

Clientes y Servicios

Javier Esteban López Peña

September 8, 2021

1 Introducción

Para este reto usamos conceptos básicos de redes. Estos conceptos a tener en cuenta son el uso de los protocolos generales de internet que son TCP y UDP, estos utilizan puertos para enviar los datos que llegan a las aplicaciones correctas. Recordemos que los puertos se representan con un entero de 16 bits en un rango de 0 a 65535. Se implemento una arquitectura orientada a los servicios, ya que, permite brindar ciertos servicios a los clientes, por medio de peticiones a un servidor de manera simple y efectiva. Por otro lado hace que su implementación sea extensible y facil

2 Aplicativo

2.1 Funcionamiento de la URL's

El programa desarrollado soporta multiples solicitudes seguidas, realizandolo por localización de recursos. Para esto se usa la URL, comunmente tienen una forma general que es

[protocolo]://[servidor]:[puerto]/[direccion del recurso en el servidor]

Tal como se puede notar en el desarrollo todas las URL lanzan excepciones por lo que es necesario colocarlos dentro de un bloque try-catch

2.2 Sockets

Los sockets son puntos finales de comunicacion entre dos programas que se encuentran en la red. Por lo que cada socket debe estar vinculado a un puerto especifico, vincularlo a un servidor y este queda esperando solicitudes de externos, por esto funcionan para implementar protocolos de comunicacion cliente-servidor (Como el caso del reto desarrollado). En el aplicativo para obtener una conexión se usa el código

serverSocket = new ServerSocket(PORT)

Donde en PORT iría el puerto a asignar, (En el caso de el aplicativo es 4567). Con esta conexión ya se podría obtener las entradas y salidas utilizando el **BufferedReader** o en su caso el **BufferedImage** para imágenes. Se debe tener en cuenta que los sockets deben cerrarse para evitar errores futuros [\[Jav\]](#)

2.3 MIME

Como se ha mencionado el aplicativo tiene un servidor que debe retornar todos los archivos solicitados, incluyendo páginas html e imágenes. Para esto se necesita usar MIME, este es un estándar de Internet que extiende el formato de los mensajes de correo electrónico para admitir texto en conjuntos de caracteres que no sean ASCII, así como archivos adjuntos de audio, video, imágenes y programas de aplicación. Los tipos MIME utilizados fueron [\[Doc\]](#)

2.4 Conclusion

Después de aclarados los conceptos que son necesarios para el desarrollo de este reto, podemos además de realizarlo, entender su funcionamiento. Pues para probarlo se hizo una página web en javascript, usando imágenes para también poner a prueba el cargado de este tipo de archivos.

| Extensión | Tipo MIME |
|-----------|-----------------|
| .js | text/javascript |
| .html | text/html |
| .css | text/css |
| .jpeg | text/jpeg |
| .jpg | text/jpg |
| .png | text/png |

Table 1: Tipos MIME importantes para la Web.

References

[Doc] MDN Web Docs. Common mime types.

[Jav] Java. Trail: Custom networking.