***PRÁCTICA: CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA***

**ENTREGABLE 1**

**Grado en Ingeniería en Informática, Grupo 82, 07/10/2023**

Realizado por el grupo de prácticas 8, formado por:

* Raúl Sanz Belmar, 100472372@alumnos.uc3m.es
* Mario Rodríguez Román, 100472194@alumnos.uc3m.es

Enlace repositorio GitHub: <https://github.com/Rawwl03/Criptografia_Uc3m>

Índice de Contenidos

[**PROPÓSITO DE LA APLICACIÓN** 3](#_Toc148019317)

[**AUTENTICACÓN DE USUARIOS** 3](#_Toc148019318)

[**CIFRADO SIMÉTRICO Y ASIMÉTRICO** 3](#_Toc148019319)

[**ATENTICACIÓN DE MENSAJES: MAC** 3](#_Toc148019320)

[**CONCLUSIÓN** 3](#_Toc148019321)

# **PROPÓSITO DE LA APLICACIÓN**

El proyecto va a estar basado en la creación de una aplicación de servicios que te permita comprar entradas para el cine y gestionar tus entradas asociadas a tu cuenta personal. Todo esto contendrá una seguridad frente ataques proporcionada por la encriptación

# **AUTENTICACIÓN DE USUARIOS**

Dentro de nuestra aplicación lo primero que se ejecuta es la autenticación de usuarios registrados o el registro de usuarios válidos. En este proyecto hemos escogido la opción de autenticar a los usuarios mediante algo que sabemos, en este caso, contraseñas secretas. Estas contraseñas serán robustas, ya que deben de cumplir las siguientes características:

* Longitud mínima de 10 caracteres
* Mínimo una mayúscula
* Mínimo un número

La idea ejecutada consiste en guardar las contraseñas encriptadas de cada usuario y su salt encriptado en un archivo .json, de manera que cada vez que acceda con un usuario solo haga falta escribir la contraseña correctamente, que se encriptará mediante el mismo algoritmo y mismas características para verificar que las contraseñas encriptadas (y las normales) son iguales. Entre todos los algoritmos que existen de cifrado de contraseñas, hemos utilizado el siguiente con las siguientes características:

**PBKDF2HMAC** con algoritmo **hashes.sha256()**

Longitud 32

Uso de salt

Iteraciones 30000

***CÓDIGO UTILIZADO:***

kdf = PBKDF2HMAC(algorithm=hashes.SHA256(), length=32, salt=salt, iterations=30000)

# En kdf tenemos nuestro algoritmo que vamos a utilizar para encriptar las contraseñas introducidas.

# **CIFRADO SIMÉTRICO Y ASIMÉTRICO**

# **ATENTICACIÓN DE MENSAJES: MAC**

# **CONCLUSIÓN**

Lista usuarios:

1. Username: Aparicio, contraseña: Oracle12

* Tarjetas Aparicio:
  1. Número: 456745674567456, Fecha: 12/2024, CVV: 456
  2. Número:1234567812345678, Fecha: 01/2028, CVV: 999
  3. Número: 3459347834593478, Fecha: 07/2025, CVV: 869