

GUÍA DIDÁCTICA Y TEMARIO DE PROFUNDIZACIÓN

Esta guía resume los fundamentos conceptuales de cada tema y contiene las preguntas clave que cada estudiante debe profundizar y responder durante su video-exposición (10 a 15 minutos).

MÓDULO 1: Fundamentos y Paradigma Relacional

Asignado a: Roberth Alonzo Campechano Perez

1. Resumen Teórico Base

- **Persistencia vs. Volatilidad:** Las variables de una aplicación web residen en la memoria RAM del servidor; al terminar una petición HTTP o reiniciarse el servicio, esos datos se pierden. Las bases de datos ofrecen almacenamiento no volátil y persistente en disco secundario.
- **SGBD (DBMS):** Software que actúa como intermediario entre la aplicación web y los archivos físicos en disco, gestionando concurrencia, permisos, transacciones y consultas.
- **Modelo Relacional (Codd, 1970):** La información se organiza de forma matemática en relaciones (tablas), compuestas por tuplas (filas/registros) y atributos (columnas/campos), regidas por un esquema estricto predefinido.
- **Integridad de Datos:** Reglas que impiden guardar datos corruptos o huérfanos. Las claves primarias (*PK*) garantizan unicidad; las claves foráneas (*FK*) vinculan datos entre tablas.
- **Propiedades ACID:** Conjunto de cuatro garantías para operaciones transaccionales:
 - **A (Atomicidad):** La operación se completa al 100% o no se aplica nada (todo o nada).
 - **C (Consistencia):** Los datos pasan de un estado válido a otro respetando todas las reglas del esquema.
 - **I (Aislamiento):** Las transacciones simultáneas no interfieren entre sí.
 - **D (Durabilidad):** Una vez confirmada la transacción, los datos persisten aunque ocurra un fallo eléctrico.

2. Puntos Críticos a Profundizar en la Exposición

- ¿Cómo fluyen los datos entre el usuario web, el backend y el motor de base de datos?
- ¿Por qué es un error crítico diseñar tablas sin definir correctamente los tipos de datos (por ejemplo, guardar números o fechas como texto)?
- ¿Qué problema ocasionaría en una tienda web la falta de integridad referencial (*FK*) al eliminar un usuario que ya tiene pedidos registrados?
- ¿Por qué el modelo ACID es obligatorio en pasarelas de pago y sistemas bancarios frente a modelos más flexibles?

MÓDULO 2: Paradigma No Relacional (NoSQL) y Análisis Técnico

Asignado a: Williams Alejandro Jimenez Pelayo

1. Resumen Teórico Base

- **Origen de NoSQL:** Nació a principios de los 2000 por la necesidad de procesar enormes volúmenes de datos web no estructurados (redes sociales, logs, clics) y requerimientos de alta disponibilidad donde los bloqueos transaccionales tradicionales generaban cuellos de botella.
- **Esquema Dinámico (*Schema-less*):** No exige una estructura de columnas fija. Dos registros en la misma colección pueden poseer atributos completamente distintos.
- **Familias de Bases NoSQL:**
 - **Documentales:** Almacenan datos jerárquicos estructurados (JSON/BSON). Un documento puede contener arreglos y subdocumentos embebidos.
 - **Clave-Valor:** Diccionarios directos donde a una clave única le corresponde un valor arbitrario; ofrecen lecturas y escrituras ultrarrápidas.
 - **Columnas Anchas:** Tablas multidimensionales donde las filas pueden tener distintas columnas; optimizadas para consultas analíticas masivas sobre billones de filas.
 - **Grafos:** Modelan entidades como nodos y sus interconexiones como aristas con propiedades; ideales para relaciones complejas.
- **Escalabilidad:**
 - **Vertical (Scale-up):** Comprar un servidor con más CPU, RAM y almacenamiento; tiene un límite físico y económico.
 - **Horizontal (Scale-out):** Distribuir la carga de datos entre decenas o cientos de servidores económicos en clúster.

2. Puntos Críticos a Profundizar en la Exposición

- ¿Por qué la rigidez de un esquema relacional tradicional se convirtió en un obstáculo para aplicaciones web con iteraciones rápidas de producto?
- ¿Cómo cambia el modelado al almacenar un pedido completo en un único documento JSON embebido en lugar de fragmentarlo en 4 tablas relacionales distintas?
- ¿Qué ventajas ofrece el modelo Clave-Valor para guardar sesiones de usuario web y carritos de compra frente a una tabla relacional?
- ¿Qué criterios técnicos justifican utilizar una base de datos relacional y cuáles justifican implementar una NoSQL en una aplicación web moderna?