



UPLA



ARQUITECTURAS DE BASE DE DATOS Y SU APLICABILIDAD TECNOLOGICA

ESPEJO QUISPE LUIS ENRIQUE
TAPIA ESPIRITU CRISTOPHER VALENTINO

CONTENIDOS

- 01.** TIPOS DE ARQUITECTURAS DE BASE DE DATOS
- 02.** CRITERIOS DE SELECCIÓN DE UNA ARQUITECTURA
- 03.** MODELO CONCEPTUAL Y LÓGICO DE DATOS
- 04.** NORMALIZACIÓN Y OPTIMIZACIÓN DEL DISEÑO





1

TIPOS DE ARQUITECTURAS DE BASE DE DATOS

CLIENTE - SERVIDOR

VENTAJAS



CONTROL TOTAL

La empresa tiene control completo sobre los servidores, datos y la infraestructura.



PERSONALIZACIÓN

Se puede configurar el hardware y software según necesidades específicas.



COSTOS FIJOS

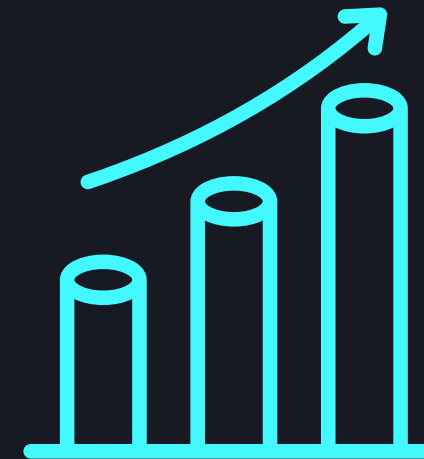
Inversión inicial en hardware y licencias, con menos costos variables a largo plazo.

DESVENTAJAS



ALTA INVERSIÓN INICIAL

Comprar servidores, licencias y configurar la infraestructura puede ser caro.



ESCALABILIDAD LIMITADA

Para aumentar capacidad, se necesitan comprar más hardware, lo que puede tardar y ser costoso.



MANTENIMIENTO

Requiere personal técnico para mantener y actualizar la infraestructura.

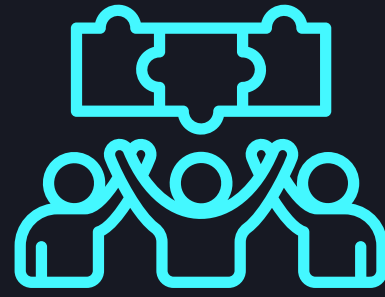
ARQUITECTURA EN LA NUBE (CLOUD COMPUTING)

VENTAJAS



MANTENIMIENTO EXTERNALIZADO

No necesitas equipo técnico para servidores físicos, el proveedor se encarga.



ESCALABILIDAD DINÁMICA

Se puede aumentar o disminuir recursos fácilmente según demanda (picos de tráfico en ventas).



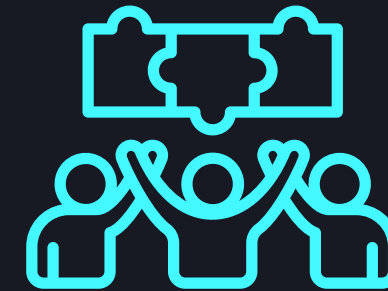
MENOR INVERSIÓN INICIAL

Se paga por uso (modelo SaaS o IaaS), reduciendo gastos en hardware.



ALTA DISPONIBILIDAD

Los proveedores en la nube suelen garantizar alta disponibilidad y redundancia.



ACCESO GLOBAL

Fácil acceso desde cualquier lugar con internet, ideal para eCommerce globales.



INTEGRACIÓN RÁPIDA

Facilita usar servicios adicionales (pagos, analytics, marketing).

DESVENTAJAS



MENOR CONTROL

Dependencia del proveedor para datos, seguridad y configuración.



COSTOS VARIABLES

Puede ser difícil predecir costos mensuales, especialmente con picos altos.



SEGURIDAD Y PRIVACIDAD

Aunque suele ser segura, hay riesgos de vulnerabilidad y cumplimiento legal (GDPR, etc.).



CONECTIVIDAD

Requiere conexión constante y estable a internet para operar.



2

CRITERIOS DE SELECCIÓN DE UNA ARQUITECTURA

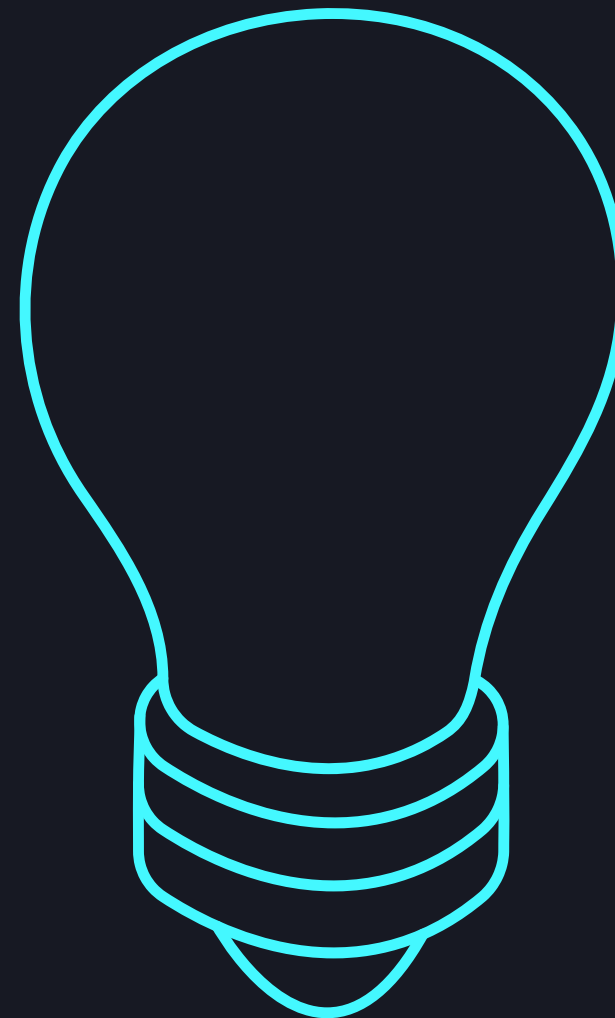
CRITERIOS DE SELECCION DE STARTUP DE SOFTWARE EDUCATIVO

Los criterios de selección de startups permitirán la delimitación de aquellos proyectos que cumplen con las características esenciales para su viabilidad, impacto y crecimiento, facilitando la identificación de iniciativas con mayor potencial de éxito en el sector del software educativo.



INNOVACIÓN Y DIFERENCIACIÓN

- ¿El producto ofrece una solución nueva o significativamente mejorada para un problema educativo?
- ¿Tiene características únicas que lo distinguen de la competencia?

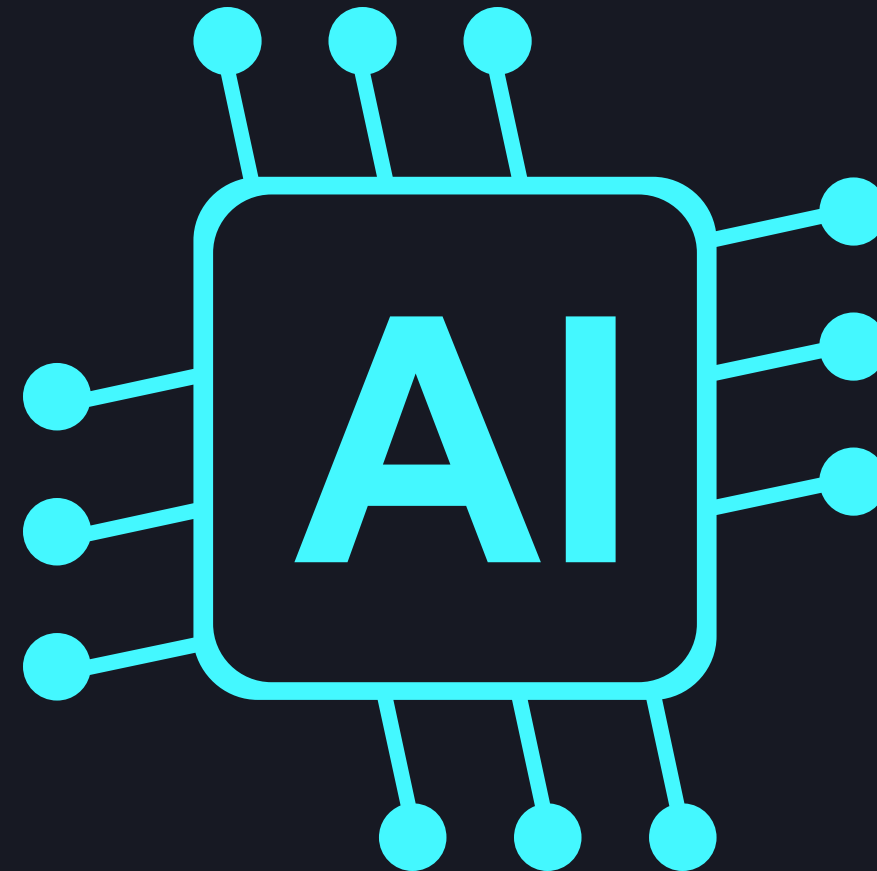


CALIDAD PEDAGÓGICA

- ¿Está basado en teorías educativas sólidas y buenas prácticas pedagógicas?
- ¿Facilita el aprendizaje efectivo y medible?
- ¿Incluye adaptaciones para diferentes estilos o ritmos de aprendizaje?

VALIDACIÓN Y RESULTADOS

- ¿Cuenta con pruebas piloto, estudios o testimonios que demuestren efectividad?
- ¿Tiene métricas claras de impacto en el aprendizaje o en la mejora de habilidades?

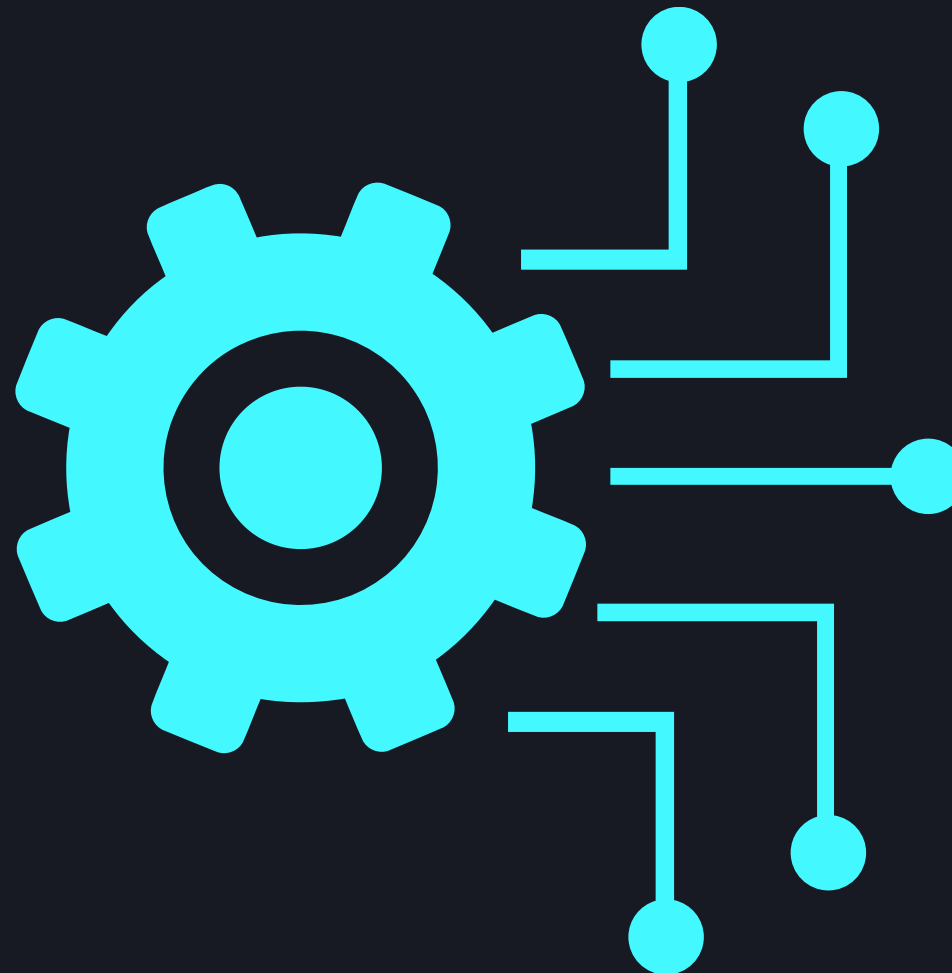


ESCALABILIDAD Y TECNOLOGÍA

- ¿La plataforma es escalable y puede adaptarse a un número creciente de usuarios?
- ¿Usa tecnología robusta y actualizada (cloud, AI, etc.)?
- ¿Es multiplataforma (web, móvil)?

EXPERIENCIA DEL USUARIO (UX/UI)

- ¿Es intuitivo, accesible y atractivo para estudiantes y profesores?
- ¿Cumple con estándares de accesibilidad para personas con discapacidades?



EQUIPO FUNDADOR

- ¿El equipo tiene experiencia en educación, tecnología y/o emprendimiento?
- ¿Demuestran compromiso y capacidad para ejecutar el proyecto?

POTENCIAL DE MERCADO

- ¿Existe una demanda clara y creciente para este tipo de software educativo?
- ¿Se dirige a un nicho específico o mercado amplio?
- ¿El modelo de negocio es viable y sostenible (suscripciones, licencias, etc.)?



CUMPLIMIENTO LEGAL Y ÉTICO

- ¿Cumple con normativas de protección de datos y privacidad (ej. GDPR)?
- ¿Promueve contenido inclusivo y ético?

INTEGRACIÓN CON ECOSISTEMAS EXISTENTES

- ¿Puede integrarse con otras plataformas educativas o sistemas LMS (Learning Management System)?
- ¿Ofrece API o compatibilidad con otros recursos digitales?



RETROALIMENTACIÓN Y MEJORA CONTINUA

- ¿Tiene un mecanismo para recibir y aplicar feedback de usuarios?
- ¿Demuestra evolución continua del producto?



3

MODELO CONCEPTUAL Y LÓGICO DE DATOS

ENTIDADES Y ATRIBUTOS

1. Cliente: representa a los clientes del banco.

- cliente_id (Clave Primaria)
- nombre
- apellido
- direccion
- telefono
- fecha_nacimiento

2. Cuenta: representa las cuentas bancarias.

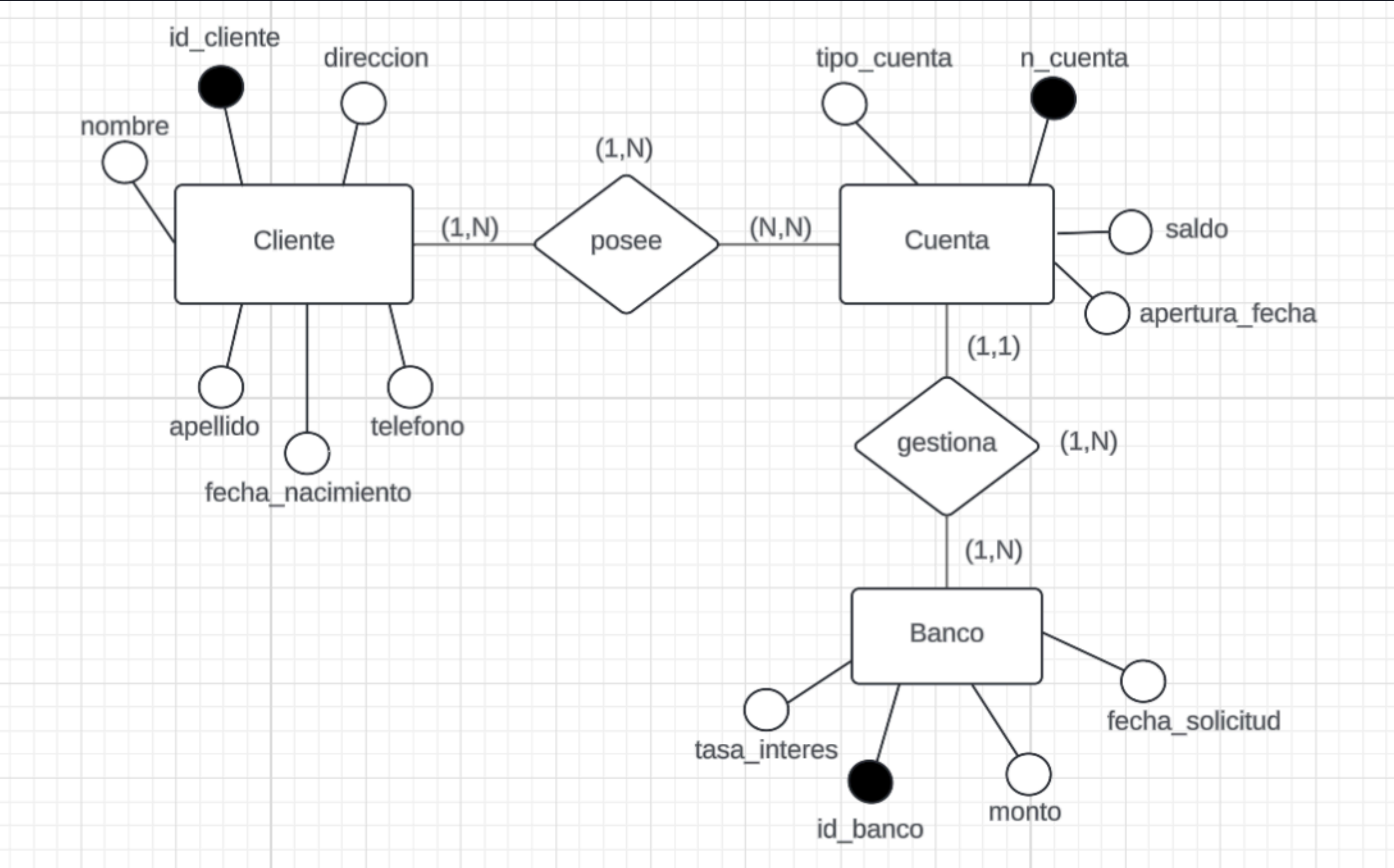
- numero_cuenta (Clave Primaria)
- tipo_cuenta (Ej: Ahorros, Corriente)
- saldo
- fecha_apertura

3. Banco: representa las sucursales físicas del banco.

- Banco_id (Clave Primaria)
- nombre_sucursal
- ciudad
- direccion



CREAR UN DIAGRAMA E-R DE UN SISTEMA BANCARIO





4

NORMALIZACIÓN Y OPTIMIZACIÓN DEL DISEÑO

CONCEPTOS CLAVE

La normalización es un proceso mediante el cual se organizan los datos en tablas siguiendo reglas conocidas como formas normales. Su objetivo es eliminar redundancia, asegurar integridad y facilitar el mantenimiento.





INDICE DE BASE DE DATOS

Un índice es una estructura que mejora la velocidad de las consultas en una tabla, similar al índice de un libro que te ayuda a encontrar información rápidamente sin tener que revisar todo el contenido.

¿Por qué crear índices?

- Para acelerar consultas frecuentes.
- Para optimizar búsquedas por columnas específicas (ejemplo: buscar por ID de producto, fecha, cliente, etc.).
- Para mejorar el rendimiento en operaciones JOIN, WHERE, ORDER BY.

INDICES

01.

Índice idx_fecha_venta

- Columnas: fecha_venta
- Funcionalidad: Este índice acelera las consultas que filtran o buscan ventas por una fecha específica.

02.

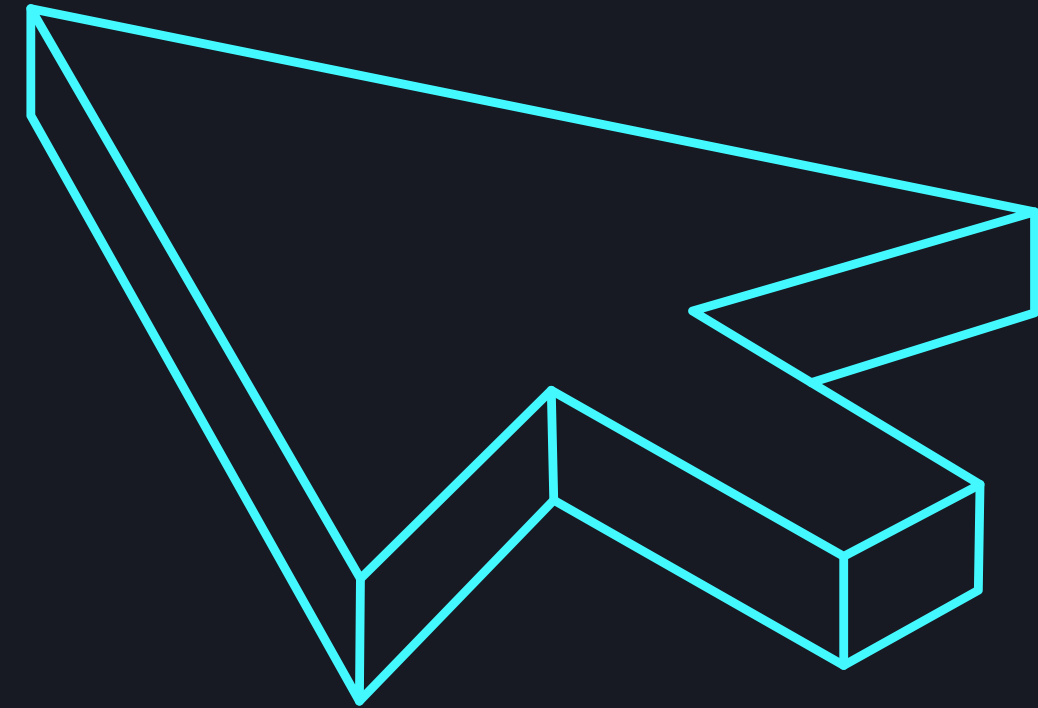
Índice idx_id_cliente

- Columnas: id_cliente
- Funcionalidad: Optimiza las búsquedas de ventas relacionadas con un cliente específico. Es útil cuando quieres obtener el historial de compras de un cliente o generar reportes de ventas por cliente.

03.

Índice idx_id_producto

- Columnas: id_producto
- Funcionalidad: Mejora el rendimiento de consultas que filtran por producto, como cuando quieres conocer las ventas de un producto en particular, ayudando a que estas búsquedas sean más rápidas.





SIMULACIÓN EN SQL SERVER

```
CREATE DATABASE sistema;

USE sistema;

-- Crear la tabla ventas con columnas básicas
CREATE TABLE ventas (
    id_venta INT PRIMARY KEY,
    id_cliente INT NOT NULL,
    id_producto INT NOT NULL,
    fecha_venta DATE NOT NULL,
    cantidad INT,
    total DECIMAL(10, 2)
);

-- Crear índice simple en la columna fecha_venta
CREATE INDEX idx_fecha_venta ON ventas(fecha_venta);

-- Crear índice simple en la columna id_cliente
CREATE INDEX idx_id_cliente ON ventas(id_cliente);

-- Índice para mejorar consultas por producto
CREATE INDEX idx_id_producto ON ventas(id_producto);
```

¡Gracias!

