

# Proyecto 2

---

Sistema de almacenamiento(clave=valor)  
distribuido

Profesor: Antonio Saad.

Moreno José, Peña Angel, Zaghab Milad  
UCAB GUAYANA | SISTEMAS DISTRIBUIDOS

Servidor 1	Servidor 2	Servidor 3
Pedir nombre de servidor		
RMI bind( )		
Crear itinerario (Ruta archivo conf)		
Mandar agente inicial →	Recibe( agente )	
	Comparte BD propia	
	Verificar itinerario del agente	
	Envía agente →	Recibe(agente)
		Comparte BD propia
		Verificar itinerario del agente
* Recibe agente inicial ( agente )		Manda agente → *
Almacena BD Actualizada		
Loop{		
LeerInput( )		
Obtiene Comando		
If(Comando==Get){		
Get_local( )		
If(not found)		
MandarAgenteGet( ) →	Recibe(agente)	
	Agente.Buscar( )	
	If(found)	
	Retorna padre	
	Else	
	MandarAgenteGet() →	Recibe(agente)
		Agente.Buscar( )
		If(found)
		Retorna padre
		Else
		MandarAgenteGet()* →
If (Comando== Set)		
Set_local()		
MandarAgenteSet( ) →	Recibe ( agente )	
	Set()	
	MandarAgenteSet( ) →	Recibe ( agente )
		Set()
		MandarAgenteSet()* →
If(Comando==Del)		
Del_local()		
MandarAgenteDel( ) →	Recibe ( agente )	
	Del( )	
	MandarAgenteDel( ) →	Recibe ( agente )
		Del( )
		MandarAgenteDel( )* →
}		
Recibe(agente)		

\*Instrucciones ejecutadas por el agente móvil

Este proyecto fue desarrollado bajo el lenguaje de programación orientado a objetos como lo es Java, como mecanismo persistente de almacenamiento de datos se usó un sistema de ficheros el cual está basado en archivo de texto plano (.txt), la librería usada para poder realizar codificación e invocación de los agentes móviles se usó JavaRMI. Como IDE se usó netbeans 8.1

La tabla anterior muestra el flujo lógico que corresponde al segundo proyecto que consiste en un sistema de almacenamiento de claveValor para sistemas distribuidos, utilizando agentes móviles.

Al iniciar el programa es solicitado el nombre o ID que va a tener ese servidor, luego es realizado el bind que se encuentra en la librería JavaRMI que esta registra al nodo/servidor en la lista de ese proceso, luego leo de un archivo de texto mi ConfigFile el cual puede poseer servidores con su ruta de acceso. Se manda un agente móvil a los demás servidores que fueron cargados del ConfigFile.

Una vez que ese agente móvil que salió llega al servidor que tiene en su itinerario, se le es compartida la BD que posee el servidor2 en ese momento, se realiza una verificación de itinerario entre el agente móvil y el servidor mediante un cruce de tablas para ver cuales otros nodos le hacen falta para completar la ruta en toda la red y el agente es enviado al siguiente servidor que tiene en su itinerario. Lo mismo va a ir ocurriendo con los demás servidores hasta lograr el recorrido completo de todos los servidores.

Ya terminado el recorrido el agente vuelve a el nodo padre (nodo inicial) donde procede a guardar la base de datos actualizada luego se entra en un Loop donde se van a solicitar un comando de acción que son (GET,SET,DEL). Luego de ser interpretado ese comando y validado se procede a realizar la acción correspondiente a ese comando y mandar el agente móvil a lo largo de la red para la actualización de la base de datos