuario al curso',

isFinalizado BIT NOT NULL COMMENT'Bandera que indica si el usuario termino el nivel',

tiempoCompletado DATETIME NOT NULL COMMENT'Tiempo de finalización del usuario al nivel',

costoNivel DECIMAL(9,2) UNSIGNED NOT NULL COMMENT'Costo del nivel en ese momento',

CONSTRAINT PK\_nivelCurso

PRIMARY KEY (nivelCurso\_id),

CONSTRAINT FK\_PK\_nivelCurso\_MetodoPago

FOREIGN KEY (MetodoPago\_id) REFERENCES MetodoPago(MetodoPago\_id),

CONSTRAINT FK\_PK\_nivelCurso\_Nivel

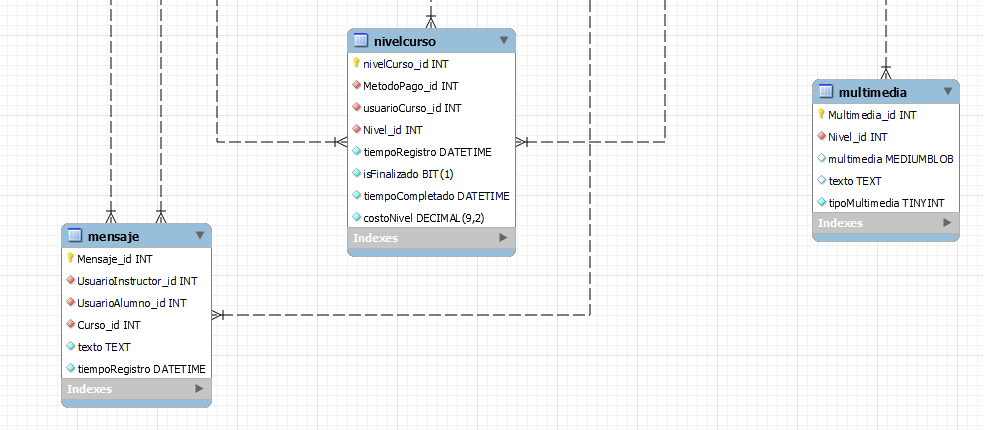
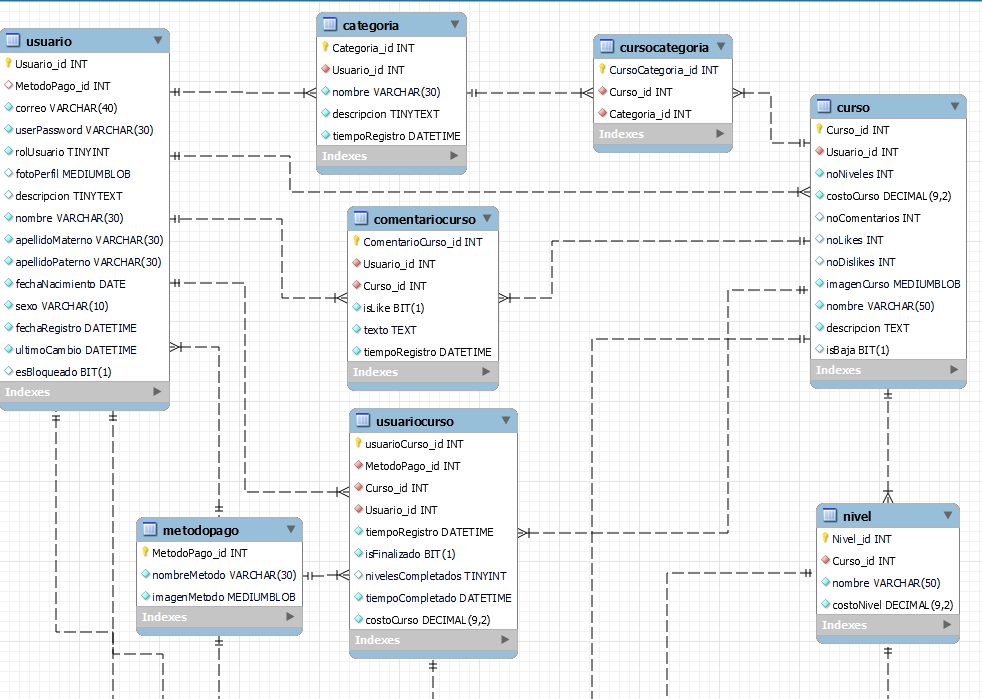
FOREIGN KEY (Nivel\_id) REFERENCES Nivel(Nivel\_id),

CONSTRAINT FK\_nivelCurso\_usuarioCurso

FOREIGN KEY (usuarioCurso\_id) REFERENCES usuarioCurso(usuarioCurso\_id)

);

# Diagrama de MySQL Workbench



# Script Stored Procedure Gestión de Usuarios

USE BDMM\_PROYECTO;

/\*--------------------------------------------------------------------------------USUARIOS--------------------------------------------------------------------------\*/

DROP PROCEDURE IF EXISTS sp\_GestionUsuario;

DELIMITER //

CREATE PROCEDURE sp\_GestionUsuario

(

Operacion CHAR(1),

sp\_Usuario\_id INT,

sp\_MetodoPago\_id INT,

sp\_correo VARCHAR(40),

sp\_userPassword VARCHAR(30),

sp\_rolUsuario TINYINT,

sp\_fotoPerfil MEDIUMBLOB,

sp\_descripcion TINYTEXT,

sp\_nombre VARCHAR(30),

sp\_apellidoPaterno VARCHAR(30),

sp\_apellidoMaterno VARCHAR(30),

sp\_fechaNacimiento VARCHAR(30),

sp\_sexo VARCHAR(10),

sp\_esBloqueado BIT

)

BEGIN

DECLARE sp\_ultimoCambio DATETIME;

IF Operacion = 'I' /\*INSERT USUARIO\*/

THEN

INSERT INTO Usuario(correo ,userPassword ,rolUsuario ,fotoPerfil ,nombre ,apellidoMaterno ,apellidoPaterno ,fechaNacimiento ,sexo,fechaRegistro, ultimoCambio)

VALUES (sp\_correo,sp\_userPassword,sp\_rolUsuario,sp\_fotoPerfil,sp\_nombre ,sp\_apellidoMaterno,sp\_apellidoPaterno,sp\_fechaNacimiento,sp\_sexo,now(), now());

END IF;

IF Operacion = 'E' /\*EDIT USUARIO\*/

THEN

SET sp\_ultimoCambio = now();

SET sp\_correo=IF(sp\_correo='',NULL,sp\_correo),

sp\_userPassword=IF(sp\_userPassword='',NULL,sp\_userPassword),

sp\_rolUsuario=IF(sp\_rolUsuario='',NULL,sp\_rolUsuario),

sp\_fotoPerfil=IF(sp\_fotoPerfil='',NULL,sp\_fotoPerfil),

sp\_nombre=IF(sp\_nombre='',NULL,sp\_nombre),

sp\_descripcion=IF(sp\_descripcion='',NULL,sp\_descripcion),

sp\_apellidoMaterno=IF(sp\_apellidoMaterno='',NULL,sp\_apellidoMaterno),

sp\_apellidoPaterno=IF(sp\_apellidoPaterno='',NULL,sp\_apellidoPaterno),

sp\_fechaNacimiento=IF(sp\_fechaNacimiento='',NULL,sp\_fechaNacimiento),

sp\_ultimoCambio=IF(sp\_ultimoCambio='',NULL,sp\_ultimoCambio);

UPDATE Usuario

SET correo = IFNULL(sp\_correo,correo),

userPassword= IFNULL(sp\_userPassword,userPassword),

descripcion= IFNULL(sp\_descripcion,descripcion),

rolUsuario= IFNULL(sp\_rolUsuario,rolUsuario),

fotoPerfil= IFNULL(sp\_fotoPerfil,fotoPerfil),

nombre= IFNULL(sp\_nombreUsuario,nombre),

apellidoMaterno= IFNULL(sp\_apellidoMaterno,apellidoMaterno),

apellidoPaterno= IFNULL(sp\_apellidoPaterno,apellidoPaterno),

fechaNacimiento= IFNULL(sp\_fechaNacimiento,fechaNacimiento),

ultimoCambio= IFNULL(sp\_ultimoCambio,ultimoCambio)

WHERE Usuario\_id=sp\_Usuario\_id;

END IF;

IF Operacion = 'D' THEN /\*DELETE USUARIO\*/

DELETE FROM Usuario WHERE Usuario\_id = sp\_Usuario\_id;

END IF;

IF Operacion = 'P' THEN /\*BLOQUEAR/DESBLOQUEAR USUARIO\*/

UPDATE Usuario SET esBloqueado = sp\_esBloqueado WHERE Usuario\_id = sp\_Usuario\_id;

END IF;

IF Operacion = 'L' THEN /\*LOG IN USUARIO\*/

SELECT Usuario\_id, rolUsuario

FROM Usuario

WHERE 1=1

AND correo = sp\_correo

AND userPassword = sp\_userPassword;

END IF;

IF Operacion = 'A' THEN /\*GET DATOS ALL USUARIOS\*/

SELECT Usuario\_id, MetodoPago\_id, correo, userPassword, rolUsuario, fotoPerfil, descripcion, nombre, apellidoMaterno, apellidoPaterno, fechaNacimiento, sexo, fechaRegistro, ultimoCambio, esBloqueado

FROM Usuario;

END IF;

IF Operacion = 'G' THEN /\*GET DATOS USUARIO\*/

SELECT Usuario\_id, MetodoPago\_id, correo, userPassword, rolUsuario, fotoPerfil, descripcion, nombre, apellidoMaterno, apellidoPaterno, fechaNacimiento, sexo, fechaRegistro, ultimoCambio, esBloqueado

FROM Usuario

WHERE Usuario\_id = sp\_Usuario\_id;

END IF;

END //

# SCRIPT DATA DICTIONARY

USE bdmm\_proyecto;

SELECT t.table\_schema AS db\_name,

t.table\_name,

(CASE WHEN t.table\_type = 'BASE TABLE' THEN 'table'

WHEN t.table\_type = 'VIEW' THEN 'view'

ELSE t.table\_type

END) AS table\_type,

c.column\_name,

c.column\_type,

c.column\_default,

c.column\_key,

c.is\_nullable,

c.extra,

c.column\_comment

FROM information\_schema.tables AS t

INNER JOIN information\_schema.columns AS c

ON t.table\_name = c.table\_name

AND t.table\_schema = c.table\_schema

WHERE t.table\_schema = 'BDMM\_PROYECTO'

ORDER BY t.table\_schema,

t.table\_name,

c.ordinal\_position

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TABLE\_NAME | COLUMN\_NAME | COLUMN\_  TYPE | COLU  MN  \_  KEY | IS\_  NULL  ABLE | EXTRA | COLUMN\_COMMENT |
| categoria | Categoria\_id | int | PRI | NO | auto\_increment | Clave Primaria de las categorías |
| categoria | Usuario\_id | int | MUL | NO |  | Clave foránea del usuario que registro la  categoría |
| categoria | nombre | varchar(30) |  | NO |  | Nombre de la categoría |
| categoria | descripcion | tinytext |  | NO |  | descripción de la categoría |
| categoria | tiempoRegistro | datetime |  | NO |  | Tiempo de registro de la categoría |
| Comentario  curso | Comentario  Curso\_id | int | PRI | NO | auto\_increment | Clave Primaria de los comentarios del curso  (reseñas) |
| Comentario  curso | Usuario\_id | int | MUL | NO |  | Clave Foránea del usuario que registro el  comentario del curso |
| Comentario  curso | Curso\_id | int | MUL | NO |  | Clave Foránea del curso que registro el  comentario del curso |
| Comentario  curso | isLike | bit(1) |  | NO |  | Bandera que indica si es una reseña positiva  o negativa |
| Comentario  curso | texto | text |  | NO |  | Texto sobre la reseña del curso |
| Comentario  curso | tiempoRegistro | datetime |  | NO |  | Tiempo de registro de la reseña |
| curso | Curso\_id | int | PRI | NO | auto\_increment | Clave Primaria de los cursos |
| curso | Usuario\_id | int | MUL | NO |  | Clave foránea del usuario que registro el curso |
| curso | noNiveles | int |  | NO |  | numero de niveles del curso |
| curso | costoCurso | decimal(9,2) unsigned |  | NO |  | Costo del curso |
| curso | noComentarios | int |  | YES |  | numero de comentarios del curso |
| curso | noLikes | int |  | YES |  | numero de reseñas buenas del curso |
| curso | noDislikes | int |  | YES |  | numero de reseñas malas del curso |
| curso | imagenCurso | mediumblob |  | NO |  | Imagen del curso |
| curso | nombre | varchar(50) |  | NO |  | Nombre del curso |
| curso | descripcion | text |  | NO |  | Nombre de usuario |
| curso | isBaja | bit(1) |  | YES |  | Bandera que indica si está bloqueado |
| cursocategoria | Curso  Categoria\_id | int | PRI | NO | auto\_increment | Clave Primaria de las categorías de los cursos |
| cursocategoria | Curso\_id | int | MUL | NO |  | Clave foránea del curso |
| cursocategoria | Categoria\_id | int | MUL | NO |  | Clave foránea de la categoría |
| mensaje | Mensaje\_id | int | PRI | NO | auto\_increment | Clave Primaria de los mensajes |
| mensaje | Usuario  Instructor\_id | int | MUL | NO |  | Clave foránea del usuario instructor del mensaje |
| mensaje | Usuario  Alumno\_id | int | MUL | NO |  | Clave foránea del usuario alumno del mensaje |
| mensaje | Curso\_id | int | MUL | NO |  | Clave foránea del curso del mensaje |
| mensaje | texto | text |  | NO |  | Texto sobre la reseña del curso |
| mensaje | tiempoRegistro | datetime |  | NO |  | Tiempo de registro del mensaje |
| metodopago | MetodoPago\_id | int | PRI | NO | auto\_increment | Clave Primaria de los métodos de pago |
| metodopago | nombreMetodo | varchar(30) |  | NO |  | Nombre del método de pago |
| metodopago | imagenMetodo | mediumblob |  | NO |  | Imagen del método de pago |
| multimedia | Multimedia\_id | int | PRI | NO | auto\_increment | Clave Primaria de la multimedia |
| multimedia | Nivel\_id | int | MUL | NO |  | Clave foránea del nivel |
| multimedia | multimedia | mediumblob |  | YES |  | Multimedia |  |
| multimedia | texto | text |  | YES |  | Texto |  |
| multimedia | tipoMultimedia | tinyint |  | NO |  | No. que identifica el tipo de multimedia del nivel  1: texto o link, 2:imagen, 3:video, 4:PDF |
| nivel | Nivel\_id | int | PRI | NO | auto\_increment | Clave Primaria de los niveles |
| nivel | Curso\_id | int | MUL | NO |  | Clave foránea del curso |
| nivel | nombre | varchar(50) |  | NO |  | Nombre del nivel |
| nivel | costoNivel | decimal(9,2) unsigned |  | NO |  | Costo del nivel |
| nivelcurso | nivelCurso\_id | int | PRI | NO | auto\_increment | Clave Primaria de usuarioNivel |
| nivelcurso | Método  Pago\_id | int | MUL | NO |  | Clave foránea del método de pago del usuario |
| nivelcurso | Usuario  Curso\_id | int | MUL | NO |  | Clave foránea del usuarioCurso |
| nivelcurso | Nivel\_id | int | MUL | NO |  | Clave foránea del nivel |
| nivelcurso | tiempoRegistro | datetime |  | NO |  | Tiempo de registro del usuario al curso |
| nivelcurso | isFinalizado | bit(1) |  | NO |  | Bandera que indica si el usuario termino el nivel |
| nivelcurso | Tiempo  Completado | datetime |  | NO |  | Tiempo de finalización del usuario al nivel |
| nivelcurso | costoNivel | decimal(9,2) unsigned |  | NO |  | Costo del nivel en ese momento |
| usuario | Usuario\_id | int | PRI | NO | auto\_increment | Clave Primaria Tabla Usuario |
| usuario | MetodoPago\_id | int | MUL | YES |  | Clave foránea de los métodos de pago |
| usuario | correo | varchar(40) | UNI | NO |  | Correo electrÃ³nico del usuario |
| usuario | userPassword | varchar(30) |  | NO |  | ContraseÃ±a del usuario |
| usuario | rolUsuario | tinyint |  | NO |  | No. que identifica el rol del usuario  1: Admin, 2:Instructor, 3:Alumno |
| usuario | fotoPerfil | mediumblob |  | YES |  | Foto de perfil tipo avatar |
| usuario | descripcion | tinytext |  | YES |  | descripcion del usuario |
| usuario | nombre | varchar(30) |  | NO |  | Nombre completo del usuario |
| usuario | apellidoMaterno | varchar(30) |  | NO |  | Apellido materno del usuario |
| usuario | apellidoPaterno | varchar(30) |  | NO |  | Apellido paterno del usuario |
| usuario | fechaNacimiento | date |  | NO |  | Fecha de nacimiento del usuario |
| usuario | sexo | varchar(10) |  | NO |  | GÃ©nero del usuario |
| usuario | fechaRegistro | datetime |  | NO |  | Fecha en la que se dio de alta el usuario |
| usuario | ultimoCambio | datetime |  | NO |  | Fecha ultima de modificacion del usuario |
| usuario | esBloqueado | bit(1) |  | YES |  | Bandera que indica si esta bloqueado |
| usuariocurso | usuarioCurso\_id | int | PRI | NO | auto\_increment | Clave Primaria de usuarioCurso |
| usuariocurso | MetodoPago\_id | int | MUL | NO |  | Clave foránea del método de pago del usuario |
| usuariocurso | Curso\_id | int | MUL | NO |  | Clave foránea del curso |
| usuariocurso | Usuario\_id | int | MUL | NO |  | Clave foránea del usuario |
| usuariocurso | tiempoRegistro | datetime |  | NO |  | Tiempo de registro del usuario al curso |
| usuariocurso | isFinalizado | bit(1) |  | NO |  | Bandera que indica si el usuario termino el curso |
| usuariocurso | nivelesCompletados | tinyint |  | YES |  | Cantidad de niveles completados del curso  por el estudiante |
| usuariocurso | tiempoCompletado | datetime |  | NO |  | Tiempo de finalización del usuario al curso |
| usuariocurso | costoCurso | decimal(9,2) unsigned |  | NO |  | Costo del curso en ese momento |

# Clase conexión con la base de datos PHP

<?php

class DB

{

    private $host;

    private $db;

    private $user;

    private $password;

    private $charset;

    public function \_\_construct()

    {

        $this->host = 'localhost';

        $this->db = 'BDMM\_PROYECTO';

        $this->user = 'root';

        $this->password = 'db\_password';

        $this->charset = 'utf8mb4';

    }

    function connect()

    {

        try {

            $connection = "mysql:host=" . $this->host . ";dbname=" . $this->db . ";charset=" . $this->charset;

            $options = [

                PDO::ATTR\_ERRMODE            => PDO::ERRMODE\_EXCEPTION,

                PDO::ATTR\_EMULATE\_PREPARES   => false,

            ];

            $pdo = new PDO($connection, $this->user, $this->password);

            return $pdo;

        } catch (PDOException $e) {

            print\_r('Error Connection: ' . $e->getMessage());

        }

    }

}