# MYSQL X ANDROID

## CYBERDONK

APLICAIONES MOVILES - 5IV9

Chávez Rincón Gabriel
Cruz Ortiz Santiago
Hernández Reyes Brandon
López Ruiz Jesús Enrique
Vargas Hernández Alonso

# ¿QUÉ ES LA TRANSACCIONALIDAD EN UNA APLICACIÓN MÓVIL?

La transaccionalidad consiste en que una aplicación pueda enviar, recibir, modificar y consultar datos que están almacenados en una base de datos centralizada, garantizando consistencia e integridad. En Android, esto se logra conectando la app con un servidor que maneja la comunicación con MySQL.

# ¿POR QUÉ ANDROID NO PUEDE CONECTARSE DIRECTAMENTE A MYSQL?

Android no tiene permitido conectarse a MySQL directamente porque sería:

- Inseguro (expondrías la IP de tu base de datos).
- Ineficiente.
- Fácil de atacar (inyecciones SQL, accesos indebidos).

Por eso se utiliza un servidor intermedio, normalmente hecho con PHP, que actúa como API.

# ARQUITECTURA CLIENTE-SERVIDOR

Android (Cliente)
↓ (HTTP / JSON)
Servidor PHP (API/Backend)
↓ (Consultas SQL)
MySQL (Base de Datos Centralizada)
Este diseño garantiza:

- Seguridad
- Escalabilidad
- Control de permisos
- Manejo correcto de transacciones

# ELEMENTOS NECESARIOS PARA CREAR UNA APP ANDROID CON CONEXIÓN A MYSQL

## BASE DE DATOS MYSQL:

- Tablas correctamente estructuradas.
- Llaves primarias y foráneas.
- Campos con tipos de dato adecuados.
- Motor InnoDB (permite transacciones).

## BASE DE DATOS MYSQL:

El backend PHP es responsable de:

1. Conectarse a MySQL
2. Recibir peticiones desde Android (GET/POST)
3. Ejecutar consultas SQL
4. Devolver la respuesta a la app en formato JSON

Una API básica incluye archivos PHP como:

- insert.php (insertar datos)
- update.php (modificar datos)
- delete.php (eliminar datos)
- select.php (consultar datos)

Todos son necesarios para la transaccionalidad.

# ELEMENTOS NECESARIOS PARA CREAR UNA APP ANDROID CON CONEXIÓN A MYSQL

## SERVIDOR LOCAL O EN LA NUBE

Puede ser:

- XAMPP (local)
- Hosting compartido
- VPS
- AWS / Azure / Google Cloud

Debe permitir:

- PHP
- MySQL
- Acceso por URL a los scripts

## CÓDIGO ANDROID – CLIENTE

La app Android debe:

- Realizar peticiones HTTP (GET/POST)
- Enviar parámetros (JSON o Form-Data)
- Recibir datos JSON
- Interpretar la respuesta
- Mostrar información al usuario
- Manejar errores, desconexiones, timeouts

# ELEMENTOS NECESARIOS PARA CREAR UNA APP ANDROID CON CONEXIÓN A MYSQL

## LIBRERÍAS DE COMUNICACIÓN EN ANDROID

Las más utilizadas:

- Volley (simple y rápido)
- Retrofit (profesional, recomendado)

Ambas permiten enviar solicitudes HTTP hacia el servidor PHP.

# FLUJO DE UNA TRANSACCIÓN COMPLETA

## Registrar un usuario.

- El usuario ingresa datos en Android.
- La app envía petición POST → insert.php.
- PHP recibe los datos.
- PHP valida y ejecuta consulta SQL en MySQL.
- MySQL responde al servidor.
- PHP genera una respuesta JSON.
- Android recibe JSON e informa al usuario.

Así se completa una transacción.

# SEGURIDAD EN LA TRANSACCIONALIDAD

Muy importante para cumplir requisitos escolares y profesionales:

- No exponer credenciales MySQL en Android.
- Usar consultas preparadas para evitar inyecciones SQL.
- Validar parámetros antes de ejecutar consultas.
- Usar HTTPS cuando la app esté en producción.
- No aceptar peticiones desde IPs desconocidas.

# ERRORES MÁS COMUNES Y CÓMO EVITARLOS

- “No se puede conectar a la base de datos” → Revisar credenciales y puerto.
- “Error al obtener JSON” → PHP no está devolviendo JSON válido.
- “La app se cierra” → No se manejó error de conexión.
- “Timeout” → Servidor lento o URL incorrecta.

# BUENAS PRÁCTICAS RECOMENDADAS

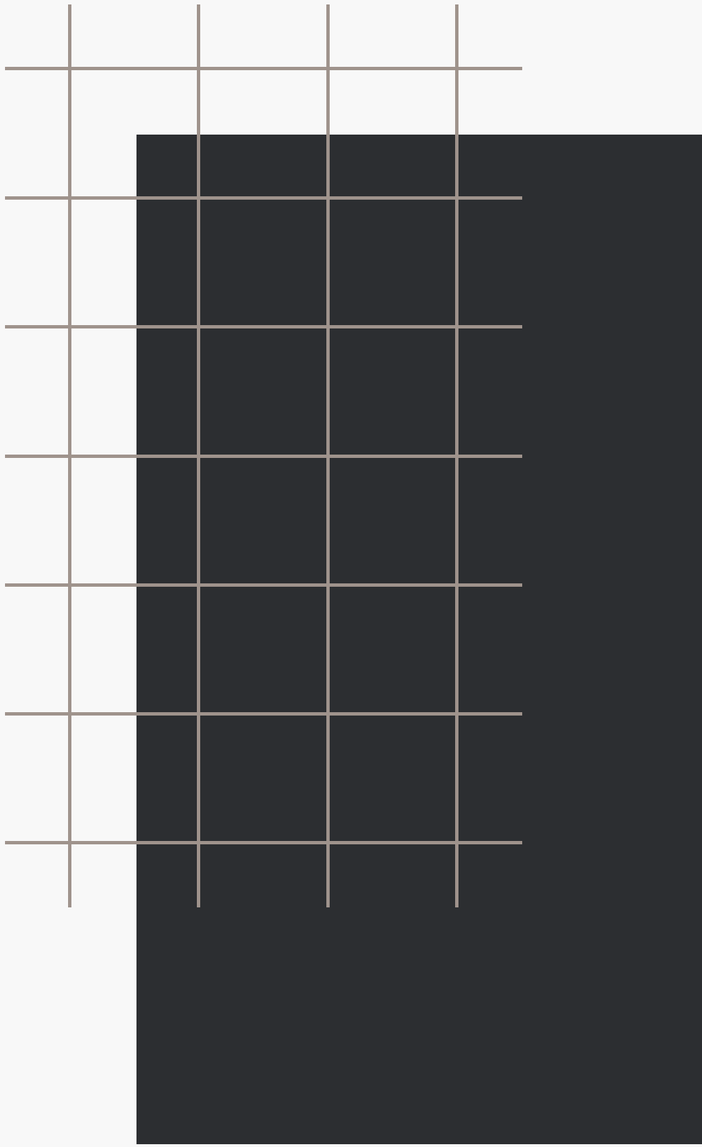
- Mantener el backend separado del código Android.
- Documentar la API (endpoints, parámetros, respuestas).
- Usar nombres consistentes en tablas y campos.
- Probar primero en Postman antes de usar Android Studio.
- Loguear todos los errores del servidor.

# CONCLUSIÓN

Para que una aplicación Android tenga transaccionalidad con una base de datos centralizada MySQL, es necesario implementar:

- Una estructura de base de datos bien diseñada
- Un servidor intermediario (API en PHP)
- Una comunicación segura mediante HTTP y JSON
- Una app Android capaz de enviar y recibir datos
- Medidas de seguridad y validación

Con estos elementos, cualquier equipo puede crear aplicaciones escalables, seguras y eficientes, como sistemas POS, inventarios, farmacias, agendas, ventas, etc.



# MUCHAS GRACIAS