

M8. Tarea 2

Usando alguna aplicación móvil de generación y gestión de códigos temporales TOTP

Ejemplo:

- Google Authenticator
 - <https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2>
 - <https://itunes.apple.com/es/app/google-authenticator/id388497605>

Programar un componente software que permita:

- Generar un código QR para añadir un servicio en las aplicaciones mencionadas anteriormente
- Validar un código TOTP introducido por el usuario para verificar si es correcto o no para el servicio anterior

Para esta tarea he utilizado la aplicación y API de Google Authenticator:

<https://authenticatorapi.com/>

He generado un pequeño programa en Python que permite el emparejamiento de la app con Google Authenticator y su validación.

Lo primero que necesitamos en emparejar nuestra aplicación con la API de Google Authenticator. Para ello vamos a generar un código QR generando una URL con los siguientes parámetros: AppInfo, AppName y SecretCode:

```
# Variables to set the pairing
AppInfo = "Module_8_Task_2"
AppName = "Modulo_8_Iker"
SecretCode = "6c6d7de78073b3920b0"
```

En mi caso el nombre de la aplicación es Modulo_8_Iker, la información es Module_8_Task_2 y el SecretCode es el hash en SHA256 de Modulo_8_Iker. Este secreto no deberíamos publicarlo, así como así, pero en este caso que es para un ejercicio de ejemplo lo he dejado así.

El programa nos genera la URL y la abre:

```
# Make the url
url = "https://www.authenticatorapi.com/pair.aspx?AppName=" + AppName + "&AppInfo=" + AppInfo + "&SecretCode=" + SecretCode
print(url)
webbrowser.open(url)
```

Tenemos el QR:



Escaneando el mismo con la app de Google Authenticator conseguimos el emparejamiento de nuestra app con la de Google:

Modulo_8_Iker (Module_8_Task_2)

148 196



El segundo paso consiste en validar el pin code que nos da la aplicación. Lo vamos a validar con un script realizado en Python que nos pide por consola el pin y nos indica si es correcto o no:

```
# Enter the pin code
pinCode = input("Enter the pin code: ")
SecretCode = "6c6d7de78073b3920b079eff2c97"
print(pinCode)
```

Junto con el ScretCode, que de nuevo esta metido en el código, siendo una practica nada recomendable en entornos de producción, se genera la URL que nos va a devolver un True or False, en referencia a nuestro pin code y nuestra app.

```
if result.text == "True":
    print("Pin correcto.")
else:
    print("El pin introducido no es correcto.")
```

Ejecutamos el programa para ver un ejemplo, el mismo nos pide el pin y si ponemos uno cualquiera nos indica que es incorrecto:

```
Enter the pin code: 123456
123456
El pin introducido no es correcto.
```

Si chequeamos Google Authenticator e indicamos el pin que nos dice la app, nos indica que sí que es válido:

```
Enter the pin code: 148196
148196
Pin correcto.
```

Con estos dos pequeños programas conseguimos emparejar nuestra app a Google Authenticator, que esta nos genere códigos para la autenticación en dos factores y los podamos validar en nuestra app.