**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**Sistemas operativos**

2014

**DOCUMENTACION SNAKEBOX**

**Gustavo Totoy, Frank Hermida**

INDICE

# Introducción

SnakeBox es una aplicación cliente-servidor que simula el almacenamiento de archivos de texto en la nube mediante el uso de mecanismos de sincronización, el cual le permite a un usuario local almacenar archivos en su directorio local y estos son subidos automáticamente al servidor.

En este documento se describen los mecanismos de sincronización usados para resolver la subida de archivos, la comunicación entre el cliente y el servidor y la resolución de problemas como la colocación de archivos en los directorios correspondientes para cada usuario, entre otros.

# Diagrama

![](data:None;base64,)![](data:None;base64,)![](data:None;base64,)![](data:None;base64,)![](data:None;base64,)

o o o o o

![](data:None;base64,)![](data:None;base64,)

Pull()

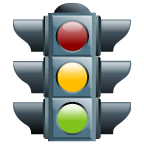
Actualiza los archivos del cliente usando los archivos del servidor



Push()

Actualiza los archivos del servidor usando los archivos del cliente



****

# Diseño Propuesto

## Manejo de archivos

Para el mecanismo de comunicación cliente-servidor se usó **RMI (**java remote method invocation sus siglas en inglés.**)** [1]El cual nos ayuda a invocación de métodos de manera remota, el cual es un mecanismo simple para la comunicación de servidores basados en **JAVA.**

La librería **RMIIO** [2] se encarga de transferir grandes cantidades de datos entre un cliente y un servidor mediante el uso del mecanismo **RMI.**

La librería **commons IO** ayuda a las funcionalidades de entrada y salida de archivos [3].

## Sincronización de archivos

Para la sincronización de los archivos, se usó **bloques Syncronized de java**, el cual sirve para especificar en ciertas partes del código están sincronizadas [4] y las operaciones de semáforos [5].

### Método push()

El método **push()** de la clase **SnakeClient**, sincroniza los archivos del servidor usando los archivos alojados en el cliente.

### Método pull()

El método **pull()** de la clase **SnakeClient**, sincroniza los archivos del cliente usando los archivos alojados en el servidor.

# Bibliografía

|  |  |
| --- | --- |
| [1] | Oracle, "Remote Method Invocation Home," Oracle Corporation, 2014. [Online]. Available: http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html. [Accessed 06 Julio 2014]. |
| [2] | openhms, «Souceforge,» 23 Septiembre 2013. [En línea]. Available: http://openhms.sourceforge.net/rmiio/. |
| [3] | Apache, «Commons IO,» Apache Org, 17 Febrero 2014. [En línea]. Available: http://commons.apache.org/proper/commons-io/. |
| [4] | sc, «Sincronizacion,» [En línea]. Available: http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/threads/sincronizacion.htm. |
| [5] | Oracle, «Class Semaphore,» Oracle corp., [En línea]. Available: http://docs.oracle.com/javase/7/docs/api/java/util/concurrent/Semaphore.html#acquire(). |
| [6] | J. Jenko, «Java Synchronized Blocks,» [En línea]. Available: http://tutorials.jenkov.com/java-concurrency/synchronized.html. |