# Tabla de contenido

[Tabla de contenido 1](#_Toc199918340)

[PLAN DE MIGRACIÓN MYSQL XAMPP A AMAZON RDS 2](#_Toc199918341)

[BASE DE DATOS ACTUAL 2](#_Toc199918342)

[Estructura de Tablas 2](#_Toc199918343)

[Relaciones Identificadas 3](#_Toc199918344)

[Puntos Críticos Detectados 3](#_Toc199918345)

[CONFIGURACIÓN ESPECÍFICA PARA RDS 3](#_Toc199918346)

[Configuración Recomendada 3](#_Toc199918347)

[MIGRACIÓN PASO A PASO 3](#_Toc199918348)

[VALIDACIÓN ESPECÍFICA 6](#_Toc199918349)

[ACTUALIZACIÓN DE LA APLICACIÓN NODE.JS 8](#_Toc199918350)

[# Base de datos RDS 10](#_Toc199918351)

[MONITOREO ESPECÍFICO 14](#_Toc199918352)

# PLAN DE MIGRACIÓN MYSQL XAMPP A AMAZON RDS

## BASE DE DATOS ACTUAL

**Información Específica**

**Base de datos:**

**Motor actual: MariaDB 10.4.32 (compatible con MySQL)**

**Charset: utf8mb4\_general\_ci**

**Backend: Node.js**

**Tablas: 7 tablas principales**

**Registros aproximados: ~50 registros totales (BD pequeña)**

**Tamaño estimado: < 5MB**

### Estructura de Tablas

**├── usuarios (17 registros) -- Tabla principal**

**├── campañas (9 registros) -- Incluye imágenes**

**├── asistencia (7 registros) -- Registro de horas**

**├── postulacion (32 registros) -- Estados: pendiente/aceptada/rechazada**

**├── notificaciones (14 registros)**

**├── rol (3 registros) -- administrador/usuario/Docente**

**└── certificacion (vacía) -- Para futuras certificaciones**

#### Relaciones Identificadas

**usuarios → rol (FK: id\_rol)**

**postulacion → usuarios + campañas (FK dobles)**

**asistencia → usuarios + campañas (FK dobles)**

**certificacion → asistencia (FK: id\_asistencia)**

##### Puntos Críticos Detectados

**Inconsistencias que corregir:**

* **Tabla con datos inconsistentes**
* **Tabla: fechas con valor '0000-00-00'**
* **Passwords hasheados con diferentes algoritmos ($2y$ y $2a$)**
* **Campo en campañas requiere manejo de archivos**

# CONFIGURACIÓN ESPECÍFICA PARA RDS

## Configuración Recomendada

**sql**

***-- Configuraciones específicas para tu aplicación Node.js***

**character\_set\_server = utf8mb4**

**collation\_server = utf8mb4\_unicode\_ci**

**sql\_mode = "TRADITIONAL"**

**time\_zone = "-05:00" *-- Zona horaria Colombia***

**max\_connections = 100**

##### MIGRACIÓN PASO A PASO

**Paso 1: Preparación del Backup**

**bash**

***## Backup específico de ldm\_academy***

**mysqldump --u root --p --single-transaction --routines --triggers \**

**--default-character-set=utf8mb4 ldm\_academy > ldm\_academy\_backup.sql**

***## Backup solo de estructura (para pruebas)***

**mysqldump --u root --p --no-data --routines --triggers ldm\_academy > ldm\_academy\_estructura.sql**

**Paso 2: Limpieza de Datos**

**sql**

***-- Script para limpiar inconsistencias antes de migrar***

**USE ldm\_academy;**

***-- Arreglar fechas en notificaciones***

**UPDATE notificaciones**

**SET fecha = CURRENT\_TIMESTAMP**

**WHERE fecha = '0000-00-00 00:00:00';**

***-- Verificar consistencia en asistencia***

**SELECT id\_asistencia, fecha, hora\_Inicio, hora\_fin**

**FROM asistencia**

**WHERE hora\_fin < hora\_Inicio;**

***-- Opcional: Normalizar passwords a un solo formato***

**Paso 3: Crear RDS Instance**

**bash**

***## Comando AWS CLI***

**aws rds create-db-instance \**

**--db-instance-identifier ldm-academy-db \**

**--db-instance-class db.t3.micro \**

**--engine mysql \**

**--engine-version 8.0.39 \**

**--master-username admin \**

**--master-user-password TuPasswordSeguro123! \**

**--allocated-storage 20 \**

**--storage-type gp2 \**

**--vpc-security-group-ids sg-xxxxxxxxxxxxxxx \**

**--db-subnet-group-name default \**

**--backup-retention-period 7 \**

**--port 3306**

**Paso 4: Migración de Estructura**

**sql**

***-- Conectar a RDS***

**mysql *--h ldm-academy-db.xxxxxx.us-east-1.rds.amazonaws.com --u admin --p***

***-- Crear la base de datos***

**CREATE DATABASE ldm\_academy**

**CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;**

**USE ldm\_academy;**

**bash**

***## Importar estructura***

**mysql --h tu-endpoint-rds --u admin --p ldm\_academy < ldm\_academy\_estructura.sql**

**Paso 5: Migración de Datos**

**bash**

***## Importar datos completos***

**mysql --h tu-endpoint-rds --u admin --p ldm\_academy < ldm\_academy\_backup.sql**

**Paso 6: Crear Usuario Específico para la Aplicación**

**sql**

***-- Crear usuario específico para tu aplicación Node.js***

**CREATE USER 'ldm\_app'@'%' IDENTIFIED BY 'PasswordSeguroParaApp123!';**

***-- Dar permisos específicos***

**GRANT SELECT, INSERT, UPDATE, DELETE ON ldm\_academy.\* TO 'ldm\_app'@'%';**

***-- Solo si necesitas crear/modificar tablas desde la app***

**GRANT CREATE, ALTER, INDEX ON ldm\_academy.\* TO 'ldm\_app'@'%';**

**FLUSH PRIVILEGES;**

##### VALIDACIÓN ESPECÍFICA

**Verificar Conteos de Registros**

**sql**

***-- Script de validación específico para ldm\_academy***

**SELECT 'usuarios' as tabla, COUNT(\*) as registros FROM usuarios**

**UNION ALL**

**SELECT 'campañas' as tabla, COUNT(\*) as registros FROM campañas**

**UNION ALL**

**SELECT 'asistencia' as tabla, COUNT(\*) as registros FROM asistencia**

**UNION ALL**

**SELECT 'postulacion' as tabla, COUNT(\*) as registros FROM postulacion**

**UNION ALL**

**SELECT 'notificaciones' as tabla, COUNT(\*) as registros FROM notificaciones**

**UNION ALL**

**SELECT 'rol' as tabla, COUNT(\*) as registros FROM rol;**

***-- Verificar integridad referencial***

**SELECT COUNT(\*) FROM usuarios u**

**LEFT JOIN rol r ON u.id\_rol = r.id\_rol**

**WHERE r.id\_rol IS NULL;**

**SELECT COUNT(\*) FROM postulacion p**

**LEFT JOIN usuarios u ON p.id\_usuario = u.id\_usuario**

**WHERE u.id\_usuario IS NULL;**

**Verificar Funcionalidad Específica**

**sql**

***-- Probar queries típicas de tu aplicación***

**SELECT u.nombre, u.apellido, r.nombre\_rol**

**FROM usuarios u**

**JOIN rol r ON u.id\_rol = r.id\_rol**

**WHERE u.estado = 1;**

**SELECT c.nom\_campaña, c.descripcion, COUNT(p.id\_postulacion) as postulaciones**

**FROM campañas c**

**LEFT JOIN postulacion p ON c.id\_campaña = p.id\_campaña**

**GROUP BY c.id\_campaña;**

##### ACTUALIZACIÓN DE LA APLICACIÓN NODE.JS

**Nueva Configuración de Conexión**

**javascript**

***// config/database.js - Nueva configuración para Node.js***

**const mysql = require('mysql2/promise');**

**require('dotenv').config();**

**const dbConfig = {**

**host: process.env.DB\_HOST,**

**port: process.env.DB\_PORT || 3306,**

**user: process.env.DB\_USER,**

**password: process.env.DB\_PASSWORD,**

**database: process.env.DB\_NAME,**

**charset: 'utf8mb4',**

**timezone: '-05:00', *// Zona horaria Colombia***

***// Configuraciones específicas para RDS***

**ssl: {**

**rejectUnauthorized: false *// Para desarrollo, en producción usar certificado***

**},**

***// Pool de conexiones para mejor rendimiento***

**connectionLimit: 10,**

**acquireTimeout: 60000,**

**timeout: 60000,**

**reconnect: true**

**};**

***// Crear pool de conexiones***

**const pool = mysql.createPool(dbConfig);**

***// Función helper para ejecutar queries***

**async function query(sql, params = []) {**

**try {**

**const [results] = await pool.execute(sql, params);**

**return results;**

**} catch (error) {**

**console.error('Database query error:', error);**

**throw error;**

**}**

**}**

***// Función para cerrar conexiones***

**async function closePool() {**

**await pool.end();**

**}**

**module.exports = {**

**pool,**

**query,**

**closePool**

**};**

**Variables de Entorno (.env)**

**bash**

***# .env - Configuración de variables de entorno***

##### # Base de datos RDS

**DB\_HOST=ldm-academy-db.xxxxxx.us-east-1.rds.amazonaws.com**

**DB\_PORT=3306**

**DB\_USER=ldm\_app**

**DB\_PASSWORD=PasswordSeguroParaApp123!**

**DB\_NAME=ldm\_academy**

**DB\_CHARSET=utf8mb4**

**DB\_TIMEZONE=-05:00**

***# Configuración de aplicación***

**NODE\_ENV=production**

**PORT=3000**

***# Para desarrollo local (comentar en producción)***

***# DB\_HOST=localhost***

***# DB\_USER=root***

***# DB\_PASSWORD=tu\_password\_local***

**Ejemplo de Uso en Controladores**

**javascript**

***// controllers/usuarioController.js***

**const { query } = require('../config/database');**

***// Obtener usuarios activos***

**exports.getUsuariosActivos = async (req, res) => {**

**try {**

**const usuarios = await query(`**

**SELECT u.nombre, u.apellido, r.nombre\_rol**

**FROM usuarios u**

**JOIN rol r ON u.id\_rol = r.id\_rol**

**WHERE u.estado = ?**

**`, [1]);**

**res.json(usuarios);**

**} catch (error) {**

**res.status(500).json({ error: 'Error al obtener usuarios' });**

**}**

**};**

***// Crear nueva postulación***

**exports.crearPostulacion = async (req, res) => {**

**try {**

**const { userId, campaignId } = req.body;**

**const result = await query(`**

**INSERT INTO postulacion (id\_usuario, id\_campaña, estado, fecha\_postulacion)**

**VALUES (?, ?, ?, NOW())**

**`, [userId, campaignId, 'pendiente']);**

**res.json({**

**success: true,**

**postulacionId: result.insertId**

**});**

**} catch (error) {**

**res.status(500).json({ error: 'Error al crear postulación' });**

**}**

**};**

**Manejo de Archivos de Imágenes**

**javascript**

***// Para Node.js con multer para subir archivos***

**const multer = require('multer');**

**const AWS = require('aws-sdk');**

***// Configurar S3 para imágenes de campañas (opcional)***

**const s3 = new AWS.S3();**

**const upload = multer({ storage: multer.memoryStorage() });**

***// Las imágenes existentes en tu BD:***

**const imagenesExistentes = [**

**'campaña\_1743595226657.jpg',**

**'campaña\_1745041272531.jpg',**

**'campaña\_1745041395722.png'**

**];**

***// Función para subir imágenes a S3***

**async function subirImagenS3(archivo, nombreArchivo) {**

**const params = {**

**Bucket: 'ldm-academy-images',**

**Key: `campañas/${nombreArchivo}`,**

**Body: archivo.buffer,**

**ContentType: archivo.mimetype**

**};**

**return await s3.upload(params).promise();**

**}**

***// Middleware para manejar subida de imágenes***

**exports.subirImagenCampaña = upload.single('imagen');**

**Dependencias NPM Necesarias**

**bash**

***# Instalar dependencias requeridas***

**npm install mysql2 dotenv**

***# Opcional para manejo de archivos***

**npm install multer aws-sdk**

##### MONITOREO ESPECÍFICO

**Métricas Importantes para ldm\_academy**

**sql**

***-- Query para monitorear actividad***

**SELECT 'Usuarios activos' as metrica, COUNT(\*) as valor**

**FROM usuarios WHERE estado = 1**

**UNION ALL**

**SELECT 'Campañas activas' as metrica, COUNT(\*) as valor**

**FROM campañas WHERE estado = 1**

**UNION ALL**

**SELECT 'Postulaciones pendientes' as metrica, COUNT(\*) as valor**

**FROM postulacion WHERE estado = 'pendiente';**

**Alertas CloudWatch Recomendadas**

**DatabaseConnections > 80% del máximo**

**CPUUtilization > 70%**

**FreeStorageSpace < 2GB**

**ReadLatency > 200ms**

**ESTIMACIÓN DE COSTOS (US East-1)**

**Configuración Básica**

**db.t3.micro: ~$15/mes**

**20GB SSD: ~$2.30/mes**

**Backup automático: Gratis (hasta 20GB)**

**Total estimado: ~$17.30/mes**

**Para Producción (Recomendado)**

**db.t3.small: ~$30/mes**

**Multi-AZ: +100% (~$60/mes total)**

**50GB SSD: ~$5.75/mes**

**Total estimado: ~$65.75/mes**

**PLAN DE ROLLBACK ESPECÍFICO**

**Mantener XAMPP como Respaldo**

**bash**

***## Mantener XAMPP corriendo en paralelo durante 1 semana***

***## Backup antes de migración***

**cp -r /xampp/mysql/data/ldm\_academy /xampp/mysql/data/ldm\_academy\_backup\_$(date +%Y%m%d)**

***## Script de rollback si es necesario***

**mysql --u root --p < ldm\_academy\_backup.sql**

**CHECKLIST ESPECÍFICO**

**Pre-migración**

* **Backup de ldm\_academy realizado**
* **Backup de archivos de imágenes de campañas**
* **Instancia RDS creada y configurada**
* **Usuario ldm\_app creado con permisos correctos**
* **Conectividad probada desde servidor de aplicación**

**Durante migración**

* **Estructura migrada: 7 tablas creadas**
* **Datos migrados: ~50 registros totales**
* **Foreign keys funcionando correctamente**
* **Charset utf8mb4 configurado**
* **Timestamps con zona horaria correcta**

**Post-migración**

* **Login de usuarios funcionando**
* **Creación de campañas funcionando**
* **Sistema de postulaciones operativo**
* **Registro de asistencias funcionando**
* **Notificaciones enviándose correctamente**
* **Imágenes de campañas cargando bien**

**COMANDOS RÁPIDOS**

**Backup Express**

**bash**

**mysqldump --u root --p ldm\_academy > ldm\_academy\_$(date +%Y%m%d\_%H%M).sql**

**Restauración Express**

**bash**

**mysql --h tu-endpoint-rds --u admin --p ldm\_academy < ldm\_academy\_backup.sql**

**Verificación Express**

**bash**

**mysql --h tu-endpoint-rds --u admin --p -e "USE ldm\_academy; SHOW TABLES; SELECT COUNT(\*) FROM usuarios;"**

**Tiempo estimado de migración: 1-2 horas (BD pequeña)  
Downtime requerido: 15-30 minutos   
Dificultad: Baja (BD simple)  
Backend: Node.js con MySQL2**