

Integrantes: Nathaly Cumbicos
David Velasco

Curso: GR2

Fecha: 20/01/2016

PROYECTO SERVIDOR DE TRANSACCIONES BANCARIAS

1. Alcance

La arquitectura a implementarse será la descrita en la figura 2. Se implementará un Servidor de transacciones bancarias el cual estará instalado en varios computadores, denominados Servidores Banco; se podrá realizar operaciones de Retiro, Depósito y Consultas asociados a los clientes del banco. Los datos provenientes de estas operaciones se almacenan un Servidor de base de datos, el cual estará instalado en cada computador servidor conjuntamente en el mismo Servidor Banco. El servidor de base de datos utilizará un esquema de base de datos distribuida para sincronizar los datos de las transacciones.

Cada servidor de transacciones bancarias contara con una sonda, la cual recopilará los datos de procesamiento (CPU, Memoria) y los almacenara en un archivo de texto en tiempo real y serán enviados al balanceador cuando este los solicite.

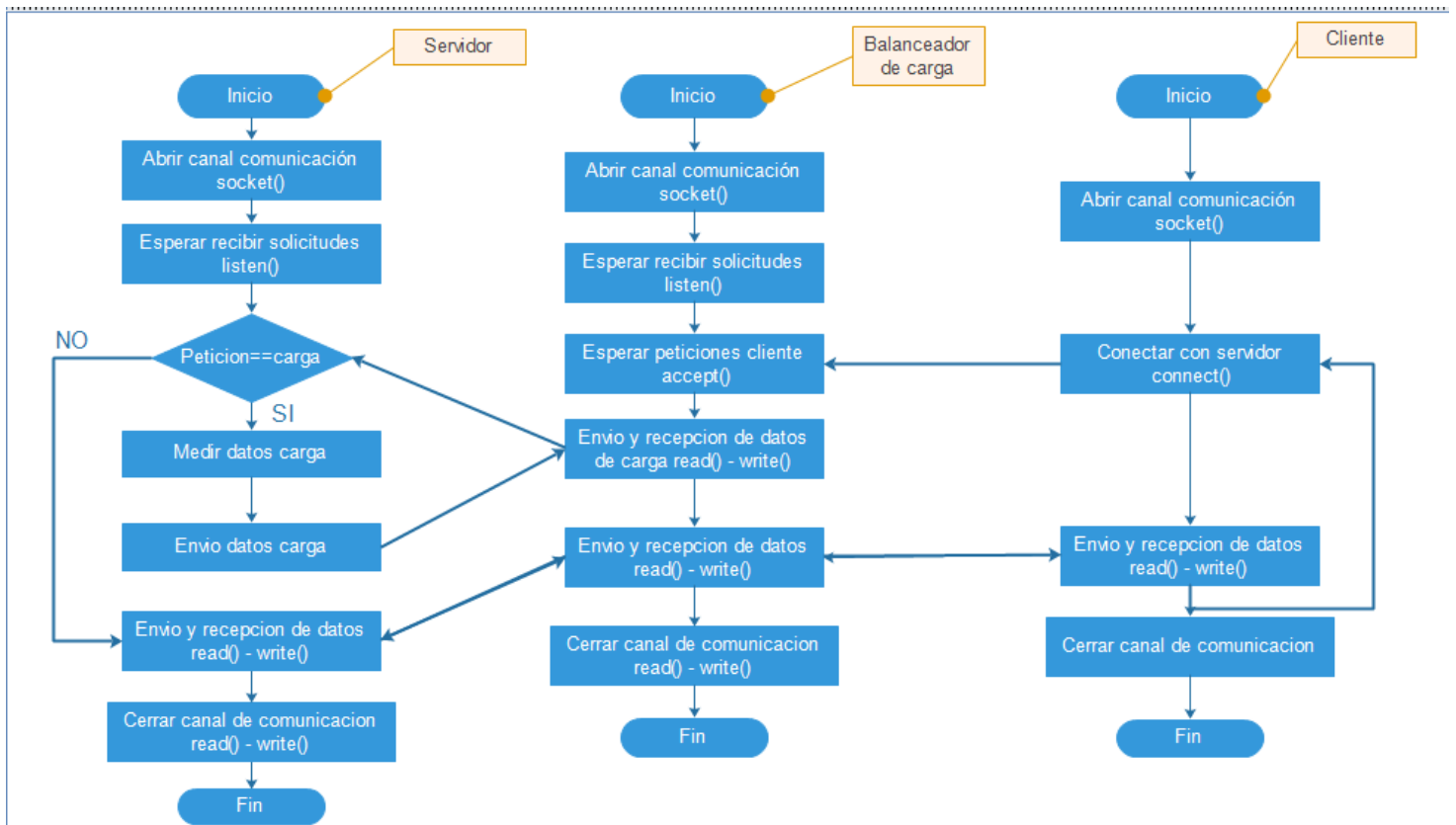
Se implementará también un balanceador de carga, el cual estará a encargado de recibir las peticiones de los clientes, luego enviará una petición a todos los servidores de banco solicitando su información de carga, de acuerdo a estos datos, se determinará el servidor a utilizarse, y se enviará la solicitud proveniente del cliente.

Cada petición de cliente a enviarse, se guardará en un solo String y tendrá el siguiente formato:

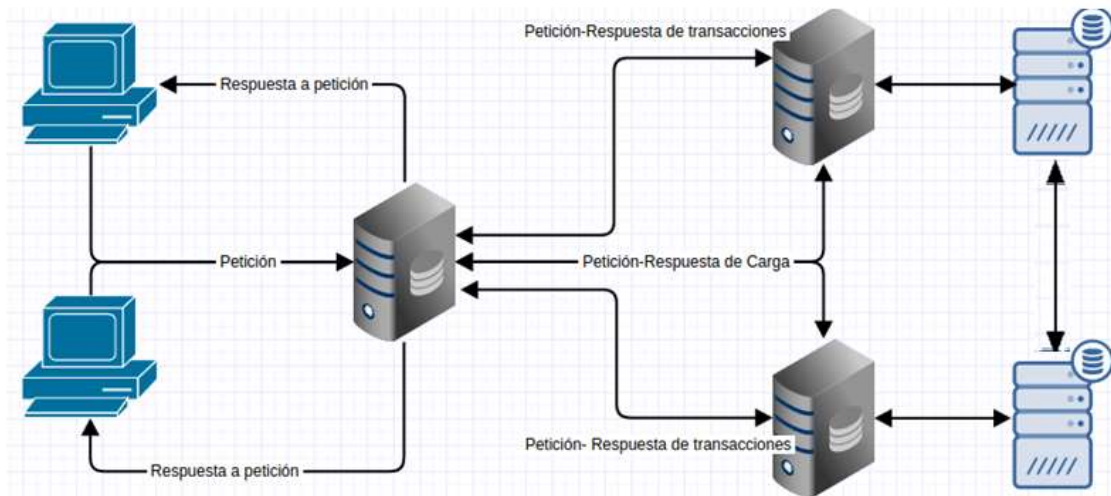
Cédula	Contraseña	Identificador	Tipo de transacción	Valor	Identificador
10 caracteres	Tamaño variable	#	0 ≈ Consulta 1 ≈ Depósito 2 ≈ Retiro	Tamaño variable	#

Para estresar a los servidores bancarios, desde consola se enviarán múltiples peticiones generadas aleatoriamente.

2. Diagrama de Flujo de arquitectura de alta disponibilidad



3. Diagrama de bloques



4. Seudocódigo

Cliente

```
Ciente.psc  x  Servidor.psc  Servidor_Balanceador.psc
1  Proceso Cliente
2      Crear socket
3      Abrir_canal_de_comunicacion
4      Conectar_con_servidor_Balanceador
5      Repetir
6          enviar datos
7          recibir datos
8      Hasta Que Cerrar_canal_de_comunicacion
9  FinProceso
```

Servidor: Balanceador de carga

```
Cliente.psc  Servidor.psc  Servidor_Balanceador.psc X
1
2  Proceso Servidor Balanceador
3      Crear socket
4      Abrir_canal_de_comunicacion
5      Escuchar_Conexion_Cliente
6
7      Repetir
8
9          Recibir_solicitud_cliente
10
11          Repetir
12              enviar_solicitud_datos_carga
13              recibir_datos
14          Hasta Que envíe_a_todos_los_servidores
15
16          Verificar_servidor_con_menor_carga
17          Conectar_con_servidor
18
19          Repetir
20              enviar_solicitud_cliente
21              recibir_datos
22          Hasta Que cerrar_conexion
23          enviar_datos_a_cliente
24
25      Hasta QueCerrar_canal_de_comunicacion
26
27  FinProceso
```

Servidor

```
Cliente.psc  Servidor.psc X  Servidor_Balanceador.psc
1  Proceso Servidor
2      Crear socket
3      Abrir_canal_de_comunicacion
4      Escuchar_Conexion_Balanceador
5
6      Si peticion=carga Entonces
7          enviar_datos_carga
8      Sino
9          Repetir
10              Recibir_solicitudes_balanceador
11              Procesar_solicitud_balanceador
12              enviar_datos
13          Hasta Que cerrar_conexion
14      Fin Si
15
16  FinProceso
```