Documentación de la Aplicación de Seguridad para el Hospital Gregorio Marañón

Pablo Moreno González - 100451061
Zhiyuan Guan Huang - 100495756

Evaluable 2

1. Propósito y Estructura Interna de la Aplicación

Propósito de la Aplicación

Sec Hosp (Security Hospital) es nuestra app de seguridad para el Hospital Gregorio Marañón. Tiene como objetivo permitir que el personal autorizado, como enfermeros, médicos y administradores, acceda de manera segura a los registros de los pacientes y realice acciones necesarias para su cuidado, tales como recetar medicamentos o actualizar información médica. Para proteger la confidencialidad de esta información sensible, todos los datos se cifran de forma segura, y únicamente el personal autorizado conoce las claves necesarias para descifrar esta información.
Además, en el caso de los médicos, se añade un nivel adicional de autenticidad mediante códigos de autenticación de mensajes (MAC) para registrar cada acción médica como evidencia irrefutable de que el mensaje proviene de una fuente confiable. Esto asegura que la información registrada por los médicos sea confiable y no haya sido alterada.

¿Para qué utiliza la firma digital?

Uso de la Firma Digital

La firma digital se utiliza para garantizar que las recetas médicas sean emitidas únicamente por médicos autorizados y que no hayan sido alteradas por terceros. Esto protege la integridad y autenticidad de los documentos críticos dentro del sistema hospitalario.

¿Qué algoritmos ha utilizado y por qué?

Algoritmos Usados

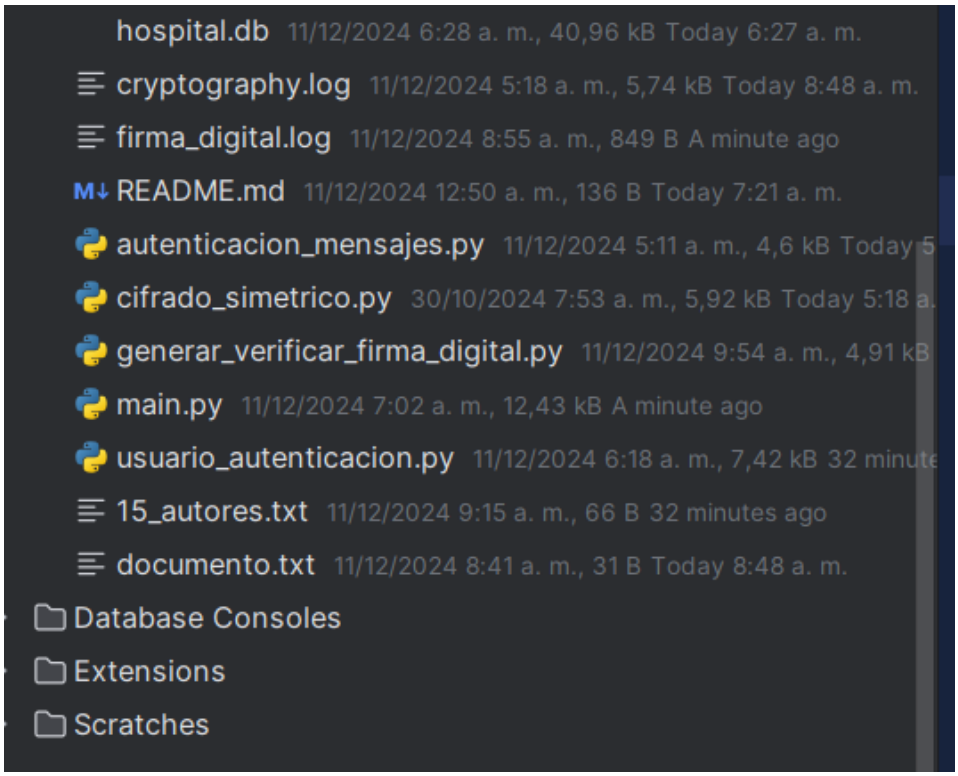
Se ha utilizado el algoritmo RSA (Rivest-Shamir-Adleman) con claves de 2048 bits para la firma y verificación, debido a su alta seguridad y amplia aceptación en sistemas críticos. Para el cálculo del hash, se empleó SHA-256, ya que proporciona un resumen único y seguro para los datos, resistente a colisiones.

¿Cómo se gestionan y almacenan las claves y las firmas?

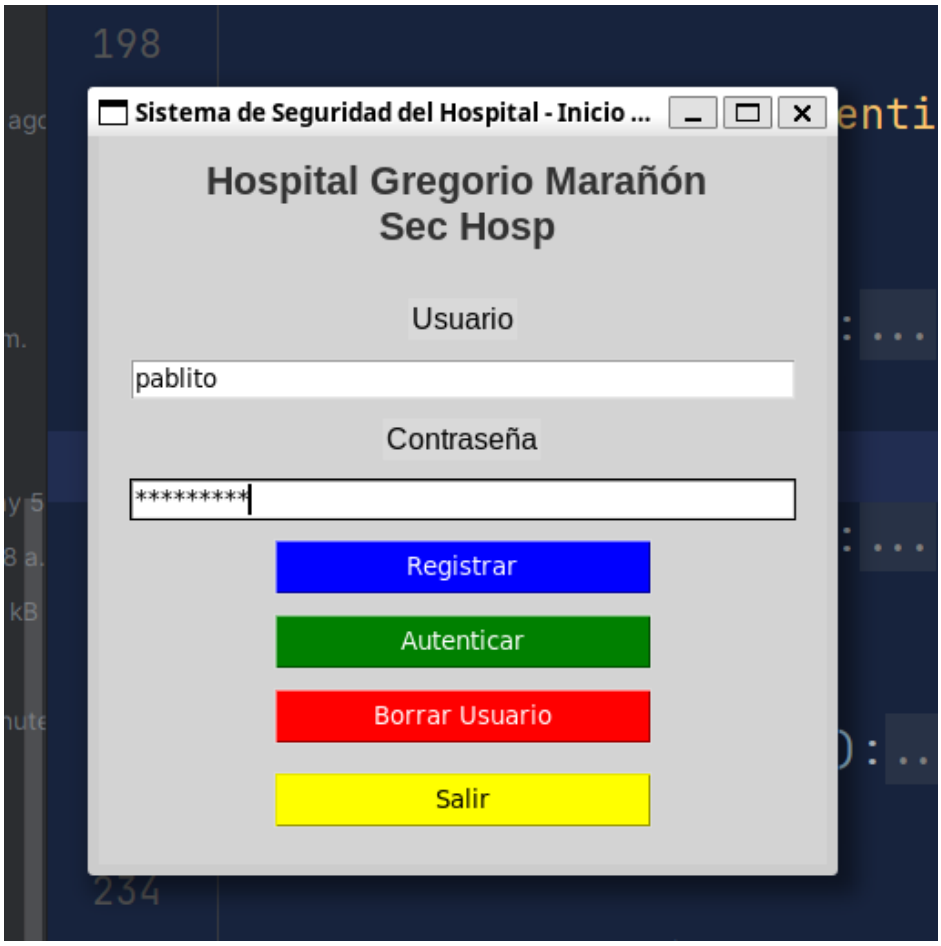
Gestión de Claves y Firmas

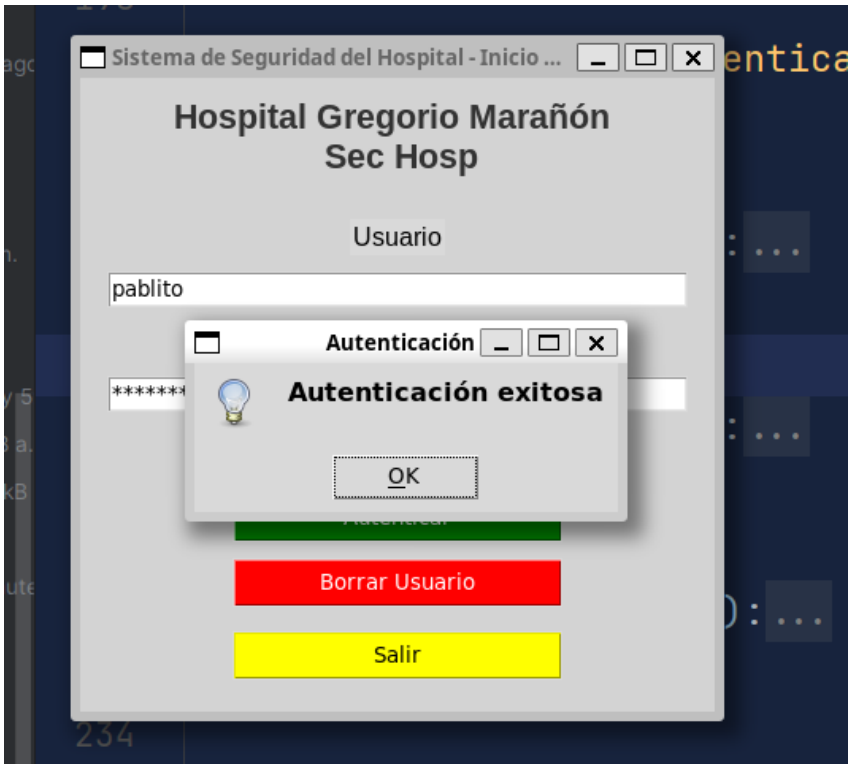
Las claves se generan y almacenan en formato PEM dentro de directorios separados para cada usuario, protegidas con contraseñas fuertes. Las firmas digitales se generan utilizando las claves privadas y se almacenan junto a los archivos correspondientes con extensión ".sig". Además, se lleva un historial de actividades en el archivo de log *firma_digital.log*, que se actualiza continuamente. Como mejora adicional, se ha implementado una interfaz gráfica para que el proceso sea más visual y accesible para los usuarios.

Archivos iniciales antes de compilar mi programa

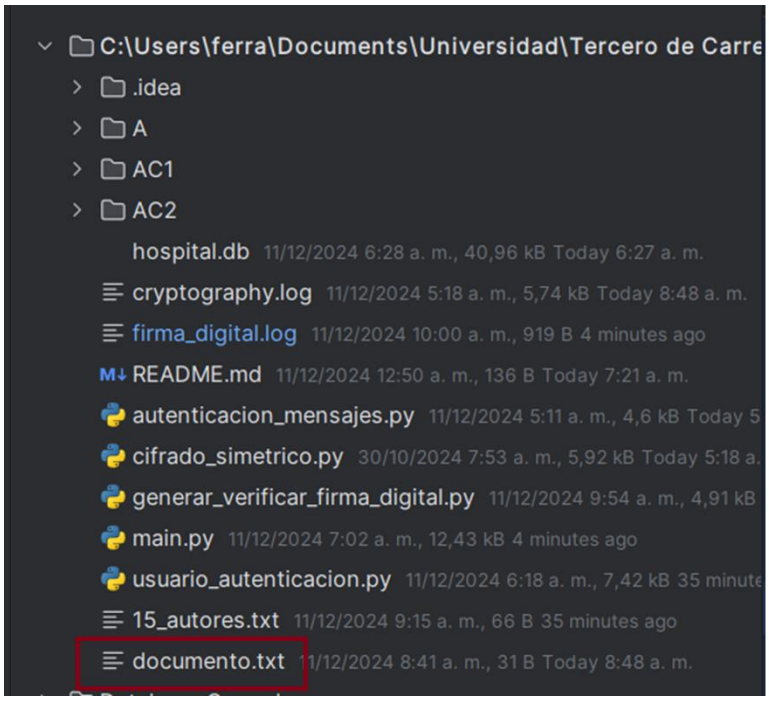


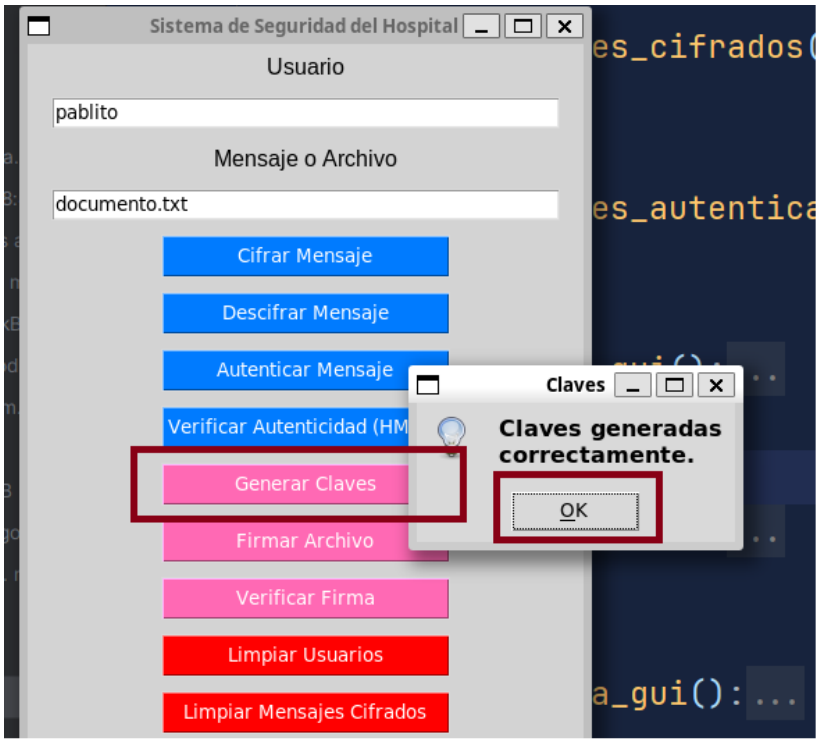
Iniciamos sesión (autenticamos usuario y contraseña)



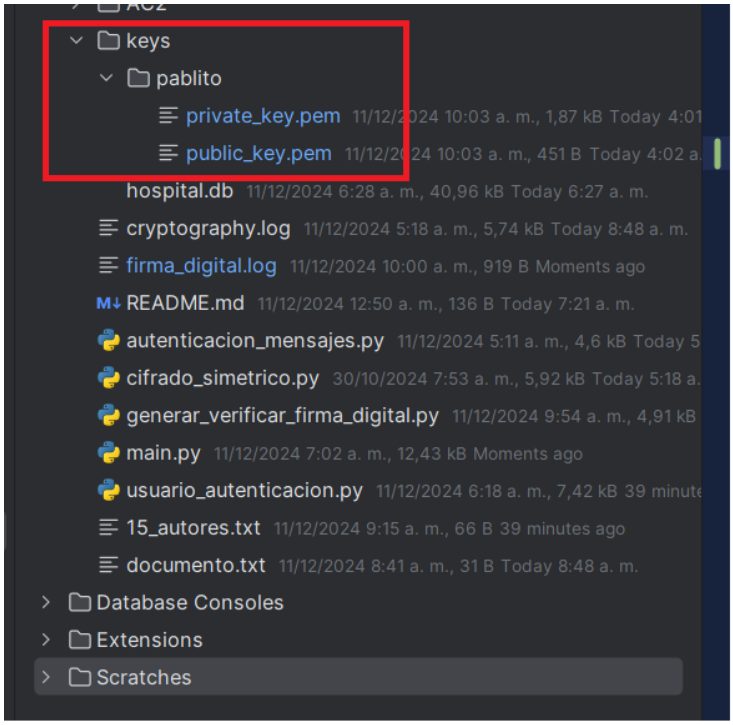


El evaluable 2 son los botones de color rosa, asi que introduzco la ruta del archivo que quiero firmar digitalmente (documento.txt)

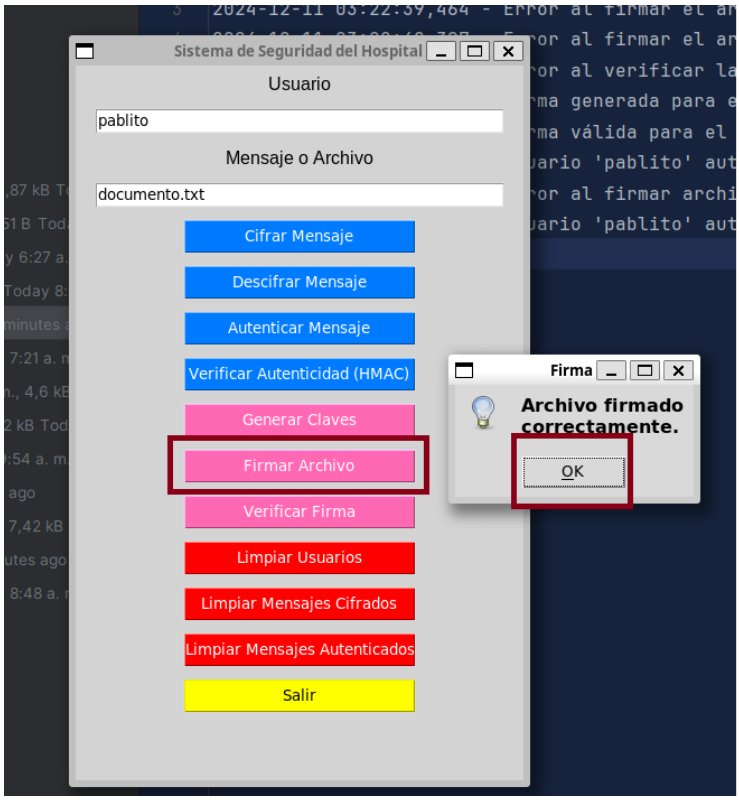




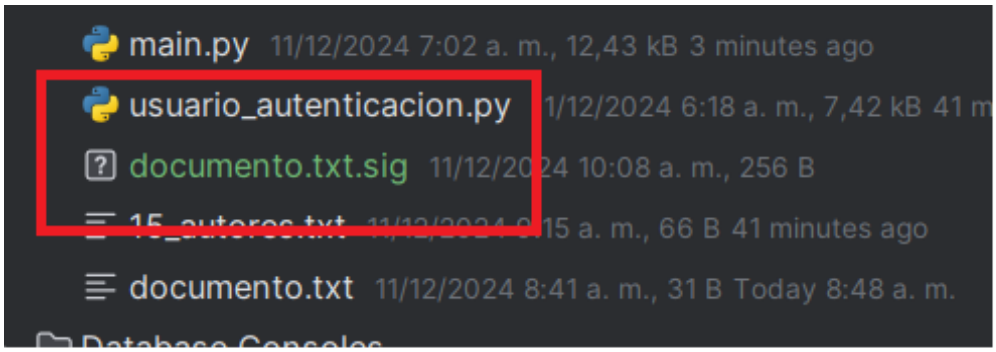
Se genera claves publicas y privadas en la carpeta keys, con el usuario ingresado (Pablito)



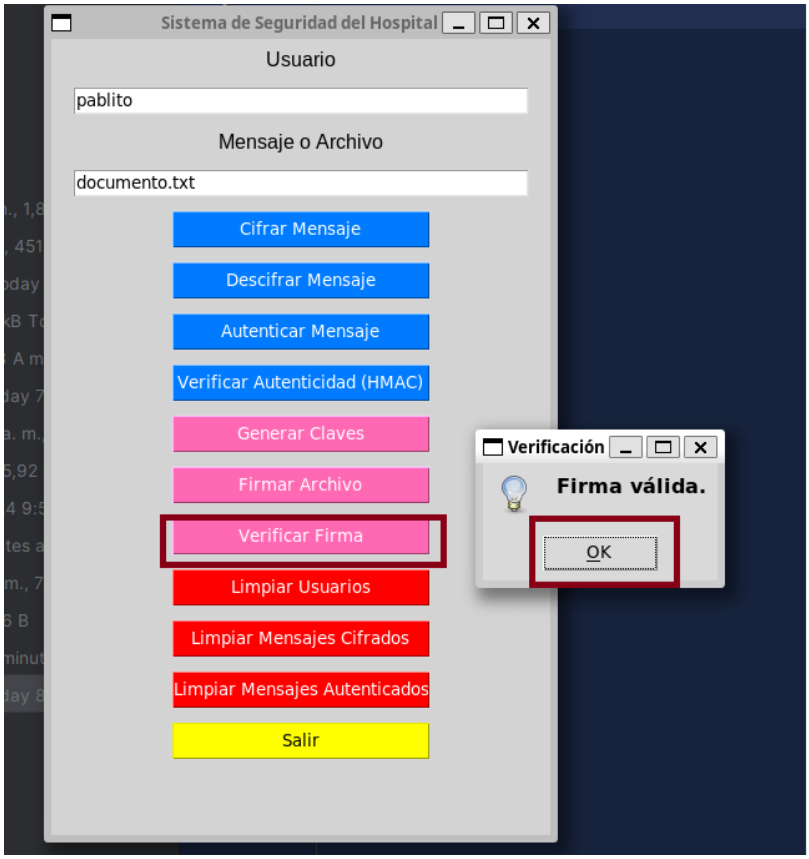
Vamos a firmar el documento.txt que se encuentra en la carpeta principal



Se genera la firma del documento txt (sig es de signed)



Ahora podemos verificar que hemos firmado



Roles en la Jerarquía de Certificación

Relación de Roles en un Hospital

En el contexto del hospital:

- **AC1 (Autoridad de Certificación Raíz):** Representa al *Departamento de TI Central del Hospital*, encargado de establecer la infraestructura de seguridad inicial y firmar el certificado de AC2. Es la entidad más confiable del sistema y se utiliza exclusivamente para tareas críticas.
- **AC2 (Autoridad de Certificación Subordinada):** Actúa como el *Departamento de Seguridad de Datos del Hospital*, que emite certificados para usuarios finales (médicos, enfermeros) y sistemas internos. Su responsabilidad es asegurar el funcionamiento seguro en las operaciones diarias.
- **A (Entidad Usuaria):** Corresponde a los *Médicos del Hospital*, quienes utilizan sus certificados y claves privadas para firmar digitalmente las recetas médicas, asegurando su autenticidad e integridad.

Esta jerarquía garantiza un flujo controlado de confianza y delegación de responsabilidades, adaptándose a las necesidades del entorno hospitalario.

¿Para qué utiliza la firma digital?

Uso de la Firma Digital

El propósito principal de la firma digital en este proyecto es garantizar que solo los médicos autorizados puedan emitir recetas médicas y asegurar que estas no sean alteradas por terceros. Esto protege tanto la integridad como la autenticidad de las recetas, promoviendo la confianza en el sistema de gestión hospitalaria.

¿Qué algoritmos ha utilizado y por qué?

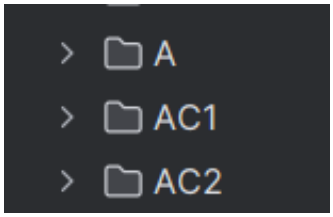
Algoritmos Utilizados

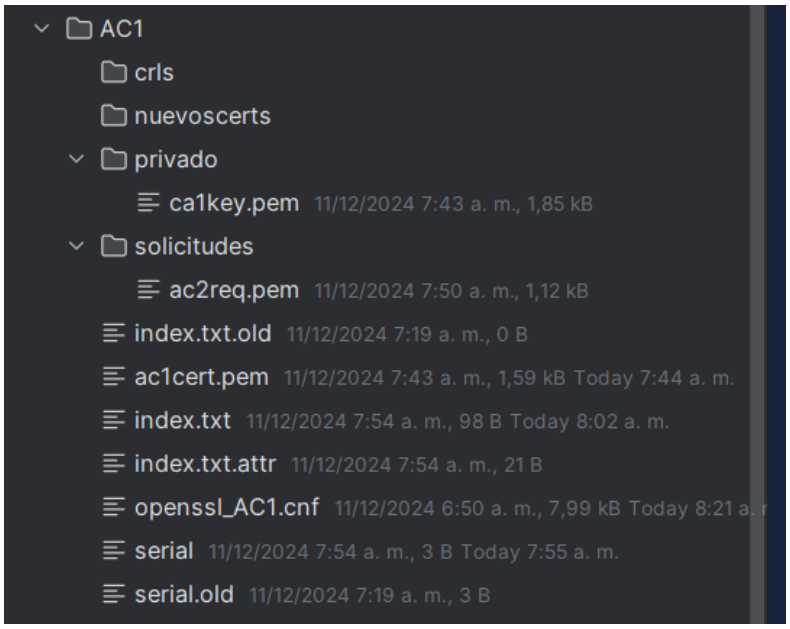
Se utiliza el algoritmo RSA para la generación de claves públicas y privadas, junto con SHA-256 como función de resumen criptográfico para generar el hash de los documentos. RSA es elegido por su robustez y amplia aceptación en sistemas de firma digital, mientras que SHA-256 asegura un alto nivel de seguridad al producir resúmenes únicos para los documentos.

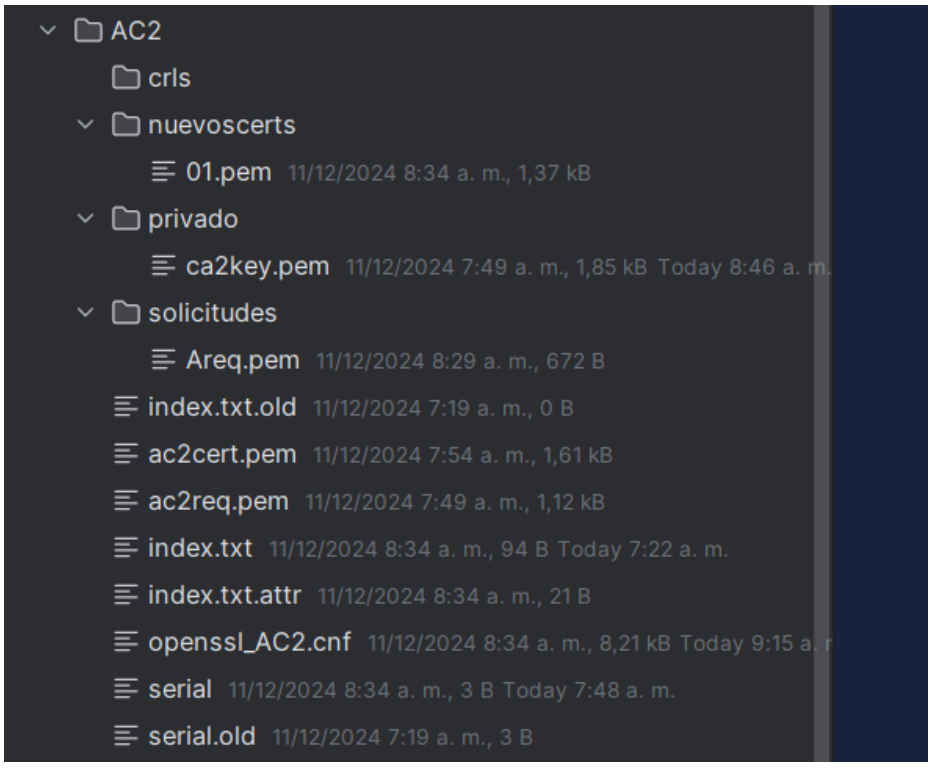
¿Cómo se gestionan y almacenan las claves y las firmas?

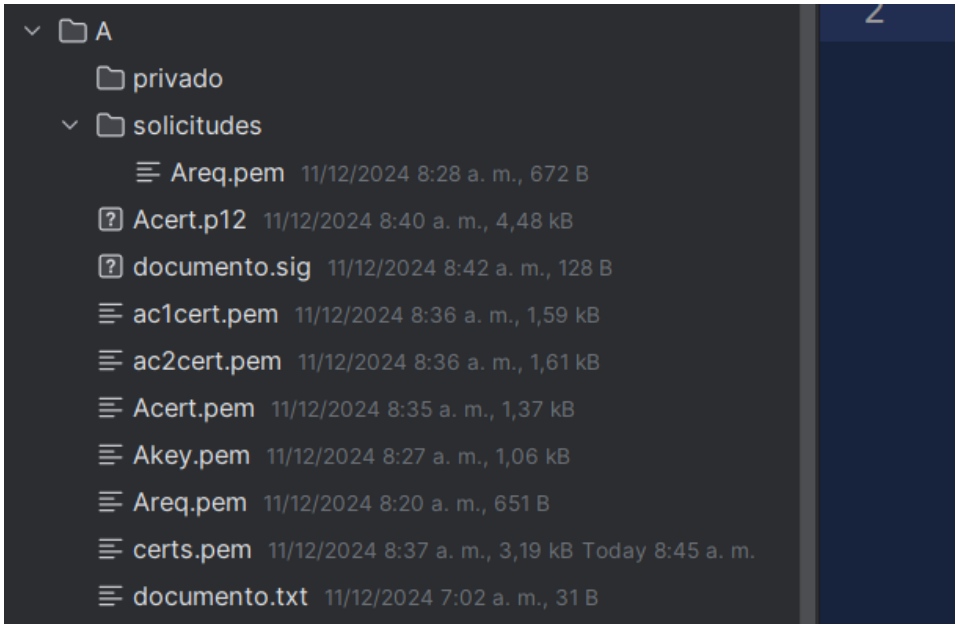
Gestión de Claves y Firmas

Las claves se generan y almacenan en formato PEM dentro de directorios separados para cada usuario, protegidas con contraseñas fuertes. Las firmas digitales se generan utilizando las claves privadas y se almacenan junto a los archivos correspondientes con extensión ".sig".


A
AC1
AC2


AC1
crls
nuevoscerts
privado
ca1key.pem 11/12/2024 7:43 a. m., 1,85 kB
solicitudes
ac2req.pem 11/12/2024 7:50 a. m., 1,12 kB
index.txt.old 11/12/2024 7:19 a. m., 0 B
ac1cert.pem 11/12/2024 7:43 a. m., 1,59 kB Today 7:44 a. m.
index.txt 11/12/2024 7:54 a. m., 98 B Today 8:02 a. m.
index.txt.attr 11/12/2024 7:54 a. m., 21 B
openssl_AC1.cnf 11/12/2024 6:50 a. m., 7,99 kB Today 8:21 a.
serial 11/12/2024 7:54 a. m., 3 B Today 7:55 a. m.
serial.old 11/12/2024 7:19 a. m., 3 B


AC2
crls
nuevoscerts
01.pem 11/12/2024 8:34 a. m., 1,37 kB
privado
ca2key.pem 11/12/2024 7:49 a. m., 1,85 kB Today 8:46 a. m.
solicitudes
Areq.pem 11/12/2024 8:29 a. m., 672 B
index.txt.old 11/12/2024 7:19 a. m., 0 B
ac2cert.pem 11/12/2024 7:54 a. m., 1,61 kB
ac2req.pem 11/12/2024 7:49 a. m., 1,12 kB
index.txt 11/12/2024 8:34 a. m., 94 B Today 7:22 a. m.
index.txt.attr 11/12/2024 8:34 a. m., 21 B
openssl_AC2.cnf 11/12/2024 8:34 a. m., 8,21 kB Today 9:15 a.
serial 11/12/2024 8:34 a. m., 3 B Today 7:48 a. m.
serial.old 11/12/2024 7:19 a. m., 3 B



Configuración de AC2 (Autoridad de Certificación Subordinada)

```
# Crear la estructura de directorios
mkdir crls nuevoscerts privado solicitudes

# Inicializar archivos necesarios
echo '01' > serial
> index.txt

# Generar el par de claves y la solicitud de certificado
openssl req -newkey rsa:2048 -days 3650 -keyout privado/ca2key.pem \
    -out solicitudes/ac2req.pem -config openssl_AC2.cnf

# Firmar la solicitud de AC2 desde AC1
cd ../AC1
openssl ca -in solicitudes/ac2req.pem -notext -extensions v3_subca \
    -config openssl_AC1.cnf

# Copiar el certificado firmado a AC2
cp nuevoscerts/01.pem ../AC2/ac2cert.pem
```

Configuración de A (Entidad Usuaria)

```
# Crear la estructura de directorios
mkdir solicitudes

# Generar el par de claves y la solicitud de certificado
openssl req -newkey rsa:1024 -days 365 -keyout Akey.pem \
    -out solicitudes/Areq.pem

# Firmar la solicitud de A desde AC2
cd ../AC2
openssl ca -in solicitudes/Areq.pem -notext -config openssl_AC2.cnf

# Copiar el certificado firmado a la carpeta de A
cp nuevoscerts/01.pem ../A/Acert.pem
```

Verificación y Exportación

```
# Verificar el certificado de A
cd ../A
cp ../AC1/ac1cert.pem ./
cp ../AC2/ac2cert.pem ./
cat ac1cert.pem ac2cert.pem > certs.pem
openssl verify -CAfile certs.pem Acert.pem

# Exportar el certificado al formato PKCS12 (opcional)
openssl pkcs12 -export -in Acert.pem -inkey Akey.pem -certfile certs.pem \
    -out Acert.p12
```

Descripción de las Mejoras

Validación de Contraseñas

Validación de Contraseñas

Se incluyó una validación de contraseñas que asegura que tengan al menos 8 caracteres y contengan tanto letras como números, fortaleciendo la seguridad de las claves privadas.

Interfaz Gráfica Mejorada

Interfaz Amigable con Tkinter

Se diseñó una interfaz gráfica utilizando Tkinter, con colores y un estilo intuitivo para facilitar su uso por el personal del hospital. Esto permite gestionar claves, firmar y verificar documentos de manera más visual.

Botones Extras

Botones Extras en la Interfaz

Se añadieron botones adicionales como "Borrar Usuarios" y "Limpiar Datos" en la interfaz para facilitar la gestión de la base de datos y eliminar información no deseada.

Sistema de Log

Historial de Actividades con Log

Se implementó un sistema de log que registra todas las acciones realizadas, como la generación de claves, firmas y verificaciones. Esto asegura la trazabilidad de las operaciones y facilita la auditoría.

Base de Datos para Usuarios

Base de Datos para Usuarios

Se desarrolló una base de datos para almacenar información de los usuarios y sus claves de forma estructurada y segura. Esto facilita la administración de las claves y asegura que solo usuarios autorizados puedan acceder al sistema.

Fuentes Externas Utilizadas

Fuentes Externas

- **Cryptography.io:** Utilizado para la implementación de métodos de generación de claves RSA, así como para la firma y verificación de documentos.
- **Material OCW UC3M:** Referencia principal para comprender y llevar a cabo la parte de certificados digitales y su implementación en el sistema.