

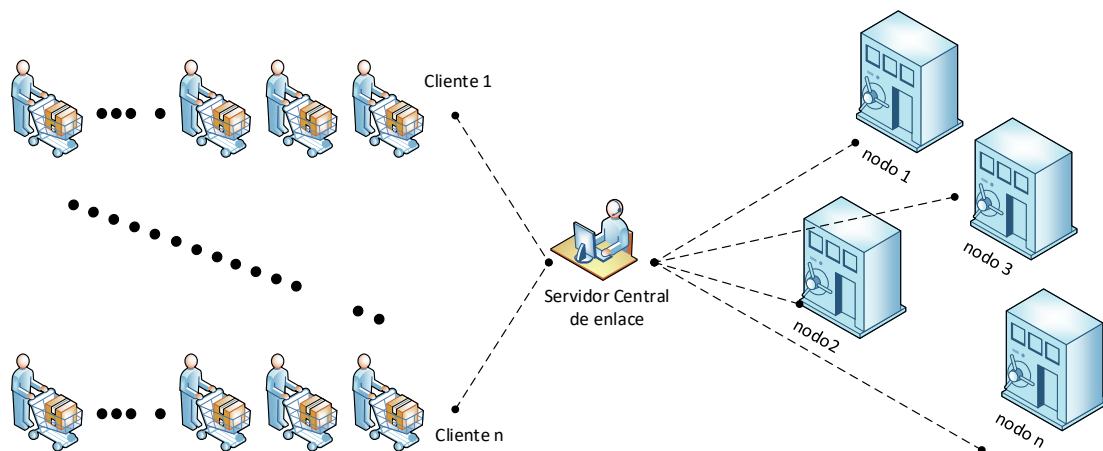
Escuela de Ciencias de la Computación
Parcial 2025-I
CC4P1 Programación Concurrente y Distribuida

Los bancos juegan un papel esencial en la economía global, proporcionando servicios financieros como depósitos, préstamos y gestión de inversiones. Cada operación bancaria involucra la transferencia de dinero entre cuentas, asegurando la seguridad y la confiabilidad. Donde se tiene las siguientes aplicaciones en la industria.

- Pagos electrónicos: Transferencias a través de apps como PayPal o banca móvil.
- Compras en línea: Uso de tarjetas para adquirir productos en tiendas digitales.
- Depósitos automáticos: Transferencias programadas de nómina o pensiones.
- Trading de acciones: Transacciones de compra/venta en mercados financieros.

Los sistemas distribuidos permiten procesar datos de manera eficiente y escalable al distribuir el trabajo en múltiples nodos. Son esenciales para aplicaciones bancarias aplicando:

- Alta disponibilidad: Los clientes pueden realizar operaciones en cualquier momento.
- Reducción de latencia: Se mejora el tiempo de respuesta en transacciones.
- Escalabilidad: Soporta miles de usuarios simultáneamente.



Para garantizar seguridad y continuidad, cada partición de la base de datos se replica tres veces en n nodos trabajadores y también que puedan tener:

- Consistencia: Se asegura que entre todos los nodos se tengan la información completa.
- Tolerancia a fallos: Si un nodo falla, otro con una réplica toma su lugar.
- Balanceo de carga: Se distribuyen consultas para evitar saturación.

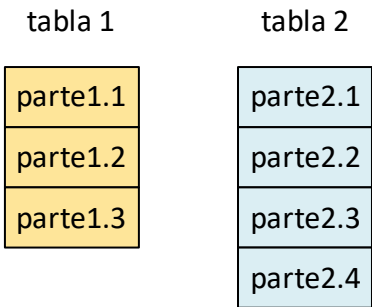
El sistema previene interrupciones mediante estrategias como:

- Replicación triple: Cada fragmento de datos tiene copias en otros nodos.

- Failover automático: Si un nodo deja de responder, el sistema redirige la consulta a otro nodo disponible, donde este replicada la información en una parte.
- Monitoreo en tiempo real: Detecta anomalías y realiza ajustes.

Estructura de las Tablas Datos Distribuida

Las tablas estarán en un archivo de formato por ejemplo txt. El sistema ahora tendrá tres tablas principales, todas las tablas particionadas en tres o más partes y replicadas tres veces en varios nodos trabajadores.



Al inicio el Se creará la tabla con miles de cuentas con sus clientes. También se tendrá una opción de arqueo de la tabla cuenta donde se sumará el dinero de todas las cuentas, para verificar el dinero del total no sufran cambios.

Tabla Cliente

Guarda información sobre los clientes del banco.

ID_CLIENTE	NOMBRE	EMAIL	TELÉFONO
1	Juan Pérez	juan@email.com	987654321
2	María López	maria@email.com	998877665

Tabla Cuenta

Registra las cuentas bancarias y su estado financiero actual.

ID_CUENTA	ID_CLIENTE	SALDO	TIPO_CUENTA
101	1	1500.00	Ahorros
102	2	3200.50	Corriente

Tabla Transacciones

Registra cada movimiento financiero realizado entre cuentas, incluyendo la validación de saldo antes de ejecutar una transferencia.

ID_TRANSACC	ID_ORIG	ID_DEST	MONTO	FECHA_HORA	ESTADO
1	101	102	500.00	2025-05-02 14:30:00	Confirmada
2	102	101	200.00	2025-05-02 15:00:00	Pendiente

Flujo del Sistema Distribuido

Clientes

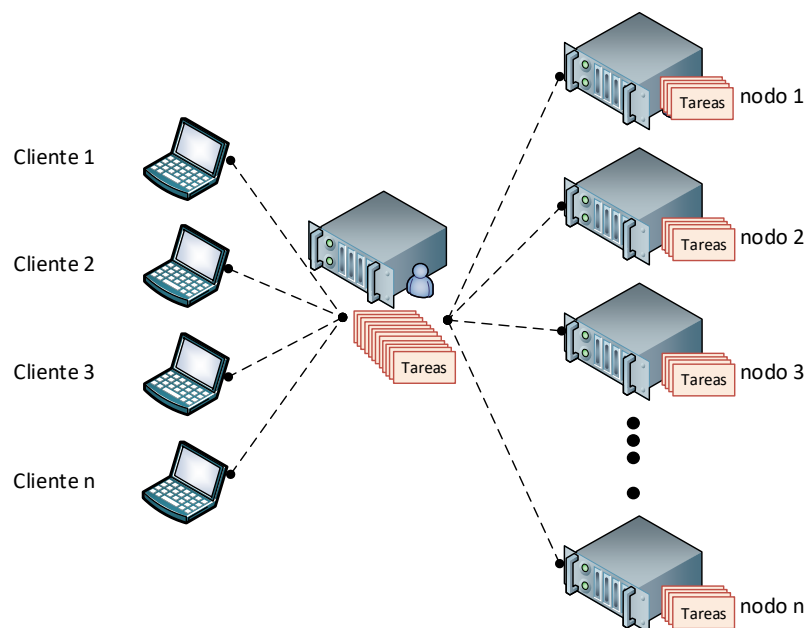
Pueden realizar múltiples transacciones con un delay aleatorio para simular concurrencia real con cientos de transacción. Envían solicitudes de consulta o transferencia al servidor central.

Servidor Central

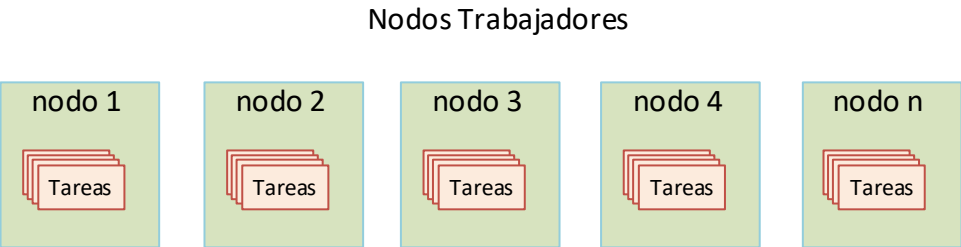
El Servidor central procesa toda la información usando threads de manera concurrente. Actúa como coordinador. Recibe solicitudes de clientes y asigna operaciones a nodos trabajadores disponibles. Si un nodo falla, reenvía la operación a otro nodo con la réplica del dato. Si un nodo falla, reenvía la operación a otro nodo. Cuando el nodo trabajador termina el servidor central recibe la respuesta del nodo trabajador y a su vez se envía el resultado al cliente como la confirmación o no.

Nodos Trabajadores

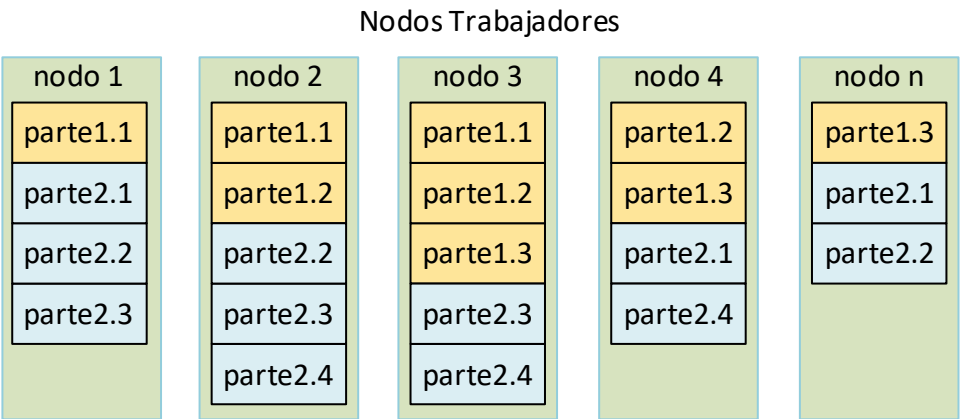
El nodo procesa las consultas o transacciones usando todos núcleos que tienen mediante threads. Cada trabajador disponible, verifica cada transacción de lectura o escritura, como la existencia de saldo antes de ejecutar transferencias y da los resultados al servidor central.



Almacenan particiones de las tablas Cliente y Cuenta. Cada tabla se divide en 3 o mas partes y se replica 3 veces en diferentes nodos. Ejecutan transacciones de lectura y actualización según lo asignado por el servidor central.



Replicación de datos



Notación para Transacciones

El sistema soporta dos tipos de transacciones:

Transacción 1: Lectura de cuenta Consulta el saldo de una cuenta específica.

OPERACIÓN: CONSULTAR_SALDO
PARÁMETROS: ID_CUENTA
SALIDA: SALDO

Ejemplo de consulta: CONSULTAR_SALDO (101) → Retorna: 1500.00

Transacción 2: Transferencia entre cuentas Realiza una transferencia de fondos entre dos cuentas, validando saldo previo.

OPERACIÓN: TRANSFERIR_FONDOS
PARÁMETROS: ID_CUENTA_ORIGEN, ID_CUENTA_DESTINO, MONTO
VALIDACIÓN: SALDO >= MONTO
ACCIONES:
- RESTAR MONTO de CUENTA_ORIGEN
- SUMAR MONTO a CUENTA_DESTINO
- REGISTRAR OPERACIÓN
SALIDA: CONFIRMACIÓN / ERROR

Ejemplo de operación: TRANSFERIR_FONDOS (101, 102, 500.00) → Retorna: Confirmación

Se verifica que la cuenta 101 tenga al menos 500.00 antes de proceder.

LP = lenguaje de programación.

Desarrollo de los Cliente y el Servidor Central (LP1) y Nodos “n” (LP2), donde se tienen que cumplir con los siguientes casos:

- Caso 1 donde LP1 = LP2 (ejem se ejecute como mínimo jdk 8) (todo el sistema distribuido paralelo funciona con java).
- Caso 2 donde LP1 <> LP2, donde los(s) Nodo(s) está en otro lenguaje que no sea Java.
- Caso 3 los que tienen grupo de tres o más agregar otros lenguajes de programación en Nodo(s) adicional(es) por participante (ejem LP3, LP4, etc.).

Desarrollar en un cluster de datos para desplegar el programa.

Desarrollando el servidor en internet o en la nube (opcional).

Describir la arquitectura diseñada.

Describir el diagrama de protocolo.

No usar websocket, no usar socket io, no usar algún middleware, no usar otras librerías de conexión u otros protocolos ya trabajados por un tercero.

Usar las Apis, librerías, sdk estándares del lenguaje de programación base.

Explicar el Desarrollo del programa.

Subir en Univirtual.

Se consultará una pregunta en su cluster o en otro cluster de clase.

Hacer pruebas donde los clientes puedan enviar cientos de transacciones regulando con un delay aleatorio entre las transacciones.

Hacer notas de recordatorio en otro idioma.

Comprimido consta

- Subir códigos fuentes de en extensión en el LP1 y LP2 (o LP3), según sea el caso.
- PDF Informe.
- PDF Presentación.
- Evaluar el desempeño con el mayor número de nodos en red con graficas.

Parte 1