

# **Taller Telemática**

## **Programación en Red: Sockets**

### ***Integrantes:***

***Jonatan David Acevedo Lopez***

***Santiago Moreno Rave***

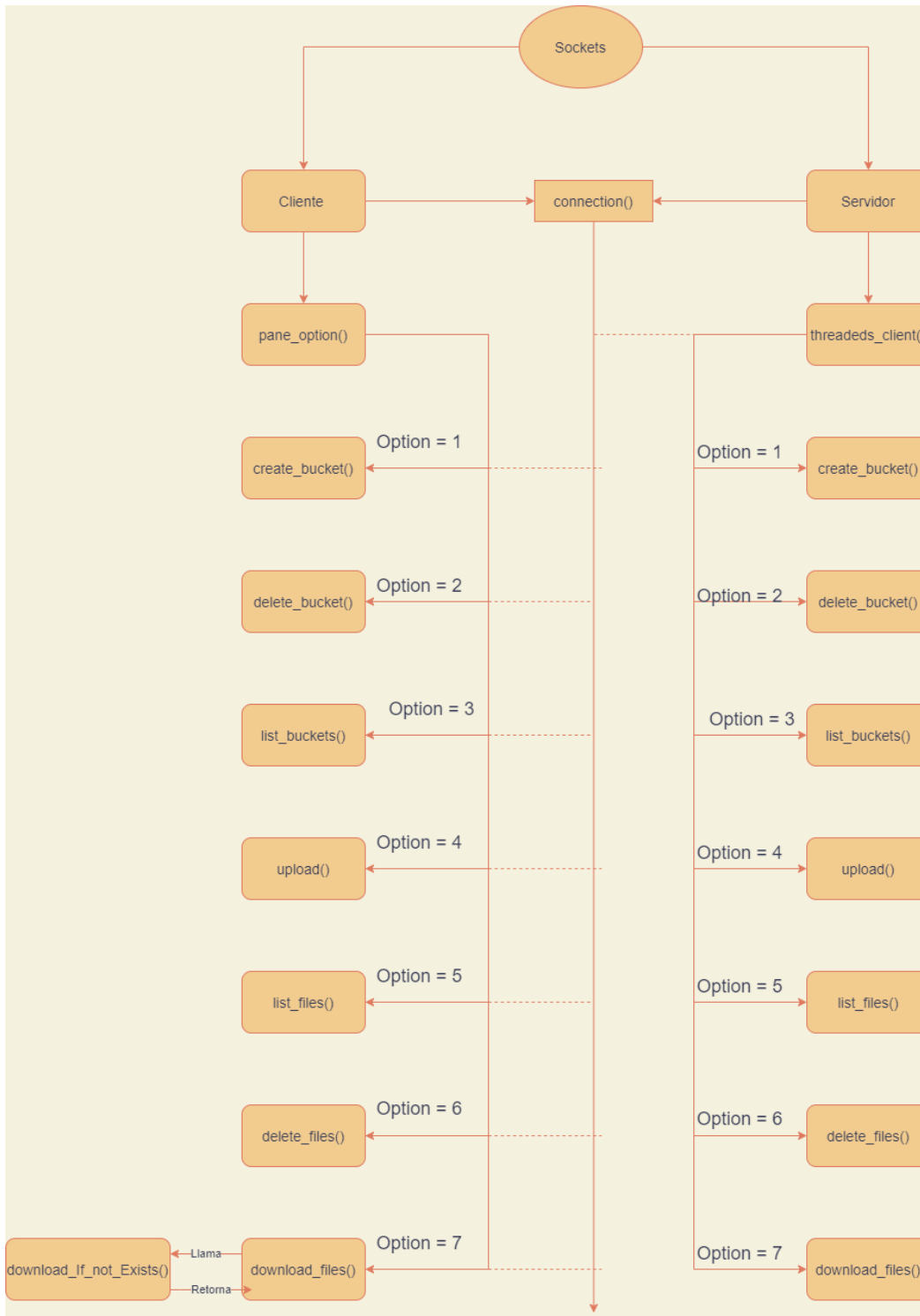
### **introducción**

Este documento mostrara como es el funcionamiento de un servicio de envíos de datos entre un cliente y servidor usando sockets, descripción del vocabulario empleado y una demostración grafica del procesamiento del Código.

### **Objetivos**

- Mostrar el funcionamiento de los sockets
- Implementar nuestro propio vocabulario para el desarrollo del servicio
- Ampliar nuestro conocimientos sobre programación con sockets

## Funcionamiento del servicio (Grafico)



## Diseño de protocolo

### *Especificación del servicio*

Este servicio es un transporte de datos que se encarga de transferir archivos de un servidor a un cliente, entre las principales funcionalidades del sistema están:

- Creación de Buckets
- eliminación de Buckets
- Mostrar todos los Buckets existentes
- Subir un archivo del cliente al servidor
- Descargar un archivo del servidor al cliente
- eliminación de archivos en Buckets
- Mostrar todos los archivos almacenados en los Buckets

### *Vocabulario de mensajes*

Nuestro aplicativo una vez se conecte el cliente, le solicita que ingrese un número, si ingresa el **0** se finalizara el proceso, pero si ingresa **1** solicitara un numero para realizar la operación a la que pertenece este número, si ingresa un numero no valido volverá a preguntar hasta que ingrese un numero valido.

### *Instrucciones*

```
{\n  Table(['Option', 'Name Option'], )\n  [1, "Create bucket"])\n  [2, "Delete bucket"])\n  [3, "List bucket"])\n  [4, "Upload bucket"])\n  [5, "List file"])\n  [6, "Delete file"])\n  [7, "Download bucket"])
```

- 1
  - Ahora debe colocar el nombre que quiere para el Bucket, no es válido que el nombre sea un espacio o este vacío.
- 2
  - Ahora debe colocar el nombre del Bucket que quiere eliminar. No es válido que el nombre sea un espacio o este vacío.
- 3
  - Solo debe esperar a que el aplicativo le liste todos los Buckets
- 4
  - Ahora debe ingresar el nombre del Bucket, una vez ingrese el nombre del Bucket debe ingresar el nombre del archivo (png, pdf, txt) y por último la dirección donde se encuentra guardado el archivo que se desea subir. No es válido que ninguno de los valores que se piden sean un espacio o estén vacíos. Nota: La ruta o path debe tener su estructura con / más no con \ , por ejemplo:
  - Manera incorrecta: C:\User\Desktop
- 5
  - Ahora debes ingresar el nombre del Bucket del cual quieres ver los archivos, una vez hecho esto se listarán los archivos. No es válido que el nombre sea un espacio o este vacío.
- 6
  - Ahora debes ingresar el nombre del Bucket del cual quieres borrar el archivo, después ingresar el nombre del archivo que deseas eliminar. No es

válido que ninguno de los valores que se piden sean un espacio o estén vacíos.

- 7
  - Ahora debes ingresar el nombre del Bucket que deseas descargar, después el nombre del archivo que deseas descargar. No es válido que ninguno de los valores que se piden sean un espacio o estén vacíos.

### ***Regla de procedimientos***

- *Manejo de errores*
  - En el momento que se genera un error ya sea porque se copió mal una instrucción o un archivo no valido, arroja un mensaje al cliente para que corrija su error y vuelva a enviar la instrucción o archivo de forma correcta.
- *Procesamiento de mensajes*
  - Cuando es un mensaje de menos o igual a 1024 bytes se envía el mensaje, de no ser así se detecta el tamaño del archivo en bytes, el proceso de envío o de descarga se hace byte por byte.
- *Envío y recepción de mensajes por parte de la entidad*
  - Hay variables específicas que se encargan de recibir los bytes que le llegan del servidor al cliente y viceversa, un ejemplo claro de la transferencia de datos es el método upload en el cliente este tiene la variable content la cual almacena la información del archivo que se va a subir al servidor, en la parte del upload del servidor tiene la variable data\_recive la cual se encargar de recibir lo que le

llega del cliente y escribir byte por byte el archivo nuevo en el servidor.

## **Conclusión**

- Con la presente práctica evidenciamos la teoría vista en clase del manejo de datos, a través de la comunicación por socket que nos permite la conexión por puertos e ips para establecerse, recibir y enviar datos
- Se evidencian además el manejo de archivos y bytes sus encabezados y como estos nos sirven para facilitar el manejo y cálculo para enviar y recibir de manera adecuada los bytes de información
- El manejo de hilos es una parte crucial para el manejo de clientes - servidor, las librerías nos facilitarán esto y mucho más