

ReadMe

Contenido

Requisitos	1
Documentación del Código	1
Creación del fichero.....	2
Guardar String en el fichero.....	3
Eliminación de acentos.....	3
Comprobar Mayúsculas o Minúsculas	3
Buscar en el fichero y contar repeticiones	4
Funcionamiento	6

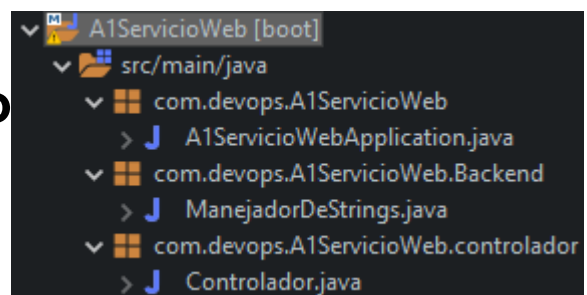
Requisitos

1. **Sistema operativo necesario:** Windows 10.
2. Se recomienda utilizar **Postman** para las consultas **Get** y **Post**.
3. **Tipo de Proyecto:** Maven.
4. **FrameWorks:** Spring.

Documentación del Código

El proyecto Java cuenta con tres paquetes:

- com.devops.A1ServicioWeb
- com.devops.A1ServicioWeb.Backend
- com.devops.A1ServicioWeb.controlador



En el paquete **com.devops.A1ServicioWeb** se encuentra la clase **A1ServicioWebApplication.java**, donde se encuentra la clase main del programa, la cual inicia el servidor y crea un nuevo fichero, en caso de que no exista ninguno ya.

```

package com.devops.A1ServicioWeb;
import com.devops.A1ServicioWeb.Backend.ManejadorDeStrings;
@SpringBootApplication
public class A1ServicioWebApplication {
    ....public static void main(String[] args) {
        ....» System.out.println("Cargando el contexto de Spring...");
        ....»
        ....» SpringApplication.run(A1ServicioWebApplication.class, args);
        ....»
        ....» System.out.println("Contexto de Spring cargado!");
        ....»
        ....» try {
        ....» ....» new ManejadorDeStrings().creacionDeFichero();
        ....» ....» } catch (IOException ex) {
        ....» ....» ex.printStackTrace();
        ....» ....» }
        ....» }
    }
}

```

En el paquete **com.devops.A1ServicioWeb.Backend** se encuentra la clase **ManejadorDeStrings.java**, donde se han programado todas las funciones para cumplir con los requisitos del ejercicio.

Creación del fichero

Si el fichero existe se imprime en pantalla “ya existe un fichero creado”, si no existe se imprime por pantalla “fichero creado”.

```

@Component
public class ManejadorDeStrings {
    ....private String fichero = "registro.txt";
    ....
    ....//CREACION DEL FICHERO "registro.txt"
    ....public void creacionDeFichero() throws IOException {
        ....File file = new File(fichero);
        ....if (file.exists()) {
        ....» System.out.println("Ya existe el fichero");
        ....» ....» file.createNewFile();
        ....» } else {
        ....» System.out.println("Fichero creado: ");
        ....» ....» file.createNewFile();
        ....» }
        ....}
    }
}

```

Guardar String en el fichero

Se crea un nuevo **FileWriter** con la opción (file, **true**) para que cada vez que se ejecute la aplicación se añada contenido al fichero en vez de sobrescribirlo.

Cada vez que se escribe en el fichero mediante el **BufferedWriter** se crea una nueva línea.

```
....////GUARDAR·STRING·EN·FICHERO·"registro.txt"¶
....public·void·guardarEnfichero(String·txt)·{¶
¶
.....File·file·=·new·File(fichero);¶
¶
.....try·(FileWriter·f·=·new·FileWriter(file,·true);¶
.....BufferedWriter·buffer·=·new·BufferedWriter(f))·{¶
.....buffer.write(txt);¶
.....buffer.newLine();¶
.....}·catch·(Exception·e)·{¶
.....//e.printStackTrace();¶
.....}¶
....}¶
```

Eliminación de acentos

Se eliminan los acentos de la String introducida por el usuario, retornando la misma String pero sin acentos, para usarla posteriormente.

```
....//ELIMINACIÓN·DE·ACENTOS¶
....public·String·normalizer(String·aux)·{¶
.....String·mystring·=·aux;¶
.....mystring·=·Normalizer.normalize(mystring,·Normalizer.Form.NFD);¶
.....mystring·=·mystring.replaceAll("[^\\p{ASCII}]",·"");¶
.....return·mystring;¶
....}¶
```

Comprobar Mayúsculas o Minúsculas

La función devuelve un booleano true o false, dependiendo de si se ha encontrado una letra mayúscula en el string introducido por el usuario o no.

```

....//COMPROBAR MAYUSCULAS O MINUSCULAS
....public boolean isMayuscula(String txt){
.....////////////////////
.....for (int i=0; i<txt.length(); i++){
.....if (Character.toUpperCase(txt.charAt(i))){
.....return true;
.....}
.....}
.....return false; //TRUE = MAYUSCULAS --- FALSE = MINUSCULAS
....}

```

Buscar en el fichero y contar repeticiones

Se inicia un contador a 0 para retornar el número de veces que se repite una cadena de texto introducida por el usuario, en el fichero “registro.txt”; el cual se guardará y se creará siempre en el mismo directorio desde el que se ejecute la aplicación.

```

....//BUSCAR EN FICHERO Y CONTAR OCURRENCIAS
....public int contarEnFichero(String txt){
.....int contador=0;
.....////////////////////
.....try {
.....FileReader file=new FileReader(fichero);
.....BufferedReader buffer=new BufferedReader(file);
.....//LEEMOS LA PRIMERA LINEA DEL FICHERO Y LE QUITAMOS LOS ACENTOS
.....String fraseSinAcentos=normalizer(buffer.readLine());
.....//LE QUITAMOS LOS ACENTOS A LA CADENA INTRODUCIDA POR EL USUARIO
.....String cadenaSinAcentos=normalizer(txt);
.....//COMPROBAMOS SI LA CADENA INTRODUCIDA POR EL USUARIO TIENE O NO MAYUSCULAS
.....if (isMayuscula(cadenaSinAcentos)==true){
.....//SI TIENE MAYUSCULAS
.....String fraseSinAcentosYenMayusculas=fraseSinAcentos.toUpperCase();
.....String cadenaSinAcentosYenMayusculas=cadenaSinAcentos.toUpperCase();
.....while (fraseSinAcentosYenMayusculas!=null){
.....if (fraseSinAcentosYenMayusculas.contains(cadenaSinAcentosYenMayusculas)){
.....contador++;
.....}
.....System.out.println(fraseSinAcentosYenMayusculas);
.....fraseSinAcentosYenMayusculas=normalizer(buffer.readLine()).toUpperCase();
.....}
.....} else {
.....//SI SOLO SON MINUSCULAS
.....while (fraseSinAcentos!=null){
.....if (fraseSinAcentos.contains(cadenaSinAcentos)){
.....contador++;
.....}
.....System.out.println(fraseSinAcentos);
.....fraseSinAcentos=normalizer(buffer.readLine());
.....}
.....}
.....} catch (Exception e){
.....e.printStackTrace();
.....}
.....return contador;
....}

```

En el paquete **com.devops.A1ServicioWeb.controlador** se encuentra la clase **controaldor.java**, con los métodos `getText` y `postText`.

1. **getText**: Es un método que utiliza el verbo http get, para introducir una cadena de texto como pathVariable, en la cabecera de la request, que se pasará como parámetro del método **contarEnFichero(String txt)** correspondiente a la clase **ManejadorDeStringsYficheros**; el cual retornará al usuario el número de veces que se repite su cadena en el fichero "registro.txt".
2. **postText**: Es un método que utiliza el verbo http post, para introducir una cadena de texto a través del body de la request, con el objetivo de añadir dicha cadena de texto al fichero "registro.txt".

```
@RestController
public class Controlador {

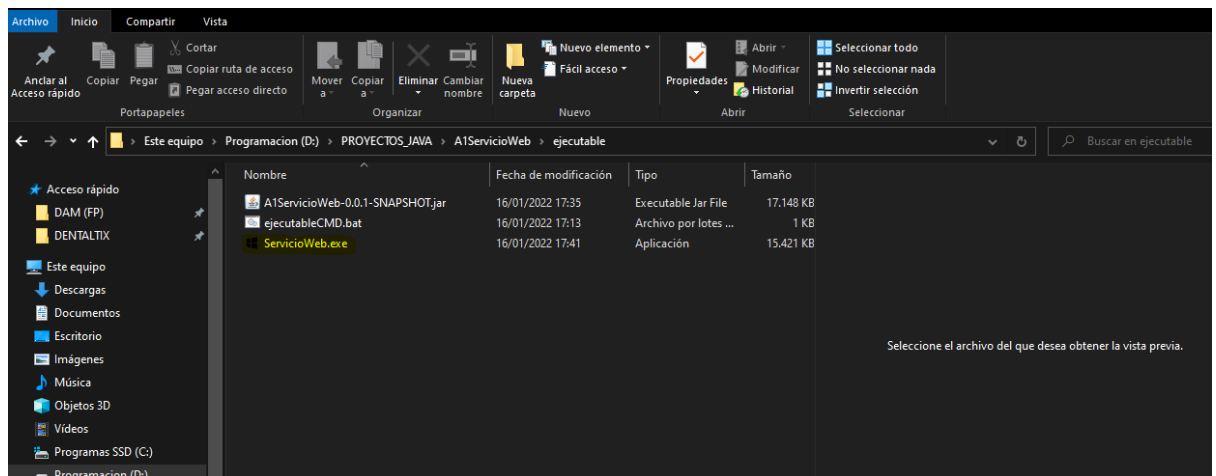
    @Autowired
    private ManejadorDeStringsYFicheros manejadorDeStrings;

    /**
     *
     * @param txt
     * @return ResponseEntity
     */
    //MÉTODO QUE DEVUELVE EL NÚMERO DE OCURRENCIAS DE
    //UNA CADENA DE TEXTO INTRODUCIDA POR EL USUARIO
    //COMO VARIABLE EN LA CABECERA DE LA REQUEST
    //http://localhost:12345/
    @GetMapping(path =("/{txt})", produces = MediaType.ALL_VALUE)
    public ResponseEntity<String> getText(@PathVariable("txt") String txt) {
        System.out.println("\nPalabra consultada " + txt);
        int numeroDeOcurrencias = manejadorDeStrings.contarEnFichero(txt);
        return new ResponseEntity<>(String.valueOf(numeroDeOcurrencias), HttpStatus.OK);
    }

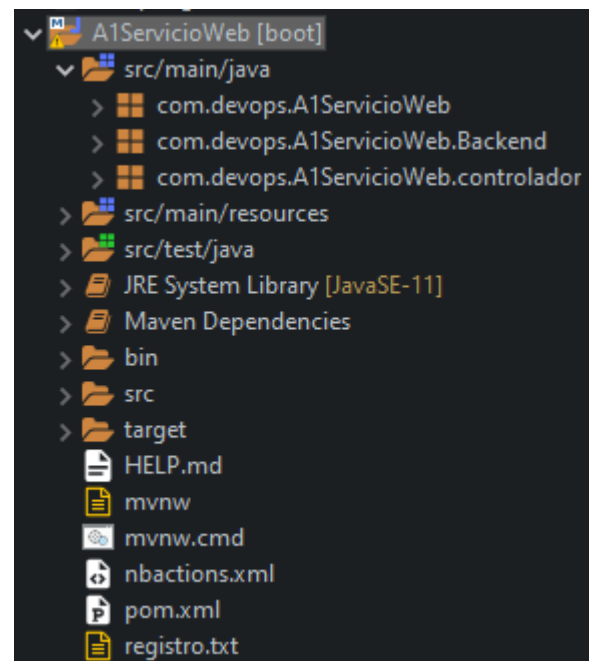
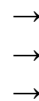
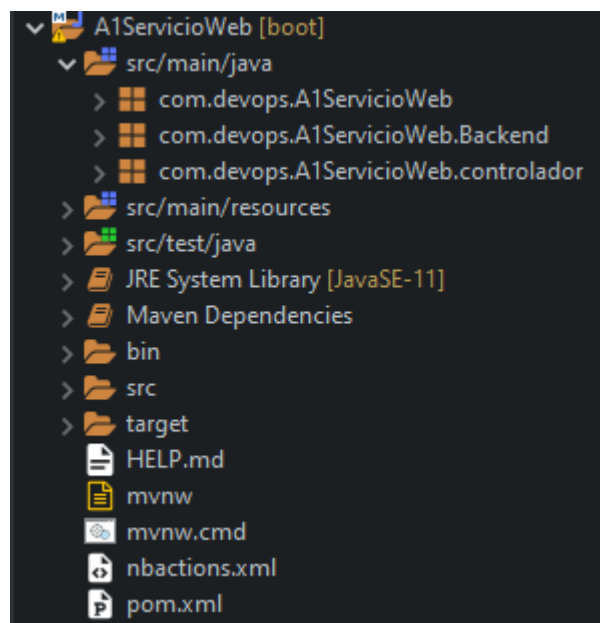
    /**
     *
     * @param txt
     * @return ResponseEntity
     */
    //MÉTODO POR EL QUE EL USUARIO AÑADE UNA CADENA DE TEXTO
    //AL FICHERO "registro.txt" (PERSISTENCIA, DENTRO DEL BODY
    //DE LA REQUEST
    //http://localhost:12345/newstring
    @PostMapping(path = "newstring", produces = MediaType.ALL_VALUE,
        consumes = MediaType.ALL_VALUE)
    public ResponseEntity<String> postText(@RequestBody String txt) {
        System.out.println("\nPalabra a insertar en fichero \"registro.txt\" : " + txt);
        manejadorDeStrings.guardarEnfichero(txt);
        return new ResponseEntity<>(txt, HttpStatus.OK);
    }
}
```

Funcionamiento

1. Hacemos doble clic en el ejecutable “ServicioWeb.exe”



Se ejecuta



```
C:\WINDOWS\system32\cmd.exe
=====|_|=====|_|_/_/_/
:: Spring Boot :: (v2.6.2)

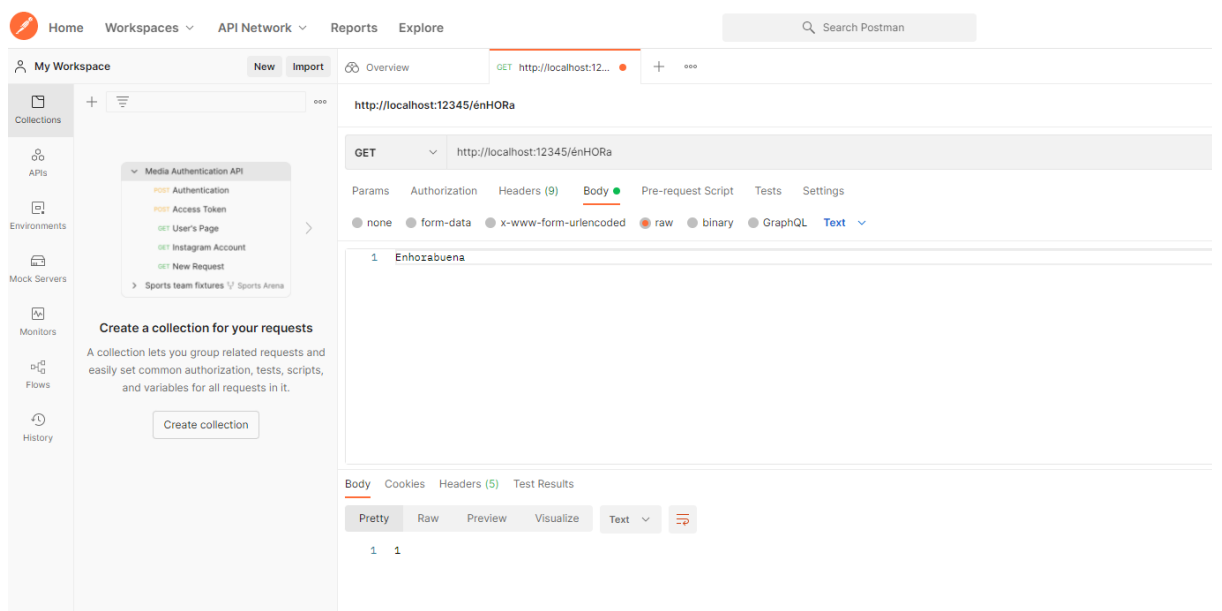
2022-01-16 17:50:45.817 INFO 14324 --- [main] c.d.A.A1ServicioWebApplication : Starting A1Servicio
WebApplication v0.0.1-SNAPSHOT using Java 11.0.12 on PCGONZALO with PID 14324 (D:\PROYECTOS_JAVA\A1ServicioWeb\ejecutabl
e\A1ServicioWeb-0.0.1-SNAPSHOT.jar started by Gonzalo in D:\PROYECTOS_JAVA\A1ServicioWeb\ejecutable)
2022-01-16 17:50:45.817 INFO 14324 --- [main] c.d.A.A1ServicioWebApplication : No active profile s
et, falling back to default profiles: default
2022-01-16 17:50:46.645 INFO 14324 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized
with port(s): 12345 (http)
2022-01-16 17:50:46.645 INFO 14324 --- [main] o.apache.catalina.core.StandardService : Starting service [T
omcat]
2022-01-16 17:50:46.645 INFO 14324 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet en
gine: [Apache Tomcat/9.0.56]
2022-01-16 17:50:46.692 INFO 14324 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring
embedded WebApplicationContext
2022-01-16 17:50:46.692 INFO 14324 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplication
Context: initialization completed in 828 ms
2022-01-16 17:50:46.942 INFO 14324 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on p
ort(s): 12345 (http) with context path ''
2022-01-16 17:50:46.942 INFO 14324 --- [main] c.d.A.A1ServicioWebApplication : Started A1ServicioW
ebApplication in 1.448 seconds (JVM running for 1.777)

Fichero "registro.txt" creado: true

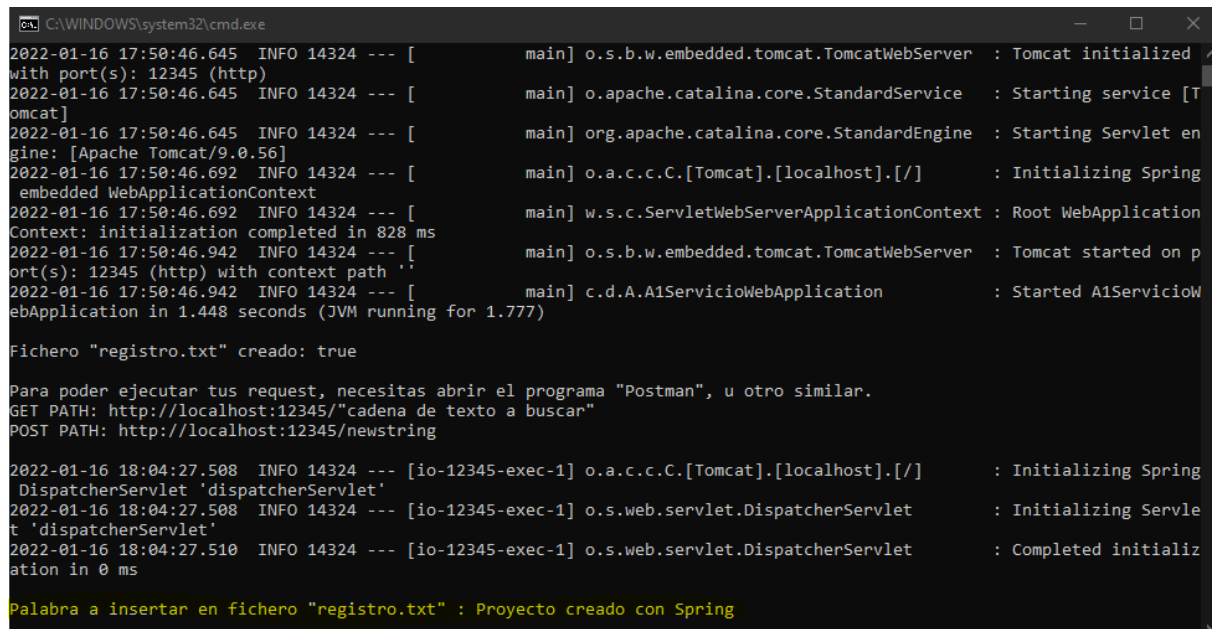
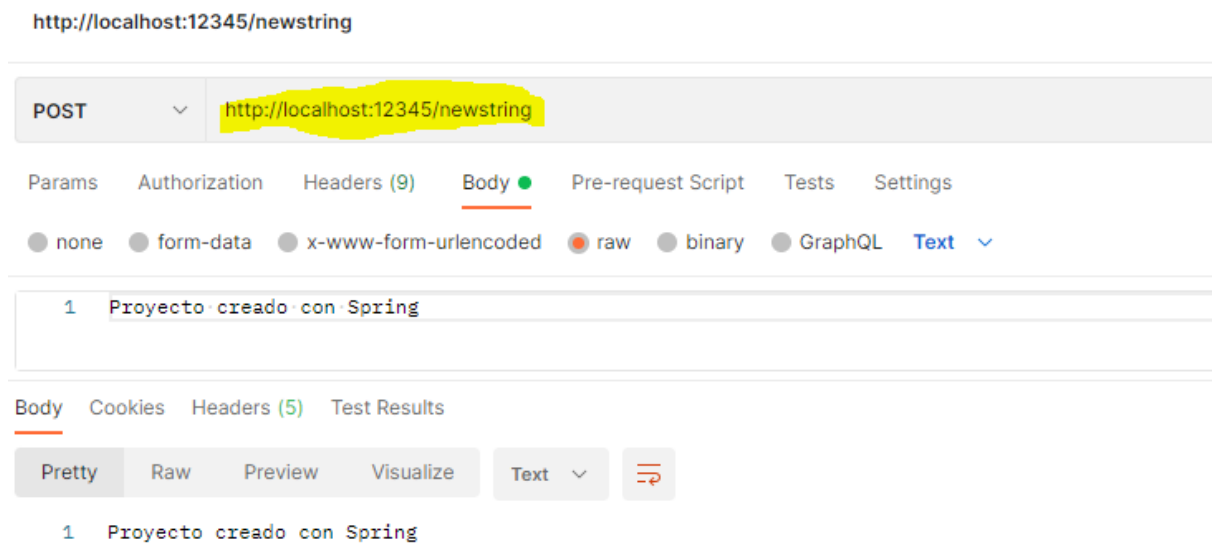
Para poder ejecutar tus request, necesitas abrir el programa "Postman", u otro similar.
GET PATH: http://localhost:12345/"cadena de texto a buscar"
POST PATH: http://localhost:12345/newstring
```

Si existe ya un fichero de persistencia “registro.txt”, el fichero no se creará, sino que se sobrescribirá el fichero ya existente. En caso contrario, el programa creará un fichero llamado “registro.txt” automáticamente, al ejecutarse.

2. Abrimos el programa postman u otro similar, necesario para poder ejecutar las request hacia nuestro servidor web.



3. Para insertar una cadena de texto en el fichero "registro.txt", debemos realizar la siguiente request: <http://localhost:12345/newstring> e insertar una cadena de texto en el body de nuestra request.



4. Para consultar cuantas veces se repite una cadena de texto en el fichero "registro.txt", debemos ejecutar la siguiente request: `http://localhost:12345/"Texto a insertar"`. Si dicha cadena de texto se encuentra en el fichero "registro.txt", el servidor devolverá el número exacto de veces que esta se repite.

http://localhost:12345/SprínG


GET ▼ http://localhost:12345/SprínG

Params Authorization Headers (7) Body Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL Text ▼

1

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text ▼ 

1 1

```
C:\WINDOWS\system32\cmd.exe
2022-01-16 17:50:46.645 INFO 14324 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.56]
2022-01-16 17:50:46.692 INFO 14324 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-01-16 17:50:46.692 INFO 14324 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 828 ms
2022-01-16 17:50:46.942 INFO 14324 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 12345 (http) with context path ''
2022-01-16 17:50:46.942 INFO 14324 --- [main] c.d.A1ServicioWebApplication : Started A1ServicioWebApplication in 1.448 seconds (JVM running for 1.777)

Fichero "registro.txt" creado: true

Para poder ejecutar tus request, necesitas abrir el programa "Postman", u otro similar.
GET PATH: http://localhost:12345/"cadena de texto a buscar"
POST PATH: http://localhost:12345/newstring

2022-01-16 18:04:27.508 INFO 14324 --- [io-12345-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2022-01-16 18:04:27.508 INFO 14324 --- [io-12345-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2022-01-16 18:04:27.510 INFO 14324 --- [io-12345-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 0 ms

Palabra a insertar en fichero "registro.txt" : Proyecto creado con Spring

Palabra consultada SprínG
PROYECTO CREADO CON SPRING
Coincidencias de la string: SprínG,en el fichero "registro.txt" = 1
```

GitHub: https://github.com/GonzaloSC95/Actividad1_ServicioWeb.git