

1. Redtrust es un Sistema de Gestión de Certificados Centralizada.

Explicado de manera muy simplificada consiste en un servidor donde el administrador puede configurar ciertas **políticas de uso** para los **usuarios** configurados en el sistema para los diferentes **certificados digitales** almacenados en el sistema. Estas políticas además pueden configurarse sobre todas las urls o **sobre determinadas urls**. Los usuarios finales instalan en su equipo un **agente Redtrust** que se conecta al servidor, en su equipo pueden logarse en el agente y listar en el agente los certificados que tienen asignados. Según las políticas configuradas en el servidor Redtrust (sistema de lista blanca) el usuario una vez logado en el agente, podrá o no hacer uso de los certificados durante la navegación web en cualquiera de los browsers de la máquina (Chrome, Edge, IE, Firefox). Si está permitido el uso podrá usar el certificado almacenado en Redtrust y se logará un evento de uso de certificado en el servidor. Si no está permitido no se podrá hacer uso de certificado y se logará un evento de uso denegado en el servidor.

Ejemplo de posibles políticas configuradas en servidor Redtrust:

- a. Usuarios en el sistema: user01, user02, user03
- b. Certificados en el sistema: cert01, cert02, cert03
- c. Política1: user01 puede usar cert02 en url: todas: .*

Se pide realizar un **Diseño de Pruebas** para verificar el correcto uso de certificados en el Agente Redtrust. No hace falta detallar los pasos de cada test de prueba. Pero sí descripción general de las pruebas a realizar, tipos de pruebas y las metodologías de test utilizadas.

En caso de dudas sobre el enunciado no dudéis en contactar con nosotros.

2. Escoger una de las 2 pruebas de automatización: frontend o api

2.1 Prueba automatización web

Realizar un test automático con la herramienta de automatización Selenium WebDriver con algún lenguaje de programación conocido: java, Python, C# u otra escogida sobre la funcionalidad de login en browserstack (https://www.browserstack.com/users/sign_in). (Crear una cuenta de prueba)

Adjuntar el código implementado, adjuntar una explicación de la prueba si se considera necesario. Si no codificas todos los casos de pruebas, especifica los que implementarías.

2.2 Prueba automatización api

Utiliza la herramienta con la que te sientas más cómodo, ya sea Postman u otra herramienta automatización de api o directamente código java, Python, C#.

La documentación de API de Spotify se encuentra en

<https://developer.spotify.com/documentation/web-api/reference/>

Usando la api pública de Spotify crear test api para la funcionalidad de añadir un tema a la playlist. **POST** https://api.spotify.com/v1/playlists/{playlist_id}/tracks

Adjuntar el fichero con el test realizado y una explicación del test si se considera necesario.

Si no codificas todos los casos de pruebas, especifica los que implementarías.

No hace falta adjuntar la parte de autenticación (se pueden usar tokens de la web de desarrollo)