

Aplicación distribuida segura en todos sus frentes

María Fernanda Hernández Vargas

Escuela Colombiana de Ingeniería Julio Garavito
de septiembre de 2020

1 INTRODUCCIÓN

En este artículo se vera la arquitectura que se realizó para un servicio web seguro. Para llevar a cabo este laboratorio se utilizaron herramientas como Docker, el servicio de virtualización ofrecido por AWS (EC2), llaves seguras para sitios web con su respectivo certificado y los servicios con los que vamos a realizar la prueba del sitio seguro programados con java.

2 DIAGRAMA

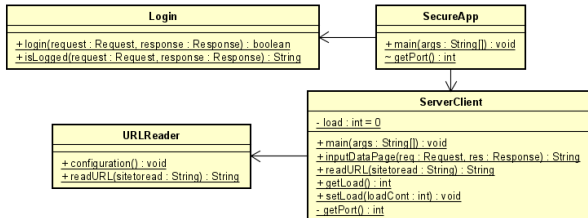


Imagen 1. Diagrama de Clases

SecureApp es un programa en java que utiliza un login, el cual tiene encriptado la contraseña del usuario que tiene acceso a la aplicación y poder acceder al servicio ServerClient, el cual realiza una consulta a una base de datos, ambos servicios con una configuración con llaves creadas en los directorios para que los sitios sean seguros y la comunicación entre ellos.

3 ARQUITECTURA

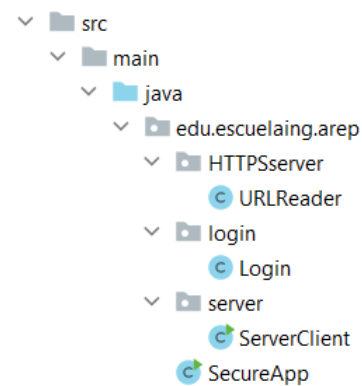


Imagen 2. Arquitectura de Archivos

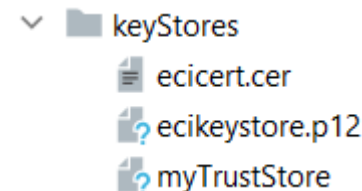


Imagen 3. Directorio de la llave y el certificado

Al subir los servicios a AWS EC2 y realizar la respectiva comunicación cliente-servidor, tenemos como resultado lo siguiente.



Login

Nombre de usuario:

Contraseña:

Login

Imagen 3. Vista inicio

En esta vista podemos observar que se está realizando la petición con el protocolo HTTPS, es decir es un sitio seguro en el que nuestro navegador confía para la comunicación y navegación libre. Al realizar el ingreso con el usuario correspondiente tenemos la correspondiente respuesta del servidor de consulta a la base de datos.

Ingrese sus datos para Acceder

Nombre de usuario:

Contraseña:

Imagen 4. Vista ingreso datos



Imagen 5. Vista respuesta del inicio de sesión

Si los datos son incorrectos el servidor no accede al servicio de consulta y arroja un error.

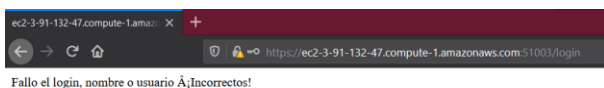


Imagen 6. Error en el inicio de sesión

En este laboratorio se aprendió como proteger un sitio web y la importancia de realizar este procedimiento en futuros proyectos.

6 BIBLIOGRAFÍA

- baeldung*. (s.f.). Obtenido de Spring boot certificate:
<https://www.baeldung.com/spring-boot-https-self-signed-certificate>
- campusvirtual*. (s.f.). Obtenido de Taller secureapp:
http://campusvirtual.escuelaing.edu.co/moodle/pluginfile.php/230587/mod_resource/content/0/TallerAllSecureAppSpark.pdf
- github*. (s.f.). Obtenido de spark ssl:
<https://github.com/tipsy/spark-ssl>
- Oracle*. (s.f.). Obtenido de gjrgy:
<https://docs.oracle.com/cd/E19798-01/821-1841/gjrgy/>
- Oracle*. (s.f.). Obtenido de AWS-Amazon:
<https://docs.oracle.com/cd/E19509-01/820-3503/ggfen/index.htmlhttps://aws.amazon.com/es/serverless/build-a-web-app/>