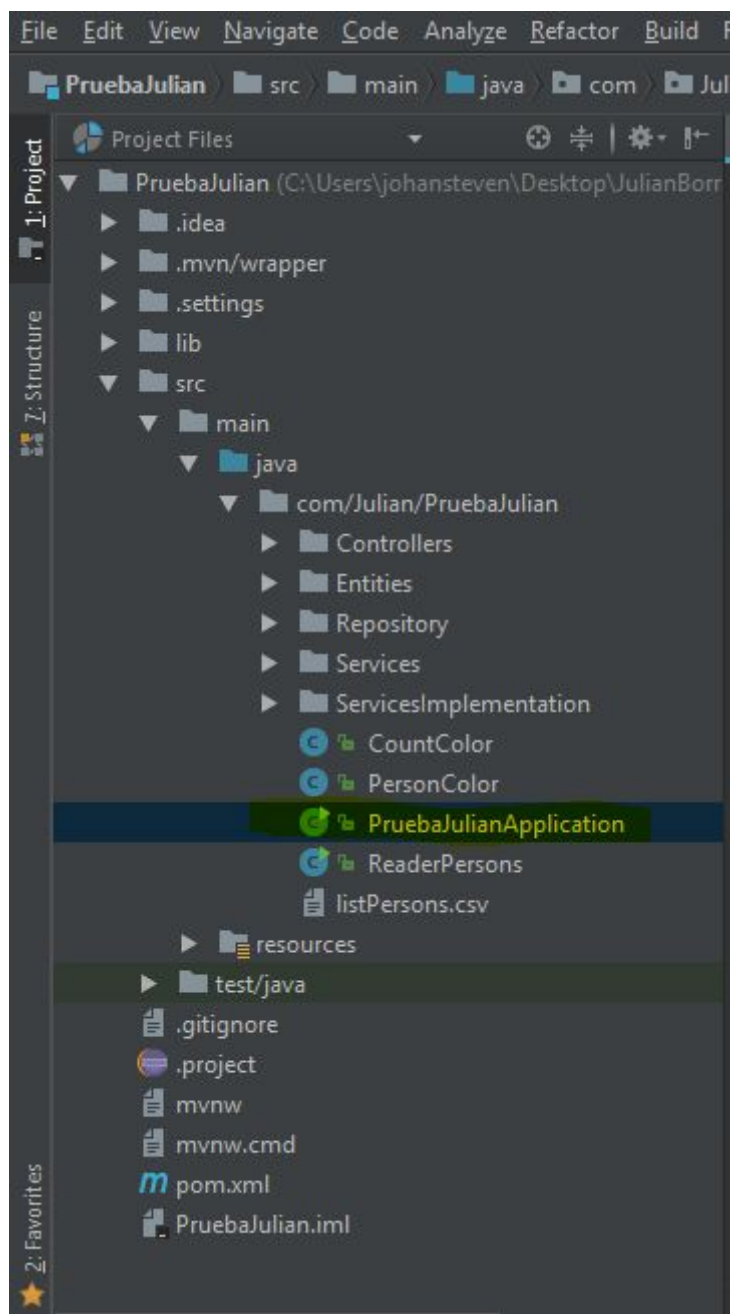


Manual “PruebaJulian”

Prueba técnica : Julian Andres Castro Ruge

1. Importe el proyecto a su IDE (es un proyecto maven).
2. Construya el proyecto maven.
3. Diríjase a la ruta
PruebaJulian/src/main/java/com/Julian/PruebaJulian/PruebaJulianApplication



4. Ejecute el archivo PruebaJulianApplication.java

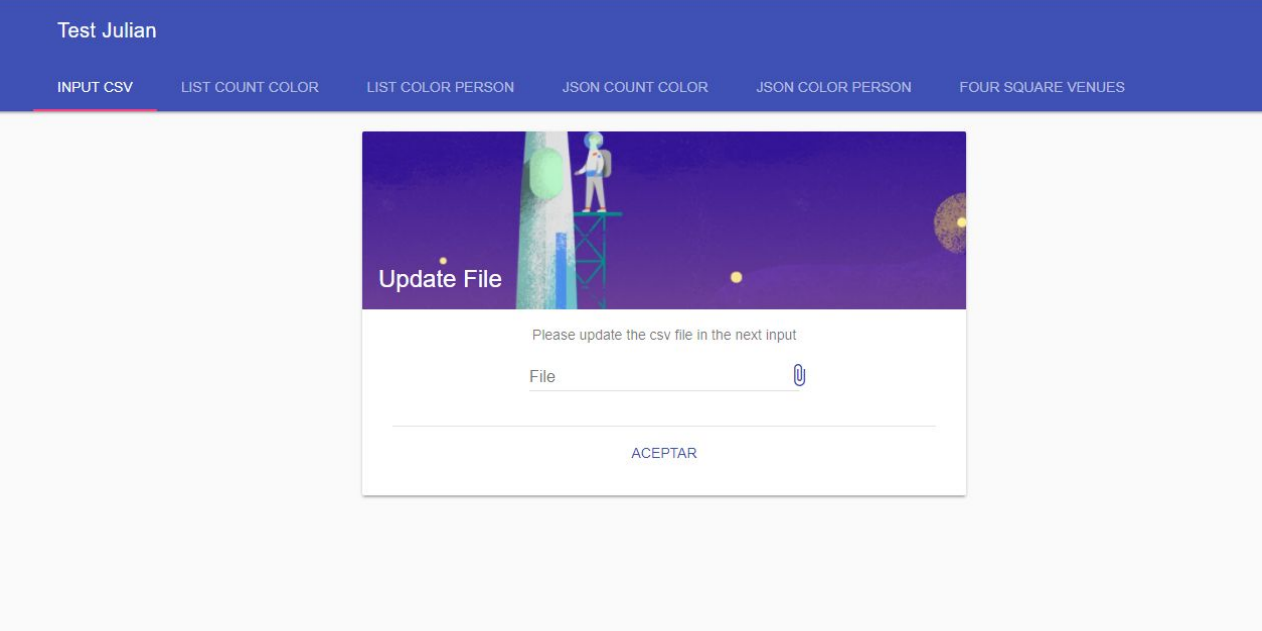
5. Posterior a eso espere a que aparezca el siguiente mensaje en la consola

```
Tomcat started on port(s): 8080 (http)
Started PruebaJulianApplication in 38.878 seconds
Initializing Spring FrameworkServlet 'dispatcherSer
FrameworkServlet 'dispatcherServlet': initializatio
FrameworkServlet 'dispatcherServlet': initializatio
HHH000397: Using ASTQueryTranslatorFactory
```

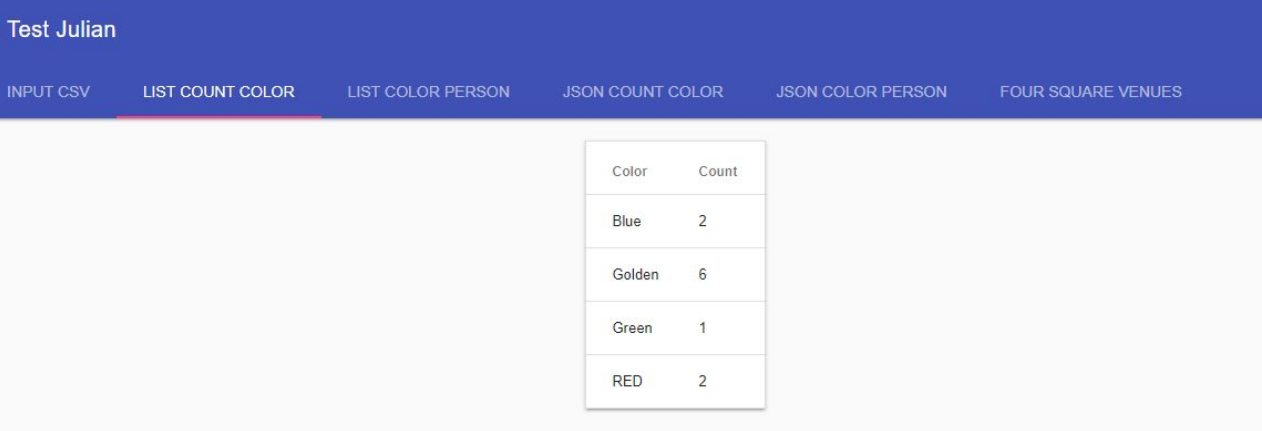
6. Diríjase a la siguiente ruta <http://localhost:8080/>

7. Este proyecto se divide en 6 vistas

a. Vista de archivo csv



b. Listado por conteo de colores



Color	Count
Blue	2
Golden	6
Green	1
RED	2

c. Listado por contenido de colores y listado de personas

Test Julian

INPUT CSVLIST COUNT COLORLIST COLOR PERSONJSON COUNT COLORJSON COLOR PERSONFOUR SQUARE VENUES

Color	Count	names
Blue	2	sebastian, castillo
Golden	6	Donald, Donald, Donald, Duck, Donald, julian
Green	1	castillo
RED	2	julian, julian

d. El conteo de colores en formato JSON que se presenta en dos vistas:

i. Principal

Test Julian

INPUT CSVLIST COUNT COLORLIST COLOR PERSONJSON COUNT COLORJSON COLOR PERSONFOUR SQUARE VENUES

[[{"color":"Blue","count":2},{ "color":"Golden","count":6},{ "color":"Green","count":1},{ "color":"RED","count":2}]]

ii. Secundaria de servicio expuesto a través de este link

<http://localhost:8080/listColor>

e. El conteo de colores y listado de personas en formato JSON que se representa en dos vistas:

i. Principal

Test Julian

INPUT CSVLIST COUNT COLORLIST COLOR PERSONJSON COUNT COLORJSON COLOR PERSONFOUR SQUARE VENUES

[[{"color":"Blue","count":2,"names":["sebastian","castillo"]},{ "color":"Golden","count":6,"names":["Donald","Donald","Donald","Duck","Donald","julian"]},{ "color":"Green","count":1,"names":["castillo"]},{ "color":"RED","count":2,"names":["julian","julian"]}]]

ii. Secundaria de servicio expuesto a través de este link

<http://localhost:8080/listPersonColor>

f. Vista de consumo de servicio de API FourSquare sobre la cual se solicitan tres datos básicos ubicación la cual es autocompletada si el usuario acepta compartir la ubicación, una palabra clave o “query” sobre la cual se realizará la búsqueda y finalmente una fecha. Si alguno de estos valores no es ingresado por el usuario, el programa tomará los valores por defecto que son lo siguientes `data=['40.7,-74','tacos','20171213']`

Al dar aceptar genera un alert con el JSON especificado en la prueba, mostrando los “places”.

Test Julian

INPUT CSV LIST COUNT COLOR LIST COLOR PERSON JSON COUNT COLOR JSON COLOR PERSON **FOUR SQUARE VENUES**

Parameters of Four Square

Please fill the input of the search

Location
4.710988599999999,-74.072092

Query
Hamburguesas

Date
13/12/2017

ACEPTAR

localhost:8080 dice:

```
{ "places": [ "Hamburguesas El Corral", "Hamburguesas El Corral",  
"Hamburguesas del Rodeo", "Hamburguesas El Corral", "Hamburguesas El  
Corral", "Hamburguesas El Corral", "Hamburguesas El Corral",  
"Hamburguesas Del Rodeo", "Hamburguesas El Corral", "Hamburguesas El  
Corral", "Hamburguesas El Corral", "Hamburguesas El Corral", "Bratwurst",  
"Hamburguesas El Corral", "Hamburguesas El Corral", "Hamburguesas El  
Corral", "Hamburguesas El Corral", "Hamburguesas Del Rodeo",  
"Hamburguesas El Corral - Estación Los Abuelos", "Hamburguesas El  
Corral", "Hamburguesas El Corral", "Hamburguesas El Corral",  
"Hamburguesas El Corral", "Hamburguesas El Corral", "Hamburguesas El  
Corral", "Hamburguesas El Corral", "Hamburguesas del Rodeo",  
"Hamburguesas del Rodeo", "hamburguesas inn", "Hamburguesas El  
Corral"] ] }
```

Aceptar

- g. Finalmente se publicó el servicio web en Amazon Web Service cuya url es la siguiente <http://ec2-18-220-21-249.us-east-2.compute.amazonaws.com:8080/>