

Trabajo Practico Base de Datos

Título: Integración de Bases de Datos Relacionales y NoSQL

Entrega: Última semana de clases

Formato: Trabajo grupal

Versión final asistida por ChatGPT

Objetivo:

El propósito de este trabajo es que los estudiantes comprendan las diferencias prácticas entre bases de datos relacionales y NoSQL, desarrollando una aplicación web que interactúe con ambos tipos de bases de datos. Se debe implementar una interfaz que permita realizar operaciones de alta, baja, modificación y consulta en ambas bases.

Requerimientos Técnicos:

1. Tecnología del Frontend:

- Los estudiantes pueden utilizar cualquier framework o biblioteca que facilite la creación de una interfaz web, como **React**, **Vue**, **Bootstrap** o **HTML5** con **JavaScript**.
- La interfaz debe contar con un formulario de entrada para agregar nuevos datos, campos para modificar información existente y botones para eliminar registros.

2. Backend y Conexión con Bases de Datos:

- La aplicación debe ser desarrollada en un lenguaje de programación de elección libre (ej. **Python** con **Flask** o **Django**, **JavaScript** con **Node.js**, **PHP**, **Java**, etc.).
- Deberán realizarse conexiones a dos bases de datos:
 - **Base de Datos Relacional:** MySQL, PostgreSQL, o SQLite. Esta base se usará para datos estructurados y normalizados.
 - **Base de Datos NoSQL:** MongoDB, Firebase, o CouchDB. Esta se empleará para almacenar información menos estructurada, ideal para datos flexibles o no necesariamente normalizados.

3. Funciones CRUD (Crear, Leer, Actualizar, Eliminar):

- La aplicación debe permitir realizar operaciones CRUD en ambas bases de datos:

- **Crear registros:** Ingresar datos nuevos mediante un formulario.
- **Leer registros:** Visualizar los datos guardados en una tabla o lista.
- **Actualizar registros:** Modificar la información existente seleccionando un registro específico.
- **Eliminar registros:** Borrar un registro existente.

4. Desarrollo de la Interfaz:

- La interfaz debe mostrar los datos de ambas bases de datos de manera separada pero en una misma página o sección.
- Utilizar un diseño que permita ver con claridad el origen de los datos y facilite las operaciones de actualización y eliminación.
- **Recomendación:** Utilizar una vista de lista o tabla para cada conjunto de datos y añadir un botón que permita visualizar los detalles de cada registro para actualizarlo o eliminarlo.

Informe Final:

Cada grupo debe entregar un informe final donde describan:

1. Elección de las Tecnologías:

- Explicación de los lenguajes, frameworks y bases de datos elegidos, justificando su elección.

2. Diagrama de Arquitectura:

- Un diagrama que muestre la arquitectura general de la aplicación, incluyendo la comunicación entre el frontend, el backend y ambas bases de datos.

3. Configuración y Conexión a Bases de Datos:

- Explicación de cómo configuraron y conectaron la aplicación a cada base de datos.
- Descripción de las librerías utilizadas para manejar las conexiones (por ejemplo, **SQLAlchemy** para relacional en Python o **Mongoose** para MongoDB en Node.js).

4. Descripción de Funcionalidades CRUD:

- Descripción detallada de cómo implementaron cada operación CRUD en ambas bases de datos, explicando las diferencias clave en el manejo de los datos relacionales y NoSQL.

5. Comparación entre Bases de Datos Relacionales y NoSQL:

- Ventajas y desventajas observadas al trabajar con cada tipo de base de datos en el contexto de la aplicación.
- Reflexión sobre situaciones en las que sería más conveniente usar una base de datos relacional frente a una NoSQL, y viceversa.

6. Dificultades y Aprendizajes:

- Detalle de los principales desafíos que enfrentaron durante el desarrollo y cómo los resolvieron.
- Conclusiones y aprendizajes personales o de equipo.

Entrega del Proyecto:

1. Código Fuente:

- Entregar todo el código de la aplicación en un repositorio de GitHub o similar, con instrucciones claras para la instalación y ejecución.

2. Informe Documentado:

- Presentar un documento en formato PDF con el informe final que incluya capturas de pantalla de la aplicación funcionando, especialmente de las operaciones CRUD en ambas bases de datos.

3. Presentación Oral (Opcional si es solicitado):

- Cada grupo puede presentar brevemente su proyecto, describiendo su enfoque y resaltando las características principales de su aplicación.

Evaluación:

La evaluación se basará en:

- Cumplimiento de los requerimientos técnicos.
- Calidad y funcionalidad de la interfaz web.
- Integración y manejo correcto de ambas bases de datos.
- Claridad y profundidad del informe.
- Resolución de problemas y aprendizaje demostrado.

Como opcional se puede agregar el manejo de Blockchain para insertar y consultar un dato dado.