

# AREP - Laboratorio 03

Eduardo Ocampo

Escuela Colombiana de Ingeniería  
eduardo.ocampo@mail.escuelaing.edu.co  
Bogotá D.C.

## 1. Resumen

Este programa fue creado utilizando java e IntelliJ, y esta hecho siguiendo los siguientes requisitos:

EJERCICIO 1:

Escriba un programa en el cual usted cree un objeto URL e imprima en pantalla cada uno de los componentes de una URL. Es decir, debe usar los siguientes métodos: getProtocol, getAuthority, getHost, getPort, getPath, getQuery, getFile, getRef. Asegúrese que ninguno imprima una cadena vacía, esto implica que la URL que use para su objeto debe contener datos suficientes.

EJERCICIO 2:

Escriba una aplicación browser que dada una URL lea datos de esa dirección y que los almacene en un archivo con el nombre resultado.html. Intente ver este archivo en el navegador. Su implementación debe ser un programa que reciba el parámetro de URL por medio de la línea de comandos.

Ejercicio 3:

Usando sockets escriba un servidor que reciba un número y responda el cuadrado de este número. Escriba un cliente para probarlo y envíele una secuencia de 20 números.

Reto 1:

Escriba un servidor web que soporte múltiples solicitudes seguidas (no concurrentes). El servidor debe retornar todos los archivos solicitados, incluyendo páginas html e imágenes. Construya un sitio web con javascript para probar su servidor. Despliegue su solución en Heroku.

## 2. Introducción

Este programa crea un objeto URL e imprime en pantalla cada uno de los componentes de una URL, también contiene una aplicación browser que dada una URL lee datos de la dirección y los almacene en un archivo con el nombre resultado.html. Además tiene un servidor que recibe un número y responde el cuadrado de este número, por último tiene un servidor web que soporta múltiples solicitudes seguidas (no concurrentes). El servidor retorna archivos.

## 3. Arquitectura

Este programa maneja un cliente y un servidor, el cliente hace solicitudes al servidor, que es quien tiene la información, este así mismo le retorna el recurso que el cliente le está pidiendo. Los datos se pasan mediante sockets al momento de hacer la solicitud desde un cliente.