**ENTENDIENDO EL PROYECTO N° 1 DE PROGRAMACIÓN EN RED.**

**Fecha de publicación:** marzo 9 de 2024

**Última actualización:** marzo 9 de 2024.

**Fecha de Entrega:** abril 9 de 2024.

1. **TEORÍAS.**

**PROXY INVERSO:** se entiende como aquel que recibe (intercepta) cada una de las peticiones del cliente y las envía al servidor con la capacidad de procesar las peticiones para finalmente enviar la respuesta al cliente.

**BALANCEADOR DE CARGAS:** es la entidad encargada de distribuir las peticiones entrantes por parte de los clientes hacia un conjunto de servidores. Para cada petición, posteriormente, debe retornar la respuesta al cliente.

**SOCKETS:** Tradicionalmente, a los desarrolladores de aplicaciones, las APIs de desarrollo de los diferentes lenguajes de programación, les realizan abstracciones que les ocultan los detalles de implementación de muchos aspectos, y el proceso de transmisión de datos no es la excepción.

Un socket es una abstracción a través de la cual las aplicaciones pueden recibir y enviar datos. Al respecto, el socket se puede entender como el mecanismo que utilizan las aplicaciones para “conectarse” a la red, específicamente con la arquitectura de red o “stack” de protocolos y de esta forma comunicarse con otras aplicaciones que también se encuentran “conectadas” a la red.

**TIPOS DE SOCKETS:** Existen diferentes tipos de sockets. Para el proyecto vamos a utilizar API SOCKETS BERKELEY, específicamente los sockets tipo “Stream (SOCK\_STREAM)” o “Datagram (SOCK\_DGRAM)”.

Dentro del conjunto de razones por las cuales se emplea un proxy a nivel de http, se resaltan aspectos de desempeño, filtrado y transformación de contenido, entre otros aspectos.

1. **REFERENCIAS**

<https://beej.us/guide/bgnet/>

<https://beej.us/guide/bgc/>

<https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/>

<https://datatracker.ietf.org/doc/html/rfc7230>