

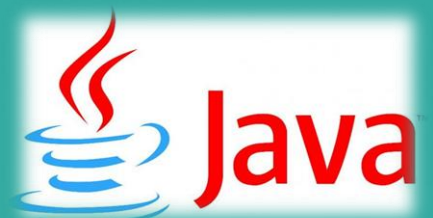
EXAMEN FINAL SISTEMAS CONCURRENTES Y DISTRIBUIDOS 2019

INTEGRANTES:

- **Ávila Santos Alex**
- **Luis Ramos Grados**
- **Víctor Ponce Pinedo**
- **Carlos Espinoza Mansilla**

10 DICIEMBRE

UNIVERSIDAD NACIONAL DE INGENIERIA



Conceptos necesarios

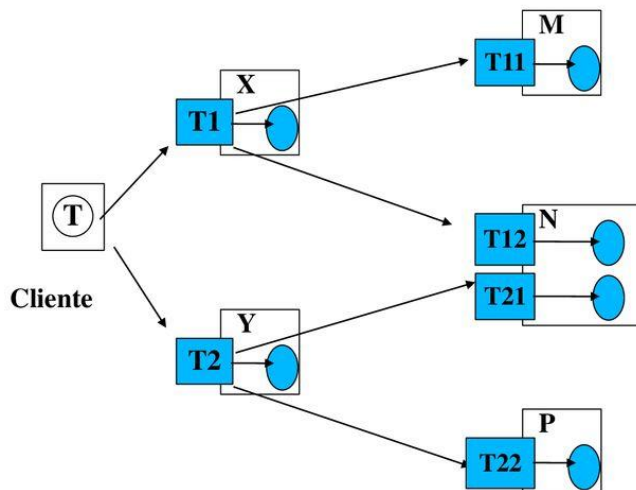
ATOMICIDAD:

La atomicidad es la propiedad que asegura que una operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias. Se dice que una operación es atómica cuando es imposible para otra parte de un sistema encontrar pasos intermedios. Si esta operación consiste en una serie de pasos, todos ellos ocurren o ninguno. Por ejemplo, en el caso de una transacción bancaria o se ejecuta tanto el depósito y la deducción o ninguna acción es realizada. Es una característica de los sistemas transaccionales.

TRANSACCIÓN:

Una transacción es una interacción con una estructura de datos compleja, compuesta por varios procesos que se han de aplicar uno después del otro. La transacción debe realizarse de una sola vez y sin que la estructura a medio manipular pueda ser alcanzada por el resto del sistema hasta que se hayan finalizado todos sus procesos.

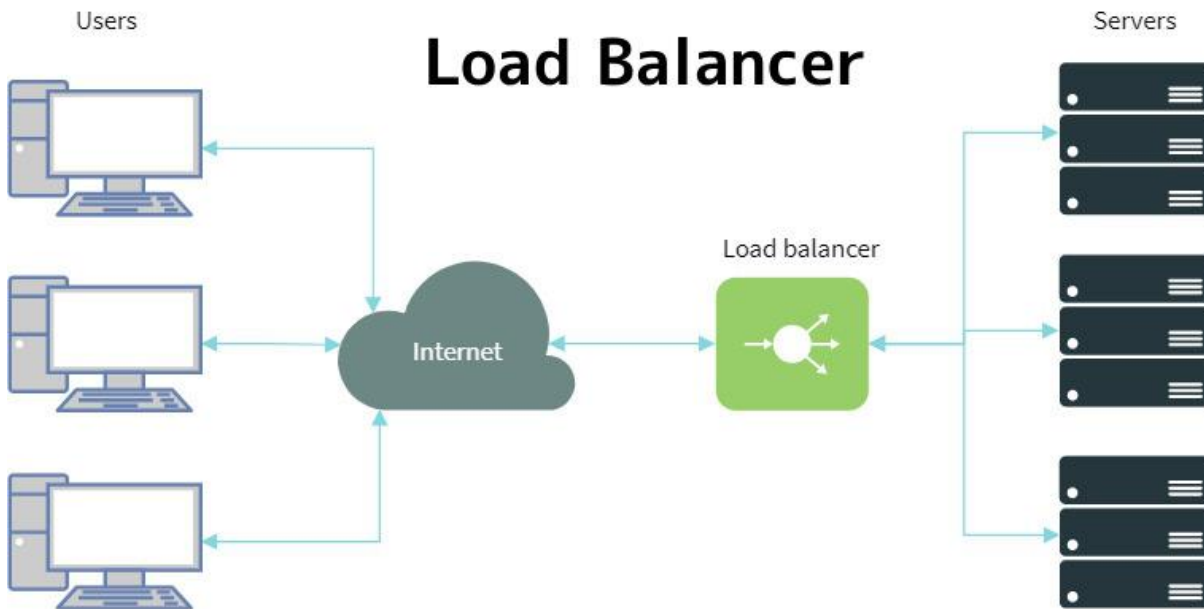
Transacciones Distribuidas



Transacción distribuida anidada

LOAD BALANCE:

Una arquitectura de balanceo de carga es un sistema que permite distribuir el trabajo computacional entre varias máquinas, con el objetivo de reducir el tiempo de respuesta global del sistema.



TOLERANCIA A FALLOS:

Si un nodo falla debe ser reemplazado por otro.

RÉPLICA:

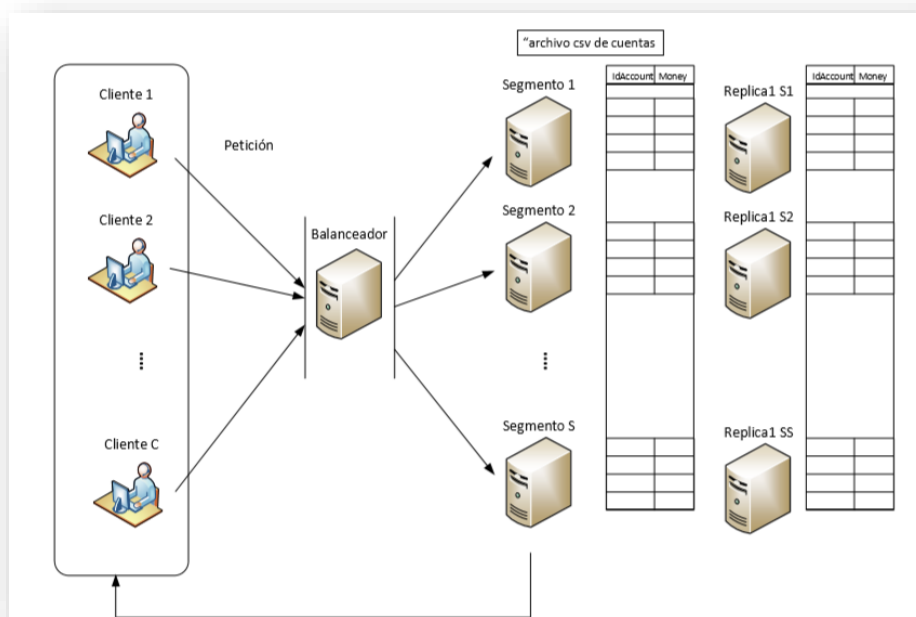
Nodos aquellos listos para actuar como reemplazo.

CONSISTENCIA:

La ejecución aislada de la transacción conserva la consistencia de la base de datos.

Arquitectura del Sistema

Nuestra arquitectura de balanceo de carga con tolerancia a fallos consiste en un número de C clientes, un servidor, un número N de nodos principales y N nodos réplica.



Al iniciar el servidor Balanceador este recibirá las peticiones de Clientes y luego distribuirá las peticiones con un algoritmo de balance las peticiones a los diferentes nodos.

En caso de que muera un nodo entra en accion su replica.

(L) Para leer el saldo se tiene la "id_solicitud" y el registro a lecturar " id_account", para poder verificar la lectura la respuesta al cliente será el saldo solicitado y el id_account.

Ejm: 02-L-1992 ; 1039

(A) Actualizar una transacción "id_solicitud" de: id_account para: id_account y la cantidad de dinero (máximo dos decimales).

Ejm: 03-A-1012 ; 1110 ; 260 ; 579.80.

Ejecutar el Sistema:

a) Componentes:

- **Clientes:**

Cliente50.java

TCPClient50.java

- **Nodos**

Nodo50.java

TCPNodo50.java

- **Balanceador de cargas**

Server50.java

TCPServer50.java

TCPServerThread50.java

b) Pasos para Correrlo:

- Primero ejecutamos el Server50.java
- Despues ejecutamos Nodo50.java n veces como el numero de nodos tengamos.
- Finalmente ejecutamos los Cliente50.java

Repositorio del Sistema

<https://drive.google.com/open?id=1zdljBNDYkFmY6Mpk4zasnUrqK-L7ArKt>