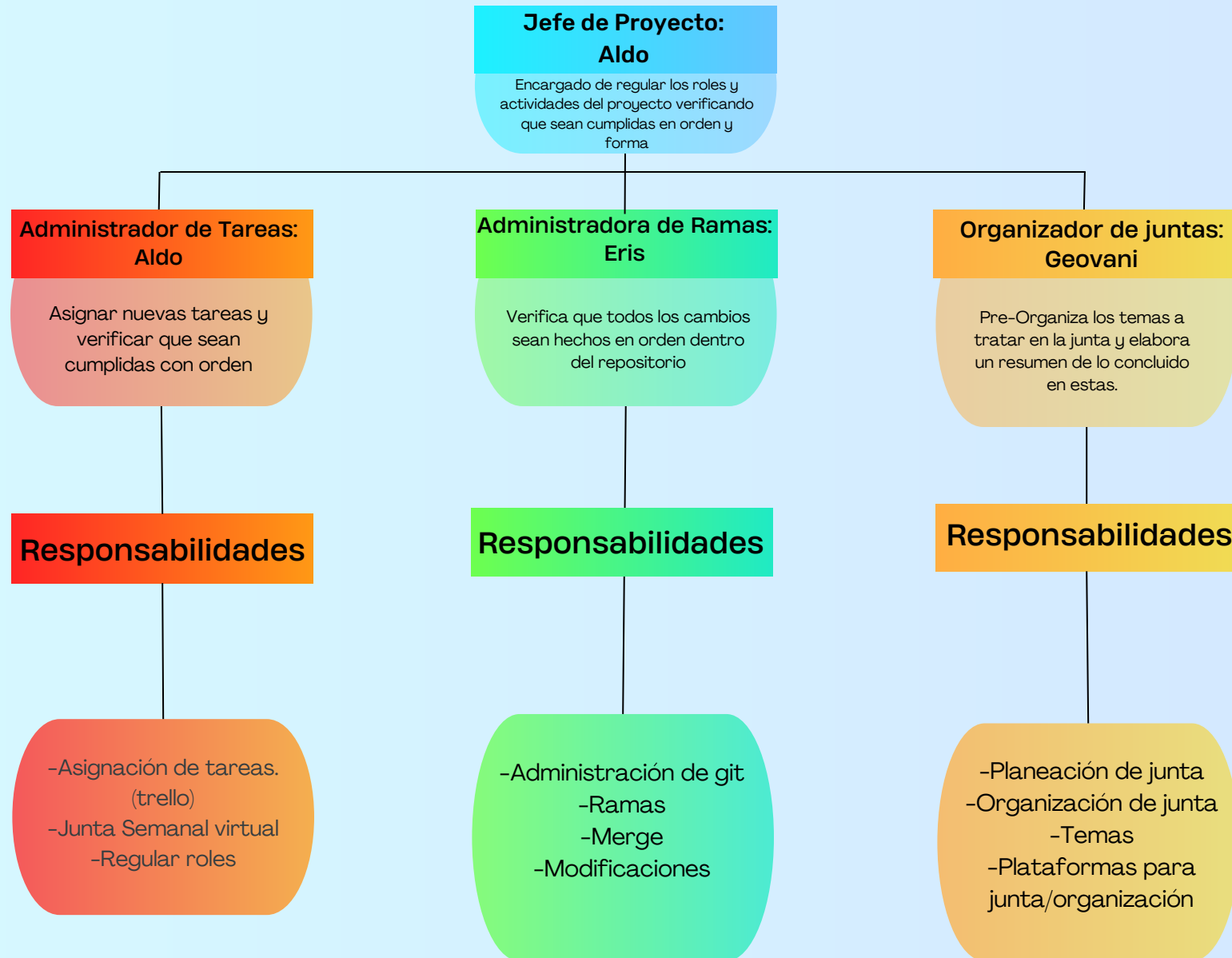


SHAMIR SECRET SHARING



REUNIÓN MARTES 05 DE DICIEMBRE

Primera reunión del proyecto llevada a cabo el 5 de diciembre de 2023 en Discord con la participación de Aldo, Eris y Geovani:

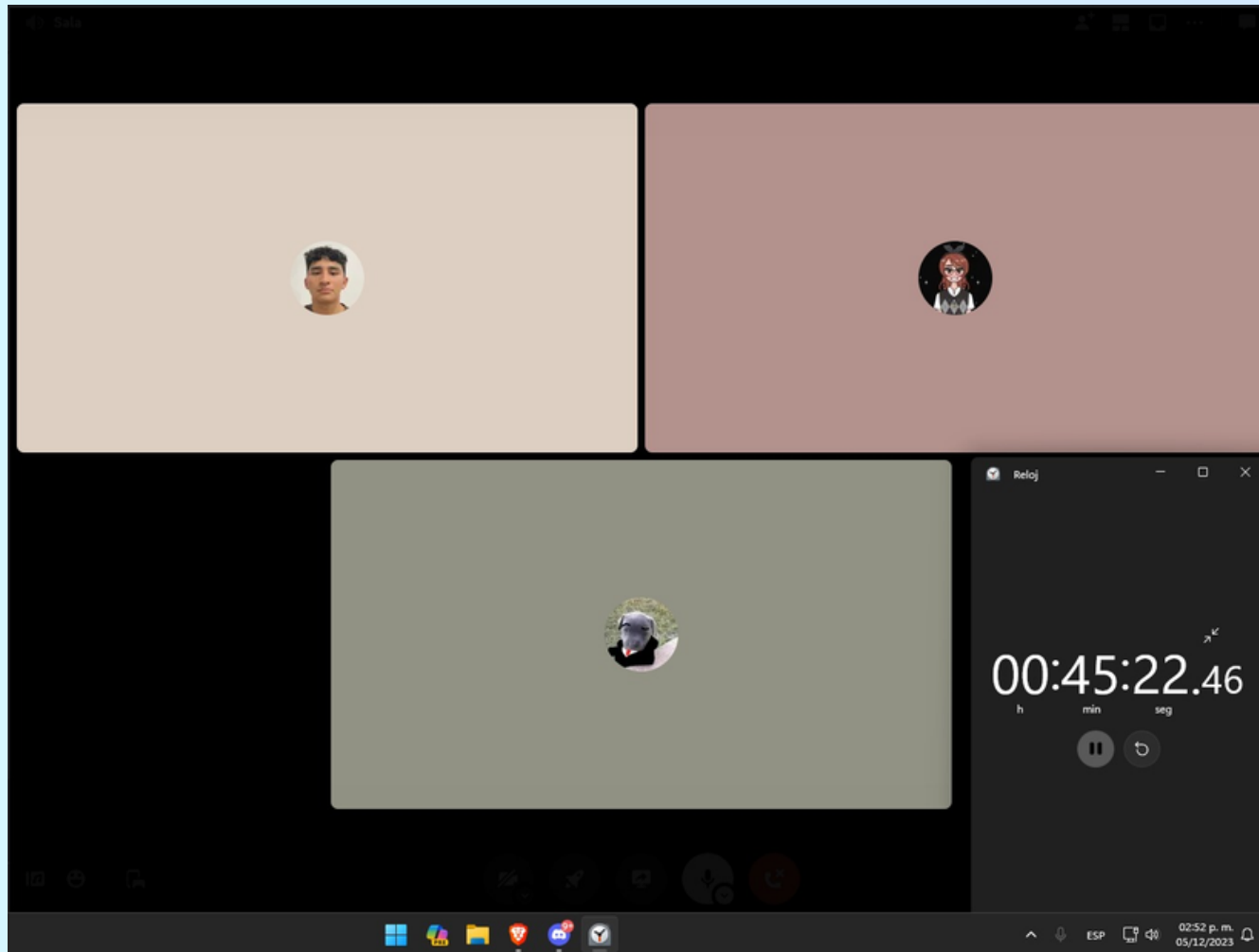
En la fase inicial de la reunión, nos propusimos aclarar cualquier duda surgida tras la lectura del proyecto y la revisión de los videos proporcionados. Se identificaron varias interrogantes, lo que generó una discusión sobre la mejor manera de abordar el proyecto. Inicialmente, consideramos utilizar Python, pero después de reflexionar, decidimos regresar a Java, ya que sentimos que era más coherente con nuestras habilidades y experiencia previa.

Una vez establecida la elección del lenguaje de programación, comenzamos a esbozar el plan de acción. En este punto, reconocimos que la implementación del polinomio era un componente clave del proyecto y decidimos empezar por esa parte. Aldo compartió su comprensión detallada de esta sección y propuso un esquema para su funcionamiento. Esta explicación permitió a Eris y a mí entender mejor el enfoque necesario.

Dada la claridad obtenida, Aldo se ofreció a liderar el desarrollo de esta parte, ya que se sentía cómodo con la implementación propuesta. Por otra parte Eris y Geovani empezaron a ver como debería de funcionar la parte de codificar algo, pues sabíamos algunas formas pero no la que pedía el proyecto. Esa fue nuestra tarea, investigar como tendría que funcionar la codificación para poder llevarla a cabo.

Con esto termino la primera reunión y fue un gran avance pero sabíamos que teníamos que apresurarnos demasiado. :)

REUNIÓN MARTES 5 DE DICIEMBRE



REUNIÓN MIERCOLES 06 DE DICIEMBRE

Segunda reunión del proyecto el 06 de Diciembre de 2023 en Discord con Aldo, Eris y Geovani:

En esta segunda reunión, nos enfocamos en revisar los avances realizados desde la última vez y compartir nuestras experiencias individuales con las respectivas implementaciones. Aldo lideró la presentación de su trabajo inicial en la implementación del polinomio, mientras Eris y Geovani compartimos nuestras experiencias y comprensiones respecto al cifrado.

Aldo nos detalló cómo había abordado la implementación del polinomio, proporcionando ejemplos concretos de código y explicaciones sobre su lógica. En este punto, ya teníamos todas las clases necesarias creadas, pero aún se requería integrarlas para lograr un funcionamiento coherente del programa.

Dado que teníamos código en las clases de polinomio y cifrado, pero aún no habíamos establecido cómo interactuarían entre sí, decidimos que Aldo compartiera su pantalla. Con la colaboración de todos, Aldo realizó modificaciones en el código en tiempo real, buscando formas de unir las clases de manera efectiva y eficiente.

En particular, dedicamos tiempo a investigar el uso de SHA256 para cifrar una clave proporcionada por el usuario. Exploramos diversas implementaciones y discutimos las mejores prácticas para integrar este componente en nuestro proyecto.

- Aldo lideró la tarea de unir las funcionalidades de las clases Polinomio y Cifrado para lograr una implementación coherente.
- Se realizaron ajustes en la llamada a métodos específicos de la clase Polinomio para obtener evaluaciones del polinomio en un punto dado.

También se discutieron estrategias para mejorar la eficiencia del código, especialmente en la generación y evaluación del polinomio.

6 de dic 17:02

Sala | Denle dizcor - Discord

Denle dizcor

Eventos

CANALES DE TEXTO

general

clips-y-destacados

CANALES DE VOZ

Sala

Establece un estado de canal

AldoJurado

erlis2934

Geovani

Voz conectada

Sala | Denle dizcor

No hay eventos

Geovani

PAPICHAMPU

Calendarios

diciembre

	D	L	M	M	J	V	S
48	29	27	26	25	24	1	2
49	3	4	5	6	7	8	9
50	10	11	12	13	14	15	16
51	17	18	19	20	21	22	23
52	24	25	26	27	28	29	30
1	31						

Relojes

01:53:49.4

Pausar

Vuelta

Mundo

Alarmas

Cronometro

Temporizador

REUNIÓN JUEVES 07 DE DICIEMBRE

Tercera reunión del proyecto el 07 de Diciembre de 2023 en Discord con Aldo, Eris y Geovani:

En esta reunión, logramos un avance significativo en nuestro proyecto. Implementamos con éxito la funcionalidad de codificación mediante el uso del algoritmo SHA-256 para procesar las contraseñas ingresadas por el usuario. Esta implementación garantiza una capa adicional de seguridad al convertir las contraseñas en un hash robusto.

Además, completamos la integración de la clase Cifrado AES, que permite cifrar texto utilizando la cifra AES.

Aldo implemento la funcionalidad de cifrado, probándola inicialmente con un texto.

Eris desempeñó un papel importante, pues modifico el cifrado del código para permitir la lectura de un archivo, cifrar su contenido y guardarlo en un archivo nuevo. Esta mejora permitió la manipulación eficiente de datos almacenados en archivos, demostrando la versatilidad y la capacidad de adaptación de nuestro código.

Algo importante durante la reunión fue la capacidad de generar evaluaciones a partir de un polinomio aleatorio. Implementamos el método `evaluas`, que utiliza un polinomio con un grado definido por el parámetro `t`. Este método evalúa el polinomio en puntos específicos y devuelve un arreglo de `BigInteger` que representa las evaluaciones.

Además, Aldo, Eris y Geovani trabajaron en la lógica de manejo de archivos para guardar las evaluaciones generadas y el texto cifrado. Se implementó la escritura de datos en archivos con éxito.

Hasta este momento solo teníamos que modificar algunas cosas para que funcionara el cifrado al 100 pero ya no era nada complicado, así que decidimos empezar ver como implementaríamos el descifrado, por lo que nos propusimos el investigar y compartir nuestras ideas de como hacerlo para la siguiente reunión.

REUNIÓN VIERNES DICIEMBRE 08 DE DICIEMBRE

En la cuarta reunión del proyecto, llevada a cabo el 08 de Diciembre de 2023 a través de Discord con la participación de Aldo, Eris y Geovani, nos centramos en la validación y optimización de nuestro trabajo hasta ese momento, con un enfoque especial en la parte de descifrado.

Lo primero que hicimos este día es revisar detenidamente el código de la clase Descifrado y las interacciones con la clase CifradoAES. Analizamos los procesos de lectura de evaluaciones, la recuperación del secreto y la aplicación del descifrado con el objetivo de identificar posibles áreas de mejora.


Durante la revisión, nos dimos cuenta de que había una oportunidad para optimizar el código y mejorar la eficiencia en la lectura del archivo de evaluaciones. Eris propuso utilizar una estructura de datos más eficiente para almacenar las evaluaciones y simplificar la lógica de recuperación del secreto.


Además, discutimos los problemas que estábamos experimentando con la lectura del archivo cifrado y la generación del archivo descifrado vacío. Geovani sugirió la posibilidad de incorporar mensajes de depuración adicionales en el código para rastrear el flujo de ejecución y encontrar el origen del problema.


En esta reunión, se discutieron los puntos finales sobre la parte de descifrado. Geovani y Aldo se encargaron de esta tarea y lograron que el descifrado estuviera casi al 100% listo. Solo quedó pendiente la realización de los tests, tarea que se dividirían entre Eris y Geovani.

Con esto le daríamos fin al proyecto pues ya estaría completo, salió bien el proyecto, y aunque no lo iniciamos con tanta anticipación pudimos concretar perfectamente el proyecto final :)

Sala

















Reloj


01:12:09.88

h min seg



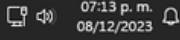






ESP

07:13 p. m.
08/12/2023



DESCRIPCIÓN DEL PROYECTO:

El proyecto consiste en implementar un programa que utilice el esquema de secreto compartido de Shamir para compartir una clave necesaria para descifrar un archivo que contiene información confidencial. El esquema de Shamir se basa en la generación de un polinomio cuyo término independiente es el dato a ocultar. Este polinomio se evalúa en puntos diferentes, y las evaluaciones se distribuyen entre las personas autorizadas. Además, se utiliza un mecanismo de cifrado simétrico (AES) para cifrar el documento claro utilizando una clave generada a partir de una contraseña del usuario.

ENTRADA DEL PROGRAMA:

1. Opción (c/d): Indica si el programa debe cifrar (c) o descifrar (d).
2. Nombre del archivo para evaluaciones del polinomio (c): Archivo donde se guardarán las evaluaciones del polinomio.
3. Número total de evaluaciones ($n > 2$) (c): Cantidad de puntos para evaluar el polinomio.
4. Número mínimo de puntos para descifrar ($1 < t \leq n$) (c/d): Número mínimo de puntos necesarios para descifrar.
5. Nombre del archivo con el documento claro (c): Archivo que contiene la información confidencial.
6. Nombre del archivo con evaluaciones del polinomio (d): Archivo que contiene, al menos, t de las n evaluaciones del polinomio.
7. Nombre del archivo cifrado (d): Archivo cifrado que se debe descifrar.

SALIDA DEL PROGRAMA:

- Cifrar (c):
 - Archivo con el documento cifrado utilizando AES.
 - Archivo con n parejas $(x_i, P(x_i))$ de las evaluaciones del polinomio.
- Descifrar (d):
 - Archivo con el documento claro y con el nombre original.

REQUISITOS FUNCIONALES:

- Generación de Polinomio:
El programa debe generar un polinomio de grado $t-1$ con coeficientes aleatorios y el dato original.
El polinomio debe evaluarse en n puntos diferentes.
- Cifrado y Descifrado:
Utilizar AES para cifrar y descifrar documentos.
Generar la clave de cifrado a partir de la contraseña del usuario mediante SHA-256.
- Entrada de Usuario:
Solicitar al usuario la información necesaria según la opción seleccionada (cifrar o descifrar).

REQUISITOS NO FUNCIONALES:

- Robustez:
El programa debe ser robusto, manejar errores del usuario y condiciones excepcionales sin terminar abruptamente.
- Documentación:
El código debe estar bien documentado utilizando herramientas como javadoc o doxygen.
- Eficiencia:
Se debe prestar atención a la eficiencia del programa, priorizando el funcionamiento correcto y luego la eficiencia.

HERRAMIENTAS:

1. AES, SHA-256:
2. Utilizar bibliotecas de java para implementar el cifrado AES, la función hash SHA-256 y la manipulación de números grandes.
3. Documentación:
4. Emplear herramientas de documentación como javadoc para documentar cada función/módulo del programa.
5. Pruebas Unitarias:
6. Realizar pruebas unitarias para cada módulo del programa.

