[[1]](#footnote-1)

*Proyecto Primer Corte AREM (Servidor Web) – Creación de un Servidor Web* (Febrero 22)

Andres G.Florez, Escuela Colombiana de Ingeniería*, Servidor Web*,

Elsiguiente articulo tiene como objetivo explicar la estructura de un servidor web que responde a peticiones de páginas HTML, imágenes png y jpg además de código JavaScript y css. Aparte de esto se busca explicar la utilización de la arquitectura en función de las anotaciones y POJOS el cual permite la reflexión de las clases que permiten colocar parámetros en la URL,este servidor funciona con peticiones no concurrentes.

# INTRODUCCIÓN

ESTE documento es una referencia para poder entender la implementación realizada al momento de desarrollar el proyecto, por lo tanto, es una guía para la comprensión de su propio diseño, implementación, observaciones y ejecución.

Se tomarán en cuenta los requerimientos planteados al momento de la creación del enunciado del proyecto y se definirá todos sus componentes detalladamente, entre los cuales se pueden observar a grandes rasgos su arquitectura, diseño, implementación, pruebas, documentación y variables definidas para el presente proyecto.

# Arquitectura

Se definieron las 3 abstracciones fundamentales para el presente proyecto entre Memoria, Interpretes y Enlaces de Comunicación con base a los conceptos aprendidos en clase, como se podrá observar en el *grafico 1* donde podemos encontrar dichas abstracciones distribuidas de la siguiente manera:

## Memoria

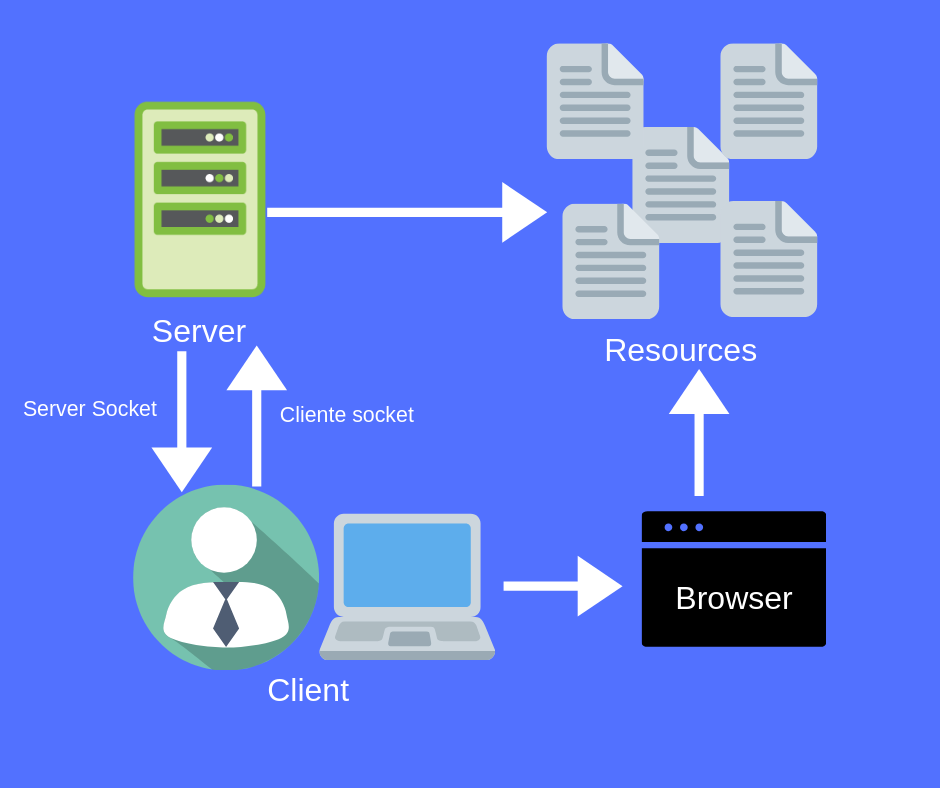
Como abstracciones de memoria podemos referenciar a los archivos del projecto, los .class y .java donde se encuentra implementado nuestro proyecto, los archivos tanto .HTML, como .png,.jpg, javascript,css y todos los recursos del proyecto.

## Interpretes

Como abstracciones de intérpretes encontramos el Servidor, junto con la Máquina Virtual de Java, así como también el Browser que interpreta los html’s y las peticiones del servidor.

## Enlaces de Comunicacion

C)**Como enlaces de comunicación**, se puede definir principalmente los Sockets que comunican al Cliente con el Servidor, También el protocolo http que comunica al servidor con internet, y el protocolo HTML que comunica a las páginas web del servidor con el Brower.

 Gráfico 1.

# Diseño

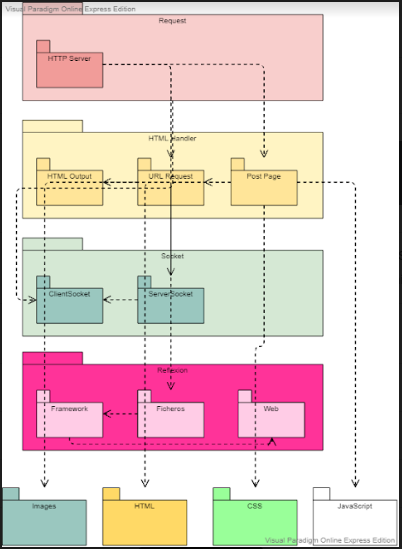
Las capas de diseño implementadas se definieron con base a los requerimientos y necesidades de la implementación, donde cada capa se encuentra aislada de la otra, en busca de generar condiciones de estabilidad a corto plazo y probablemente extensibilidad a futuro (ver gráfico 2). Por este motivo se definieron las siguientes 5 capas:

## Request

Constituye la capa principal del proyecto, está conformada por el main principal del proyecto, ubicado en la única clase del paquete “HTTP Server”, desde aquí se corre el servidor llamando a ambos sockets y se empiezan a pedir solicitudes del Cliente hacia el servidor.

## Socket

Es la capa del proyecto que se encarga meramente de crear e inicializar los Sockets, tanto el del Cliente, como el del servidor; cada uno en su respectiva clase, está conformada por el main principal del proyecto, ubicado en la única clase del paquete “HTTP Server”, desde aquí se corre el servidor llamando a ambos sockets y se empiezan a pedir solicitudes de el Cliente hacia el servidor.



## HTMLHandler

Esta capa es la que genera la visualización de todas las dependencias .HTML del proyecto, es acá donde tanto se leen, como se interpretan, visualizan y publican, los resultados de cualquier solicitud en el servidor. Por lo tanto, esta capa se encuentra encargada de leer los correspondientes archivos .html de las paginas de pueden corresponder a una solicitud para visualizarlos luego en el servidor.

## WebPages

En esta capa del proyecto, podemos hacer una referencia a la abstracción de memoria que tenemos con los archivos que vamos a visualizar dada una solicitud cualquiera. Es por esta razón que esta capa se encuentra separada de las demás debido a que constituye los archivos .HTML diferentes de la implementación .java del proyecto, donde generamos las distintas ramas de páginas de nuestro servidor web.

## Images

Al igual que la capa anterior, se encuentra separada de la demás capa lógica y solamente se encarga de interactuar con los archivos que componen las paginas HTML. Esta capa contiene a las imágenes que se usaron para ser visualizadas en el servidor.

***F****.Reflexion*

Esta capa permite implementar un framework a partir de anotaciones y POJO’s, se basa en la idea en crear un Pojo asignarle una anotación y luego hacer la lectura de esa clase que tenga esos métodos con anotaciones y luego llamar al método directamente y adornarlo con HTML para poder mostrarlo.

G.Css

Al igual que la capa anterior, se encuentra separada de la demás capa lógica y solamente se encarga de interactuar con los archivos que componen las paginas HTML. Esta capa contiene los códigos css que se usaron para ser visualizadas en el servidor.

H.JavaScript

Al igual que la capa anterior, se encuentra separada de la demás capa lógica y solamente se encarga de interactuar con los archivos que componen las paginas HTML. Esta capa contiene los códigos css que se usaron para ser visualizadas en el servidor.

# Estructura del Servidor

La estructura del servidor web se basa en una página principal desde la cual se puede tener acceso a las demás raíces del proyecto (“/index.html”), teniendo en cuenta. De aquí se generan todos los vínculos a las demás dependencias del servidor, Incluyendo tanto las páginas HTML, como las imágenes usadas para visualizarlas además del código css y JavaScript. Las paginas HTML se componen además del índex,

En donde se visualiza todo el contenido de las páginas y el salto de una pagina a otra mediante enlaces, además de la explicación de como usar las páginas que tienen aplicación con reflexión.



# Conclusion

## Como conclusiones, podemos establecer la gran importancia que tienen los sockets en la implementación de servidores, que, aunque si bien se sabe son de bajo nivel, ayudan notablemente en la comunicación Cliente-Servidor.

También se puede concluir la importancia de separar por capas igualmente dicha implementación de sockets, debido a la gran complejidad que llevaría comunicarse entre los mismos, si todo se realizara en una misma clase, o incluso en un mismo paquete.

Y para finalizar el uso de anotaciones y los POJOS ayuda radicalmente en la escritura de código y en los despliegues de las aplicaciones.

**Andres Florez Perez** nació en Bogotá, Cundinamarca, Colombia en 1997. Graduado de el Colegio Salesiano Juan del Rizzo inició sus estudios de Ingeniería de Sistemas en 2014-2.

Actualmente estudia en la ECI, y se ha desempeñado como monitor del curso de programación aritmética modular, actualmente monitor de la sala de sistemas es maratonista de programación, tiene un certificado de CISCO en Fundamental Networking, además de haberse desempeñado como trabajador como proyectos pequeños en empresas sobre desarrollo web.

1. [↑](#footnote-ref-1)