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

**ENTREGABLE 2**

**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](#_Toc152103624)

[**DISEÑO** 3](#_Toc152103625)

[**MEJORAS IMPLEMENTADAS** 4](#_Toc152103626)

[**FIRMA DIGITAL** 5](#_Toc152103627)

[**CERTIFICADOS DE CLAVE PÚBLICA** 5](#_Toc152103628)

[**COMPLEJIDAD DE LA APLICACIÓN** 5](#_Toc152103629)

# **PROPÓSITO DE LA APLICACIÓN**

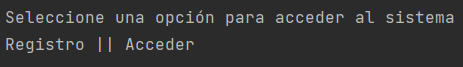
El proyecto va a estar basado en la creación de una aplicación de servicios que te permita comprar entradas para el cine y gestionar tus entradas asociadas a tu cuenta personal. Además, se podrán gestionar tarjetas asociadas a tu usuario para realizar los pagos, y cambiar la contraseña desde dentro de la aplicación. Todo esto contendrá una seguridad frente ataques proporcionada por la encriptación, de manera que vamos a guardar datos como contraseñas y tarjetas en una base de datos, pero estos estarán cifrados (y autenticados en el caso de las tarjetas) gracias a la librería cryptography. El diseño de la aplicación está comprendido por una serie de menús, cada uno con unas funcionalidades, que se manejan a través de la terminal (escribiendo inputs) y se puede navegar entre ellos. Aparte de la aplicación, se han añadido mejoras opcionales que se comentarán más adelante en profundidad, como implementación de base de datos y rotación de claves.

## **DISEÑO**

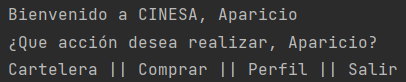
Como se ha mencionado antes, el diseño de la aplicación se basa en una serie de menús que están enlazados entre sí. Todos estos menús se generan en la clase Terminal, que representa a la terminal donde se va a mostrar todo, y se va a utilizar una variable global llamada ‘contrasena\_sys’, que almacenará la contraseña en claro del usuario actual.

Los menús principales que podemos encontrar son los siguientes:

1. **Menú acceso al sistema** -> este menú se trata del menú inicial que se muestra al iniciar la aplicación. Se pueden realizar dos cosas: registrarse o acceder al sistema con un usuario existente. La funcionalidad está recogida en sys\_inicio() {clase Terminal, línea 27}, y una vez accedido al sistema correctamente, irá al menú inicio del cine.



1. **Menú inicio del cine** -> desde aquí podemos realizar 4 acciones: cartelera (podemos ver las películas disponibles y seleccionar una de ellas para ver más información), comprar (nos lleva al proceso de compra, que nos hace seleccionar película, seleccionar un asiento disponible y seleccionar una tarjeta válida para proceder al pago), perfil (nos lleva al menú perfil) y salir del sistema. Corresponde a la función accion\_cine(user\_accedido) {clase Terminal, línea 159}, que recibe el usuario que accedido al sistema, y proporciona las funcionalidades descritas anteriormente.



1. **Menú perfil** -> las funcionalidades que recoge este menú van asociadas a la interacción con la información de un perfil, que en este caso puede ser guardar una tarjeta, borrar una tarjeta, cambiar la contraseña del usuario, ver las entradas compradas, ver las tarjetas guardadas del usuario y salir (volver al menú inicio del cine). La función que se encarga de realizar las funcionalidades de este menú es acc\_perfil(user\_accedido) {clase Terminal, línea 325}, que también necesita el user\_accedido.



Por lo tanto, estos son los 3 menús principales de la aplicación, el primero por ser el menú que sirve de entrada a la aplicación, y los otros 2 porque después de realizar cada acción se va a regresar a uno de ellos para seguir con la ejecución. Cada acción de las expresadas en cada menú va a llevar a cabo la creación de un menú de selección o de visualización, dependiendo de la acción escogida. Además, cada acción solicitada por input o valor dado tiene que ser válido, o de lo contrario el sistema no avanzará hasta que se haya introducido un input correcto.

## **MEJORAS IMPLEMENTADAS**

Entre todas las mejoras opcionales que se proponían en el enunciado, nosotros hemos decidido implementar 2 de ellas. Estas son las siguientes:

* **Implementación de base de datos:** para guardar los datos, hemos usado sqlite3 en Python, que nos permite crear bases de datos y gestionarlas para añadir los datos que querramos. Para ver la información de la base de datos, hemos usado DB Browser. La base de datos se corresponde con la clase Database, que contiene todas las acciones con querys SQL para realizar funcionalidades en cada caso. De esta manera, podemos tener los datos mejor organizados y con un formato adecuado, además de poder acceder fácilmente a nuestros datos mediante sentencias SQL.
* **Rotación de claves:** se ha decidido implementar la rotación de claves de dos maneras. La primera consiste en que cada vez que se realiza un acceso, se va a generar el cifrado de la contraseña con un salt diferente, de manera que el cifrado va cambiando (función rotación\_claves(user\_accedido, False, None) y así solo cambia el salt). La segunda manera consiste en la funcionalidad del menú perfil “cambiar contraseña”, que pedirá una nueva contraseña y generará un hash nuevo con un salt nuevo (función rotación\_claves(user\_accedido, True, contraseña\_antigua)). En estos dos casos, en rotación\_claves() {clase Terminal, línea 574}, se tendrán que descifrar las tarjetas con la clave antigua y cifrarlas de nuevo con la nueva clave generada, para que así los datos se conserven correctamente.

# **FIRMA DIGITAL**

La firma digital nos va a aportar dentro de nuestra aplicación autenticidad respecto a un mensaje. De esta manera, conseguimos verificar que un mensaje proviene de una persona concreta correctamente. Para realizar todo esto, primero hemos tenido que decidir dos cosas importantes: cómo gestionar las claves asimétricas que se utilizarán en este proceso, y cómo implementar la firma de manera que tenga utilidad. Además, es necesario decidir el algoritmo que se va a usar. En nuestro caso, el algoritmo usado es RSA, en el que se necesitará una clave privada y pública por cada usuario para que cada uno pueda firmar (con clave privada del emisor) y verificar (con clave pública del emisor) en su respectivo momento. A continuación, se van a explicar ambas cosas detalladamente.

* Gestión de claves asimétricas:

# **CERTIFICADOS DE CLAVE PÚBLICA**

# **COMPLEJIDAD DE LA APLICACIÓN**