

# Programación en Red: Sockets

ST0255 - Telemática

Profesor: Juan C. Montoya

5 de octubre de 2020

## 1. Objetivo

Desarrollar habilidades en los aspectos relacionados con la programación de aplicaciones en red.

## 2. Introducción

La forma estándar para desarrollar programas que se comunican sobre una red de datos es a través de la utilización de los sockets.

Este proyecto pretende ofrecerle a usted la experiencia de programación básica con sockets. Específicamente, se pretende que usted escriba un par de programas (cliente y servidor) que se comuniquen entre sí a través de un protocolo que usted implementara utilizando la API de sockets.

De esta manera, se requiere que usted desarrolle su propio protocolo con el fin de aplicar y profundizar todo lo concerniente con el diseño e implementación de éstos.

## 3. Problemática

La empresa XYZ S.A requiere implementar un servicio de almacenamiento de datos que permita poder alojar diferentes tipos de archivos como: documentos, imágenes, archivos binarios, etc. De esta forma, cada archivo se considera un objeto el cual puede ser almacenado en un contenedor que se denomina bucket.

El servicio del almacenamiento puede ser accedido de manera remota y permite a los clientes consultar el conjunto de buckets que se tienen definidos en la organización. Es en cada bucket, donde los clientes deben almacenar los diferentes objetos (archivos).

Considerando lo anterior, se requiere que usted desarrolle un protocolo de capa de aplicación que le permita a los usuarios acceder al servicio y poder realizar diferentes operaciones en los buckets como la carga, recuperación, listado borrado de archivos, entre otras mas.

## 4. Requerimientos

Se requiere que usted diseñe e implemente un protocolo que describa la comunicación entre una aplicación cliente y una servidor. En este sentido, se espera que la aplicación cliente debe enviar una petición al servidor para ser procesada. Una vez el proceso servidor reciba la petición, esta debe ser procesada y un mensaje de respuesta debe ser enviado al cliente.

A continuación, se plantean el conjunto de requerimientos mínimos que se deben satisfacer:

- La implementación de su protocolo debe correr o ejecutarse sobre sockets directamente. En este sentido, no puede hacer uso de algún protocolo existente ( p.ej., HTTP, FTP o cualquier otro protocolo de capa de aplicación).
- La aplicación cliente debe operar de manera interactiva. Se espera que pueda implementar las siguientes operaciones:
  - Crear y eliminar un bucket. Cada bucket debe tener un identificador único.
  - Listar los diferentes buckets existentes en el directorio de trabajo.
  - Realizar la carga de un archivo desde el cliente hasta el bucket definido. Cada objeto alojado en un bucket debe tener un identificador único.
  - Poder listar los diferentes archivos que hay en cada bucket.
  - Descargar un archivo del servidor al cliente.
  - Eliminar un archivo del bucket.
- Para el caso de la aplicación servidor debe considerar:
  - Uno de los parámetros que debe recibir la aplicación servidor al momento de ejecutarse es el path donde se crean o están alojados los buckets.
  - Debe tener la capacidad de manejar múltiples clientes de manera concurrente. Al respecto, cada cliente debe poner conectarse y desconectarse del servidor en el momento en que lo considere. El servidor debe seguir operando de manera transparente.
  - El servidor debe estar en la capacidad de poder recibir comandos del cliente así se encuentre enviando un archivo o respuesta a éste.
  - Recibir y enviar los diferentes archivos por parte de los clientes conectados al servidor.

## 5. Diseño del Protocolo

Para efectos del diseño y especificación del protocolo a implementar, se requiere que usted defina los siguientes elementos:

- Especificación del servicio: Este apartado debe contener una descripción del servicio a implementar.
- Vocabulario de mensajes: En este apartado se requiere que usted defina el conjunto de mensajes que implementa su protocolo. Igualmente, para cada mensaje se hace necesario definir la sintaxis (el formato y/o representación del mismo) así como el significado (semántica) de cada uno.
- Regla de procedimientos: Hace referencia a todos los aspectos relacionados con el estado de las entidades, el envío y recepción de mensajes por parte de las entidades, el procesamiento de los mensajes, manejo de errores, etc.

Para la especificación formal de su protocolo, apóyese en UML para la formalización de su especificación. Una vez tenga esta especificación puede proceder a implementarlo en un lenguaje de programación.

En este sentido, a continuación se listan un conjunto de preguntas que le pueden servir en el desarrollo de su especificación:

- ¿Cuáles son los mensajes que van a intercambiar las entidades del protocolo?
- ¿Cuál es la sintaxis así como el significado de cada mensaje?
- ¿Cómo por el envío o recepción de un mensaje cambia el estado de la entidad cliente/servidor?
- ¿Requiere el servidor mantener algún tipo de información de estado en relación con alguna petición del cliente? ¿Son las peticiones del cliente independientes entre si o tienen alguna relación?

## 6. Implementación

Para efectos de la implementación del protocolo, usted puede realizar la implementación de este utilizando los lenguajes de programación en Python 3.

### 6.1. Aspectos a considerar

- Tenga presente que menos líneas de código puede estar correlacionado con mejores resultados. Todo esto ya que puede indicar/implicar un protocolo más simple y bien estructurado.
- Realice pruebas de su código de manera incremental y con cambios pequeños. Recuerde que es más fácil encontrar errores cuando ha hecho pequeñas modificaciones.

## **7. Entrega**

El trabajo se debe realizar en parejas. No se aceptan entregas individuales. La fecha de entrega es Domingo 25 de Octubre de 2020 hasta las 23:59. Posteriormente se le indicará el mecanismo de entrega.