

Taller de Clientes y servicios

Por:

Andres Orlando Sotelo Fajardo



AREP 2021-1

1. Descripción

Este taller presenta diferentes retos que nos ayudaron a explorar los conceptos de esquemas de nombres y de clientes y servicios. Adicionalmente, el taller nos ayudo a explorar la arquitectura de las aplicaciones distribuidas sobre internet.

2. Requisitos funcionales

- 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. NO use frameworks web como Spark o Spring use solo Java y las librerías para manejo de la red.

- RETO 2 (AVANZADO)

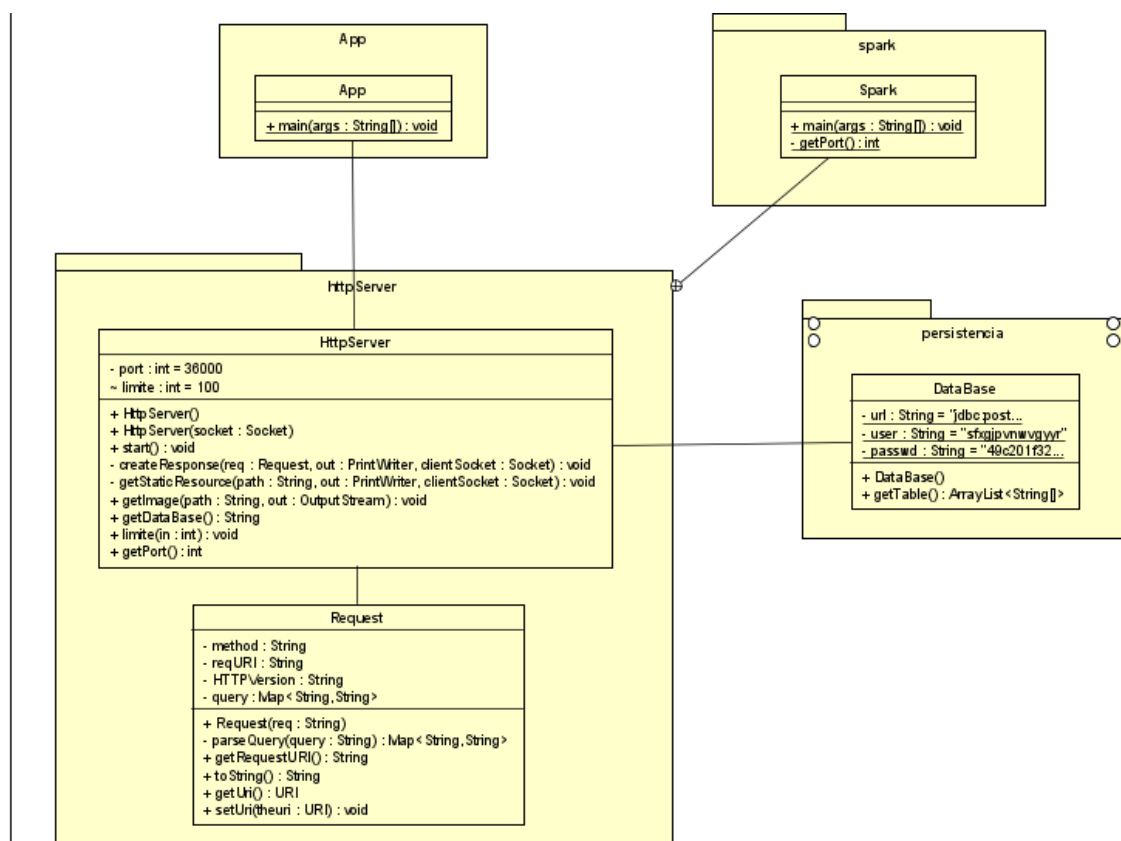
Usando su servidor y java (NO use frameworks web como Spark o Spring). Escriba un framework similar a Spark que le permita publicar servicios web "get" con funciones lambda y le permita acceder a recursos estáticos como páginas, javascripts, imágenes, y CSSs. Cree una aplicación que conecte con

una base de datos desde el servidor para probar su solución. Despliegue su solución en Heroku.

ENTREGABLES

- El proyecto con los retos implementados y funcionando desplegado en Heroku
- Repositorio en GITHUB con los más altos estándares de calidad
- Artículo describiendo la arquitectura de su prototipo y la evaluación
- Evaluación al proyecto de otro de sus compañeros.

3. Diseño y uso del programa



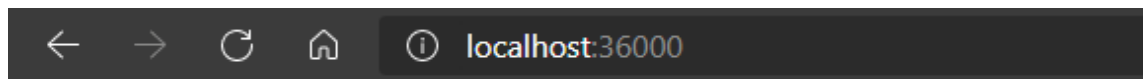
La clase app es la encargada, por medio de httpserver, de crear el servicio web que inicialmente muestra una pagina principal de bienvenida donde el usuario puede interactuar y acceder a los diferentes recursos que tiene la pagina, estos recursos se encuentran en la carpeta recursos del proyecto.

La clase httpserver es la encargada de generar el servicio web, identificar los recursos necesarios e imprimirlos, además de hacer el llamado a la clase database en el paquete de persistencia que se encarga de hacer la conexión con la base de datos postgres que hay detrás y retornar los valores correspondientes que se consultaron.

El servicio de spark crea un socket que es utilizado por la clase httpserver para generar este nuevo servicio.

Ejecución:

- clonar el repositorio `git clone https://github.com/aosfandres/AREP-LAB3`
- construir el proyecto `mvn package`
- ejecutar el proyecto `java -cp target/classes edu.escuelaing.arep.App` se recomienda tener la última actualización de maven, de lo contrario ejecutar la clase 'App' desde su IDE preferido.
- ingresar a <http://localhost:36000/>



Bienvenido al servicio Web

Con las siguientes rutas podrá ver imagenes y tablas de base de datos.

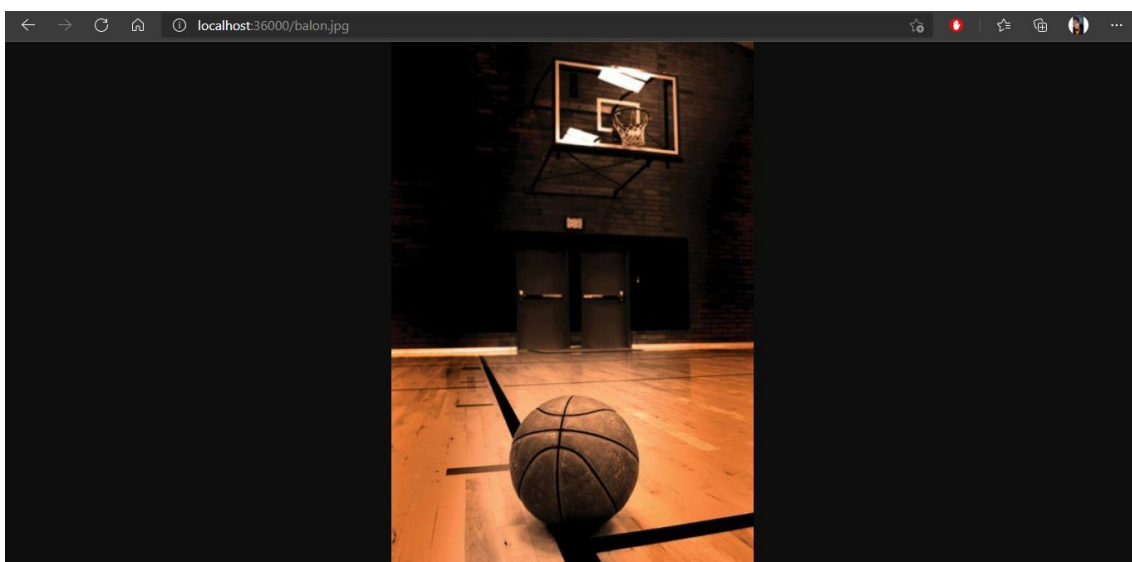
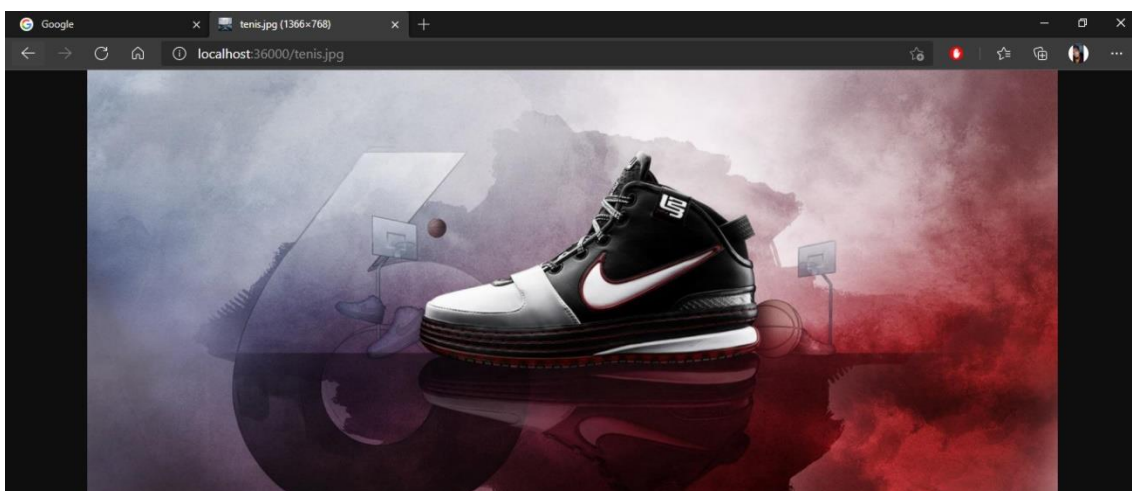
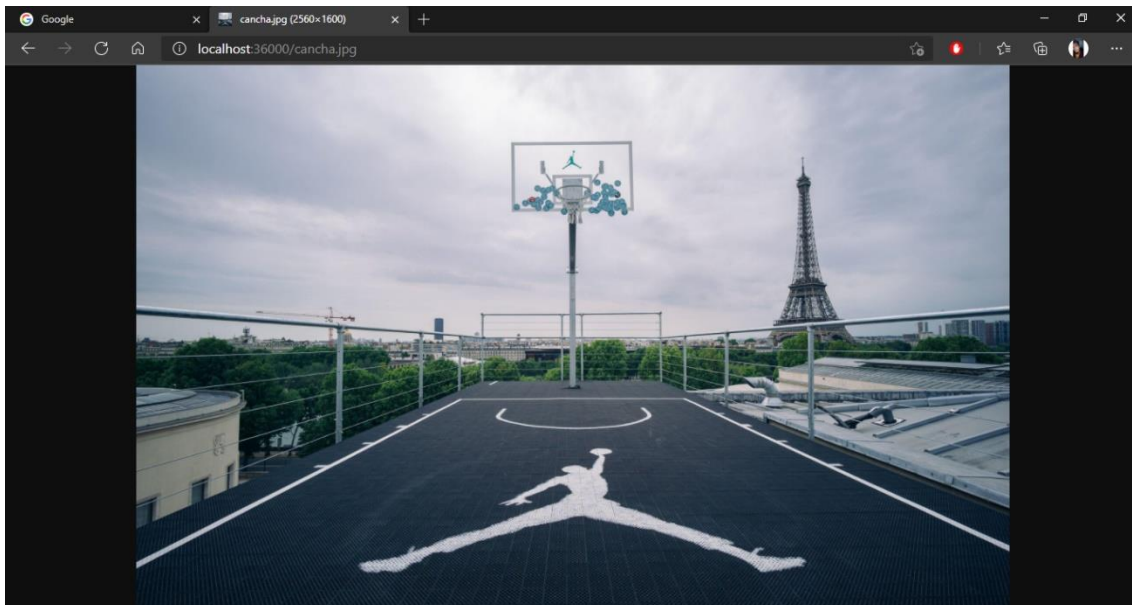
[Imágen Cancha](#)

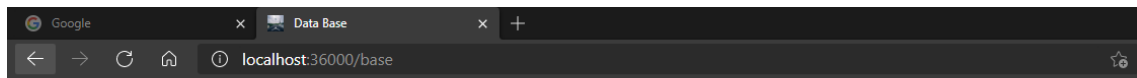
[Imágen Zapatillas](#)

[Imágen Balon](#)

[Base de Datos de elementos fundamentales para tu juego](#)

- Allí podrá ver y acceder a los recursos disponibles
- imagen





Basketball items

Nombre producto: 1: Balon Nombre producto: 2: Zapatillas Nombre producto: 3: Jersey Nombre producto: 4: Pantalón Nombre producto: 5: Protectores