**Herramientas usadas:**

- IDE Eclipse Neon

- Java version "1.8.0\_151"

- Apache Maven 3.3.9

- dropwizard framework 1.1.0

- dropwizard.swagger 1.0.6-1

- Apache commons-jcs-core 2.1 → Para cache

Sistema operativo y caracteristicas:

- Ubuntu 16.04.4 LTS

- Procesador Core i7, 12 GB RAM

Justificación:

Dropwizard es un framework que ofrece un conjunto de librerias específicas para el desarrollo de servicios web Restful.

Swagger por su parte permite describir la estructura del API.

El core de Dropwizard está formado por :

**Jetty** como servidor HTTP

**Jersey** para construir los servicios web Rest

**Jackson** como soporte a los objetos JSON

**Metrics** para obtener métricas del código

Descripción a grandes rasgos (arquitectura):

* Se genero un id de usuario para poder consumir el API de openweathermap, el cual se guardo en el archivo properties. Este id viaja como queryparam para el consumo de dicha API.
* Se genero un Client ID y Client secret (Oauth) para consumir el API de Spotify, el cual se guardo en el archivo properties.
* Se genera un encoding en Base64 (ClientID:ClientSecret) para consumir el endpoint que nos da un TOKEN necesario para la interacción con el API de Spotify.
* Se genera un mapa cache que almacena el TOKEN asi como su expiracion, esto con el fin de guardar el TOKEN y que solo consuma de nuevo el servicio cuando este ya haya expirado.
* Se usa un filtro para que por cada request (dentro del filtro se hacen las validaciones correspondientes) se anexe en el HEADER el TOKEN para la transacción de las playlists y sus tracks (Bearer) o bien para obtener el TOKEN (Basic).
* Se usa un archivo de herramientas con el patrón singleton para que no se instancie una vez. Este archivo extrae (contiene) las URLs de los endpoints de openweathermap y Spotify así como los IDS de las aplicaciones entre otros datos.

##############################################################################

**Instrucciones:**

* Descargar el proyecto maven "sin-delantal"
* Compilarlo con "mvn clean install"
* Se generará una carpeta target, entrar en dicha carpeta y teclear:

"java -jar sin-delantal-1.0-SNAPSHOT.jar server ../config.yml"

Donde:

config.yml -> es el archivo de configuración con los parametros necesarios/específicos para levantar la aplicación (puerto http, context path, etc).

Si se levanta en un ambiente WINDOWS entonces se recomienda colocar el archivo "config.yml" a nivel del jar generado, para que se teclee:

"java -jar sin-delantal-1.0-SNAPSHOT.jar server config.yml"

* Una vez ejecutada la instrucción anterior entonces se puede ingresar al framework swagger de la siguiete forma:

"http://localhost:8088/sin-delantal/prueba/v1/swagger"

Donde:

8088 es el puerto definido para esta prueba, este puerto se puede cambiar en el archivo "config.yml"

##############################################################################

**NOTAS:**

* El tamaño del jar generado es de aproximadamente 20 MB.
* Como característica del programa solo extrae los TRACKS de la **primera** playlist que encuentra.