

**Aplicación distribuida segura en todos
sus frentes**

Santiago Martínez Martínez

**Ingeniería de Sistemas
Noveno Semestre
Escuela Colombiana de Ingeniería Julio Garavito
Bogota DC - Colombia**

Glosario

Docker: Es una tecnología para la creación de contenedores.

Contenedores: Son entornos ligeros, que tienen un tiempo de ejecución que proporcionan a las apps las dependencias o aplicaciones que se necesitan para su ejecución.

SSL: Estándar de seguridad global que permite la transferencia de datos cifrados entre un navegador y un servidor web.

Cloud Computing: Servicios ofrecidos para tener acceso remoto a Software, almacenamiento y alojamiento para nuestros servicios.

EC2 amazon: Permite la creación de instancias o servidores virtuales en la nube de AWS.

Resumen

Este documento tiene como objetivo aprender a implementar servicios seguros utilizando certificados, en este caso implementaremos una aplicación que nos permita leer información solamente si estamos autenticados en la aplicación y que esta se pueda conectar a otra aplicación mediante el uso de certificados entre los dos servicios.

La arquitectura se base en que desde el Browser se conecte al servidor web por medio de HTTPS, luego a esto, este servidor debe de tener un LoginService el cual utilice, y que desde este servidor nos podamos conectar a otro servidor web utilizando https.

Para implementar este laboratorio se utilizó Spark, también se utilizó request session de spark para manejar las sesiones, para hacer los request HTTP se utilizó JavaScript y para la interfaz grafica se utilizó HTML y CSS.

Contents

1	Introduction	2
2	Objetivos	3
3	Solución Requerimientos	3
4	Conclusion	3

1 Introduction

Para poder realizar una aplicación segura se utilizó la herramienta keytool, la cual es una herramienta que ya viene al descargar el JDK de java que tiene el propósito de administración de claves y certificados para generar y administrar los certificados SSL. Con esto garantizamos el poder tener una conexión segura y certificada entre los dos servicios que se implementan.

Se manejaron sesiones para poder garantizar que el usuario al momento de autenticarse se pueda garantizar que si es el y que tiene permisos para poder realizar las acciones que se le permitan.

```
secure("keystores/ecckeystore.p12", "123456", null, null);
port.getPort();
users.put("user@ect.com", "ect");
SecureURL.openConnection();
StaticFilesConfiguration staticHandler = new StaticFilesConfiguration();
staticHandler.configure("/public");

before((request, response) -> staticHandler.consume(request.raw(), response.raw()));

before("/private/*", (request, response) -> {
    request.session(true);
    if (request.session().isNew()) {
        request.session().attribute("isLoggedIn", false);
    }
    boolean auth = request.session().attribute("isLoggedIn");
    if (!auth) {
        halt(401, "<h1>Por favor realiza el login/h1");
    }
});

before("/index.html", ((request, response) -> {
    request.session(true);
    boolean auth = request.session().attribute("isLoggedIn");
    if (auth) {
        response.redirect("/private/index.html");
    }
    if (request.session().isNew()) {
        request.session().attribute("isLoggedIn", false);
    }
}));
```

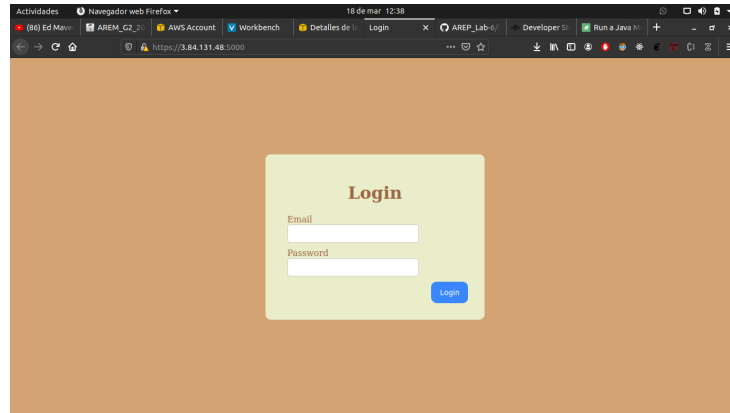
2 Objetivos

Poder realizar una aplicación segura la cual utilice certificados SSL y pueda realizar una conexión por medio de HTTPS a otro servicio corriendo.

Poder realizar autenticación por parte de un usuario apartir de un login.

3 Solución Requerimientos

Para cumplir con el requerimiento se crearon dos certificados, uno para cada proyecto, se utilizo la herramienta keytool, se integro con Spark usando la función Secure que nos permite almacenar e integrar nuestros certificado a la aplicación, ademas de esto, se implemento un Login con el cual se maneja la session del usuario para que pueda autenticarse.



4 Conclusion

1. Al momento de implementar esta aplicación segura, uno puede encontrar poca documentación, esto hizo que el proceso sea más complicado.
2. Herramientas como Keytool nos permiten mantener nuestros servicios seguros y de forma gratuita.
3. Roles y session es la mejor forma de manejar la autenticidad, autorización y la integridad de la aplicación.

References

- [1] *MDV*.
https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Promise
- [2] *KeyTool*.
<https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html>
- [3] *Docker Documentation*.
<https://docs.docker.com/>
- [4] *Tutorial*.
<https://es.overleaf.com/learn/latex/Tutorials>