**ESCUELA POLITÉCNICA NACIONAL**



**COMPUTACIÓN DISTRIBUIDA**

**PAOLA GUAMANÍ**

**CARLOS GUTIÉRREZ**

**PROYECTO SEGUNDO BIMESTRE**

**COMPARTICIÓN DE CARPETAS – P2P**

**2018**

# **REPORTE**

## **TEMA: APLICACIÓN COMPARTICIÓN DE CARPETAS USANDO UNA ARQUITECTURA P2P**

OBJETIVOS:

* Desarrollar una aplicación que utilice arquitectura P2P para compartir carpetas en Linux entre diferentes computadores de una misma red.

DESCRIPCIÓN DEL SISTEMA

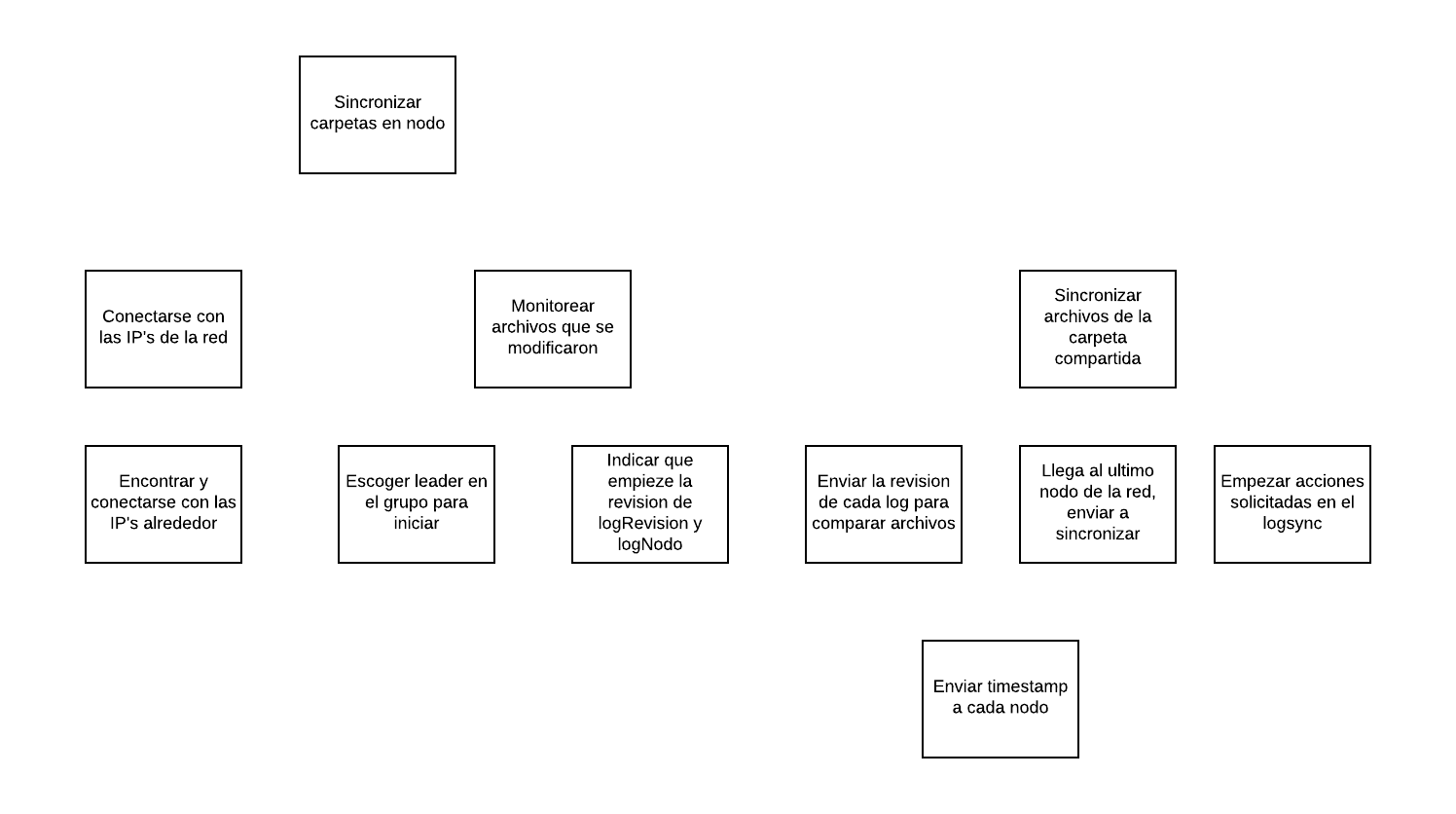
La aplicación se ejecuta con una arquitectura P2P, aquí se levantará en hilo a un cliente y un servidor en el mismo nodo, tanto cliente como el servidor se encarga de realizar las acciones tanto de revisión de los cambios en toda la red, escoger el líder, sincronizar y actualizar los archivos de la carpeta compartida.

Se realizará un monitoreo constante donde cada minuto se revisan la carpeta compartida para controlar los archivos.

### ESPECIFICACIONES DE REQUISITOS

* El sistema debe poder sincronizar carpetas compartidas alojadas en cada computador
* El sistema debe contener mínimo 3 computadores para realizar una sincronización completa
* El sistema debe ser descentralizado, no debe usar una arquitectura C/S
* El sistema debe sincronizar sin la necesidad de enviar comandos en consola, debe ser de forma automática dicha sincronización.
* El sistema debe monitorear los archivos de las carpetas compartidas, es decir, actualizar el estado de los archivos de la carpeta compartida.
* El sistema, tal como una aplicación, debe correr en el sistema operativo Linux como un servicio.

### DISEÑO ESTRUCTURADO



MÓDULOS

Para iniciar se requiere sincronizar las carpetas de un computador con otro

Módulo Principal

* Levantar el nodo, tanto con su parte cliente como servidor y ponerse a escucha

Módulo direccionamiento

* Buscar las posibles direcciones IP’s en la red
* Realizar intentos de conexión con cada nodo alrededor, a su parte server

Módulo Monitoreo

* Escoger el leader del grupo conectado
* Realizar monitoreo de las carpetas constantemente en un hilo cada 180 segundos
* Implementar un envío de conteo propio al resto de computadores en la red para que inicie la revisión del logRevision.txt, este pasará por todos los nodos de la red y así sabrán que actualizar
* Comparar con su logNodo.txt los cambios dados

Módulo Sincronizar

* Indicar que el log de revisión ha pasado por todos los nodos y que debe iniciar la sincronización
* Enviar a cada nodo el conteo, para que el último se sincronice
* Pedirle al nodo anterior que le envié los archivos faltantes para que su carpeta se actualice

### DISEÑO ARQUITECTÓNICO

* Partes: Cliente y Servidor en el mismo nodoP2P
* Comunicación: con la misma conexión que se levantará el Cliente y el Servidor
* Carga: cada nodoP2P realiza sus acciones, no hay sobrecarga en un servidor central.
* Máquinas: 3 máquinas virtuales con S.O. Linux en VMWare
* Subsistemas:
  + Principal: levantar cliente y servidor en el nodoP2P
  + Direccionamiento: buscar y conectarse a las IP’S de la red
  + Monitoreo: Constantemente buscar cambios en las carpetas y registrar en el log local del nodo.
  + Sincronización: ubicar quien necesita el cambio y enviarle el archivo que necesita en caso de serlo.

### DISEÑO DE COMPONENTES

* Capa de direccionamiento: direccionamiento de Protocolo Internet (IP), las direcciones IP tienen una estructura específica y no se asignan al azar
* Comunicación: mediante sockets
* Sincronización: mediante el monitoreo, a través de comandos “cp”, “rm” y “ls”
* Actualización: utiliza un monitoreo con archivos, a través de comandos “cp”, “rm” y “ls”
* logNodo.txt: contiene el registro de los datos de los archivos de cada nodo, como nombre, operación(add, delete), timestamp, ip de donde proviene
* logRevision.txt: contiene registros iguales, pero son los que deben contener cada nodo de la red para que se sincronicen posteriormente
* logSync.txt: Indica las operaciones que deben realizarse para que el nodo se actualice

### IMPLEMENTACIÓN

Herramientas

* Máquinas virtuales con sistema operativo Linux, sin costo, el software usado es VMWare y los distro’s son descargados de la página oficial de manera gratuita.
* Github: se alojarán las distintas versiones del código
* Código desarrollado en lenguaje de programación Python
* IDE Pycharm, free

### PRUEBAS

* En el ambiente de las máquinas virtuales y con cambios constantes a github, además máquinas físicas en Windows para el desarrollo del código inicial.