

INFORME TÉCNICO DE DESARROLLO

Proyecto: Sistema de Gestión de Grados y Títulos Institución: Universidad Peruana Los Andes (UPLA) Tecnología: Java Web (Jakarta EE) + MySQL

1. REPORTE DE ARQUITECTURA DE DATOS

Para la persistencia de datos del sistema, se diseñó una base de datos relacional normalizada utilizando MySQL Server 8.0. A continuación se detalla la estructura final utilizada en producción.

1.1. Esquema de Base de Datos

Nombre de la BD: proyecto_final_db

Motor: InnoDB (Soporte para claves foráneas y transacciones).

Colación: utf8mb4_general_ci (Soporte para caracteres especiales y tildes).

1.2. Diccionario de Datos (Tablas)

Tabla: usuarios

Almacena la información de todos los actores del sistema. Se utiliza una columna discriminadora (rol) para diferenciar los permisos.

INFORME TÉCNICO DE DESARROLLO

Proyecto: Sistema de Gestión de Grados y Títulos Institución: Universidad Peruana Los Andes (UPLA) Tecnología: Java Web (Jakarta EE) + MySQL

1. REPORTE DE ARQUITECTURA DE DATOS

Para la persistencia de datos del sistema, se diseñó una base de datos relacional normalizada utilizando MySQL Server 8.0. A continuación se detalla la estructura final utilizada en producción.

1.1. Esquema de Base de Datos

Nombre de la BD: proyecto_final_db

Motor: InnoDB (Soporte para claves foráneas y transacciones).

Colación: utf8mb4_general_ci (Soporte para caracteres especiales y tildes).

1.2. Diccionario de Datos (Tablas)

Tabla: usuarios

Almacena la información de todos los actores del sistema. Se utiliza una columna discriminadora (rol) para diferenciar los permisos.

Columna	Tipo de Dato	Restricciones	Descripción
id	INT	PK, AI	Identificador único del usuario.
nombre	VARCHAR(100)	NOT NULL	Nombre completo del usuario.
email	VARCHAR(100)	UNIQUE	Correo institucional (@upla.edu.pe).
password	VARCHAR(50)	NOT NULL	Credencial de acceso.
rol	VARCHAR(20)	NOT NULL	Define el nivel de acceso (admin, docente, alumno).

Tabla: tesis

Registra los proyectos de grado, vinculando a un alumno (autor) y a un docente (jurado).

Columna	Tipo de Dato	Restricciones	Descripción
id_tesis	INT	PK, AI	Identificador único del proyecto.
titulo	VARCHAR(200)	NOT NULL	Título de la investigación.
id_alumno	INT	FK	Relación con tabla usuarios (Autor).
id_jurado	INT	FK	Relación con tabla usuarios (Jurado).
nota	INT	DEFAULT 0	Calificación numérica (0-20).
estado	VARCHAR(50)	DEFAULT 'Pendiente'	Estado del trámite (Pendiente, Calificado).

1.3. Relaciones (Diagrama Entidad-Relación)

Uno a Muchos (1:N): Un Docente puede ser jurado de múltiples Tesis.

Uno a Uno (1:1): Lógicamente, un Alumno solo puede tener una Tesis activa en este periodo (controlado por lógica de negocio).

2. REPORTE DE INCIDENCIAS Y SOLUCIONES TÉCNICAS

Durante el ciclo de desarrollo del software, se presentaron diversos desafíos técnicos ("complicaciones") relacionados con la configuración del entorno, la conexión a datos y la lógica de negocio. A continuación se documentan los errores críticos y sus soluciones.

Incidencia Nº 1: Conflicto de Dependencias (Jakarta vs Javax)

Descripción del Problema: Al intentar compilar el proyecto, el servidor Apache Tomcat arrojaba errores de clase no encontrada (ClassNotFoundException) para HttpServlet.

Causa Raíz: Incompatibilidad de versiones. Se estaba utilizando Tomcat 10 (que usa jakarta.servlet) con librerías antiguas de Java EE 8 (que usan javax.servlet).

Solución Implementada: Se migró el archivo pom.xml para utilizar la dependencia jakarta.jakartaee-web-api versión 10.0.0, estandarizando todo el proyecto a la nueva especificación Jakarta EE.

: Incidencia Nº 2 Fallo en Conexión a Base de Datos (Error 500)

Descripción del Problema: Al intentar iniciar sesión, el sistema arrojaba una excepción NullPointerException en el DAO de usuarios.

Causa Raíz: El driver JDBC de MySQL no estaba siendo empaquetado correctamente en el archivo WAR, provocando que Conexion.getConexion() devolviera null.

Solución Implementada:

Se agregó explícitamente la dependencia mysql-connector-j versión 8.3.0 en Maven.

Se aplicó el procedimiento de "Clean and Build" para forzar la descarga e inclusión de la librería.

Incidencia Nº 3: Inconsistencia en Mapeo de Datos (Column Names)

Descripción del Problema: El inicio de sesión indicaba "Usuario incorrecto" a pesar de que las credenciales eran válidas en la base de datos.

Causa Raíz: Error humano en el mapeo Objeto-Relacional (ORM manual). El código Java buscaba la columna nombres (plural), mientras que la base de datos tenía la columna nombre (singular).

Solución Implementada: Se realizó una auditoría del código SQL dentro de UsuarioDAO.java para alinear los nombres de las columnas con el esquema exacto de MySQL Workbench.

Incidencia Nº 4: Manejo de Valores Nulos (Null Safety)

Descripción del Problema: Cuando un usuario tenía un rol mal definido o vacío, el sistema colapsaba con una pantalla de error 500 al intentar redirigir.

Causa Raíz: Intentar ejecutar el método `.equals()` sobre una variable que venía null desde la base de datos.

Solución Implementada: Se refactorizó la lógica del LoginServlet utilizando "programación defensiva" (comparación inversa constantes vs variables) y se añadieron validaciones previas para asegurar que el objeto Usuario no fuera nulo antes de leer sus propiedades.

3. TECNOLOGÍAS UTILIZADAS

Este reporte certifica el uso de las siguientes herramientas en la construcción del software:

Lenguaje de Programación: Java (JDK 17/21).

Tecnología Web: Jakarta EE 10 (Servlets & JSP).

Gestor de Dependencias: Apache Maven.

Base de Datos: MySQL Community Server 8.0.

Frontend: HTML5, CSS3, Bootstrap 5.3 (Diseño Responsivo), SweetAlert2.

Entorno de Desarrollo (IDE): Apache NetBeans 21.

Servidor de Aplicaciones: Apache Tomcat 10.1.