Universidad de los Andes
Ingeniería de Sistemas y Computación ISIS 2203 Infraestructura Computacional Semestre 2025-10

Caso de Estudio 3 – Canales Seguros Sistema de rastreo de paquetes en una compañía transportadora

Mateo Parra – 202213933
Juan Felipe Puig – 202221336

I) Descripción de la organización de los archivos en el zip

El archivo sigue la siguiente estructura

```
aerolinea-proyecto/
│
├── client/
│   └── Cliente.java
│   └── ClienteIterativo.java
│   └── ClienteConcurrente.java
│
├── common/
│   └── AESUtils.java
│   └── DHUtils.java
│   └── HMACUtils.java
│   └── LlaveSesionUtils.java
│
├── server/
│   └── ServidorPrincipal.java
│   └── ServidorAuxiliar.java
│
│
├── keys/
│   └── GeneradorLlavesRSA.java
│   └── private.key
│   └── public.key
```

```
│
├── docs/
│   └── Caso3.xslx
    └──Documentacion Caso 3.docx
│
├── README.txt
```

Para correr y ejecutar el proyecto hay que seguir los siguientes pasos de acuerdo con el escenario:

Desde CASO 3\aerolinea-proyecto

Compilar Proyecto

javac common/*.java server/*.java client/*.java

Ejecutar

1. Lanzar servidor -------> java server.ServidorPrincipal (Terminal 1)
2. Lanzar cliente -------> java client.Cliente (Terminal 2)

Escenario 1

1. Lanzar servidor --------> java server.ServidorPrincipal (Terminal 1)
2. Lanzar cliente --------> java client.ClienteIterativo (Terminal 2)

Escenario 2

1. Lanzar servidor --------> java server.ServidorPrincipal (Terminal 1)
2. Lanzar cliente --------> java client.ClienteConcurrente 4/16/32/64 (Terminal 2)

Comparar cifrado simétrico y asimétrico

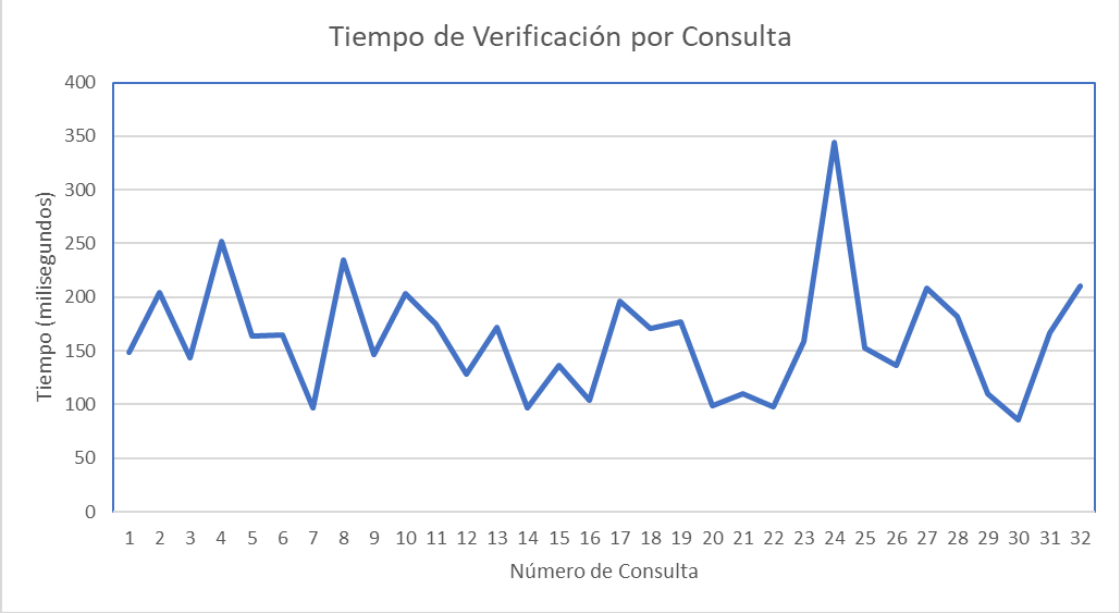
1. Lanzar servidor --------> java server.ServidorAuxiliar (Terminal 1)
2. Lanzar cliente --------> java client.Cliente/java client.ClienteIterativo/java client.ClienteConcurrente (Terminal 2)

II) Datos recopilados de cada Escenario

Escenario 1

Para el escenario 1 se tabularon los tiempos de ejecución en ns para el cifrado, firmado y las 32 consultas secuenciales. (Esta tabla se puede evidenciar en el archivo Caso3.xslx que se encuentra en la carpeta docs/ del proyecto) Para efectos de la organización del informe en este documento solo se visualizara los tiempos promedios y la gráfica asociada a la ejecución del escenario.

Operación	Tiempo (ns)
Cifrado Tabla	516900
Firma tabla	2726900
Verificación Promedio	161793,75



El tiempo promedio de verificación fue de aproximadamente 161,794 ns, mostrando una distribución de tiempos relativamente estable con ligeras variaciones. Por otra parte, el tiempo de firmado de la tabla fue el más costoso en términos de ejecución, superando ampliamente al tiempo de cifrado de la tabla y al tiempo promedio de verificación, confirmando que las operaciones de firma digital requieren un procesamiento computacional significativamente mayor en comparación con las operaciones de cifrado simétrico realizadas con AES.

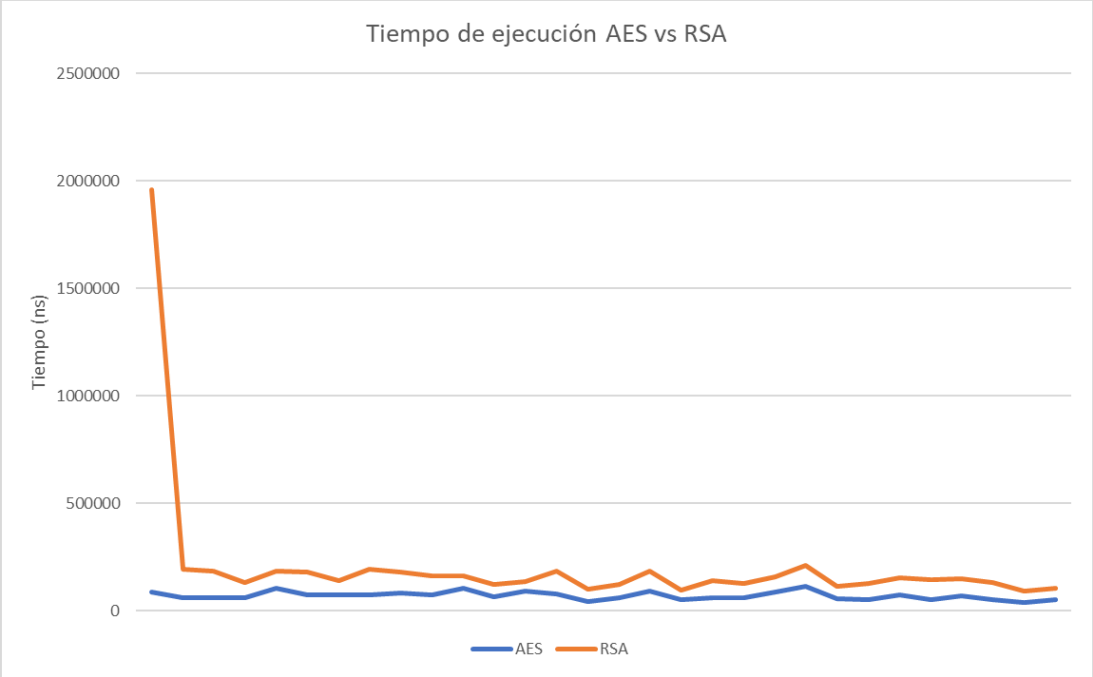
Escenario 2

Para el Escenario 2 se realizaron múltiples ejecuciones para cada una de las opciones de concurrencia (4, 16, 32 y 64 clientes). Se observó una discrepancia mínima entre los resultados obtenidos en cada ejecución, por lo cual se tomó una ejecución representativa a partir de la cual se calcularon los valores promedios utilizados en el análisis. La tabulación detallada de los datos de ejecución del Escenario 2 se encuentra en el archivo Caso3.xlsx, ubicado en la carpeta docs/ del proyecto. Para favorecer la organización del informe, en este documento se

analizarán únicamente los resultados promedios, mientras que las tablas completas podrán ser consultadas en el archivo Excel.

A medida que se incrementa la cantidad de clientes concurrentes, se observa que los tiempos promedio de cifrado, firmado y verificación tienden a estabilizarse, con solo ligeras variaciones entre las diferentes configuraciones. El servidor mantiene un desempeño eficiente incluso bajo cargas mayores, lo que refleja su capacidad de escalabilidad. Esto sugiere que el proceso de firmado es más sensible a la concurrencia, mientras que el cifrado y la verificación muestran una robustez mayor frente al aumento de conexiones simultáneas.

III) Comparación entre cifrado Simétrico y Asimétrico



Algoritmo	Tiempo (ns)
Promedio AES	70253,3

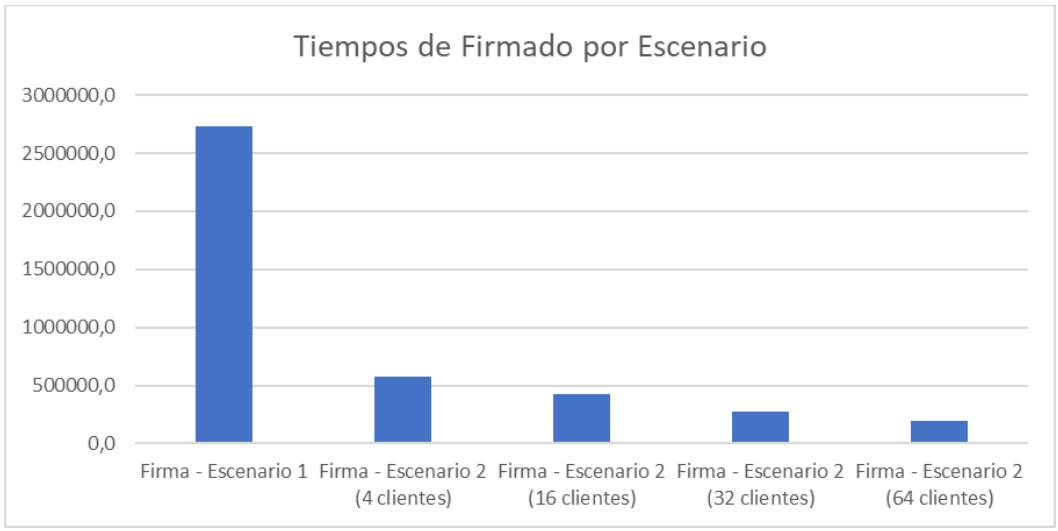
Promedio RSA	78937,9

En la comparación de tiempos de ejecución entre los cifrados simétrico (AES) y asimétrico (RSA), se observa que AES presenta consistentemente tiempos de ejecución inferiores a los de RSA. El promedio de cifrado de AES fue de aproximadamente 70253,3 ns, mientras que RSA alcanzó un promedio significativamente mayor de 78937,9 ns. Esta diferencia se explica porque el cifrado asimétrico implica operaciones matemáticas más complejas, mientras que el cifrado simétrico como AES es computacionalmente más eficiente. La gráfica también muestra que, pese a algunas fluctuaciones, RSA mantiene un desempeño sistemáticamente más costoso en tiempo que AES, confirmando su naturaleza menos adecuada para operaciones de cifrado de grandes volúmenes de datos. Adicionalmente, se ve que el primer valor de la ejecución del RSA es extremadamente alto, esto puede estar relacionado a la maquina en la cual se realizaron las pruebas y or ende se decidio desafectar este valor para el cálculo del promedio.

IV) Análisis de Graficas

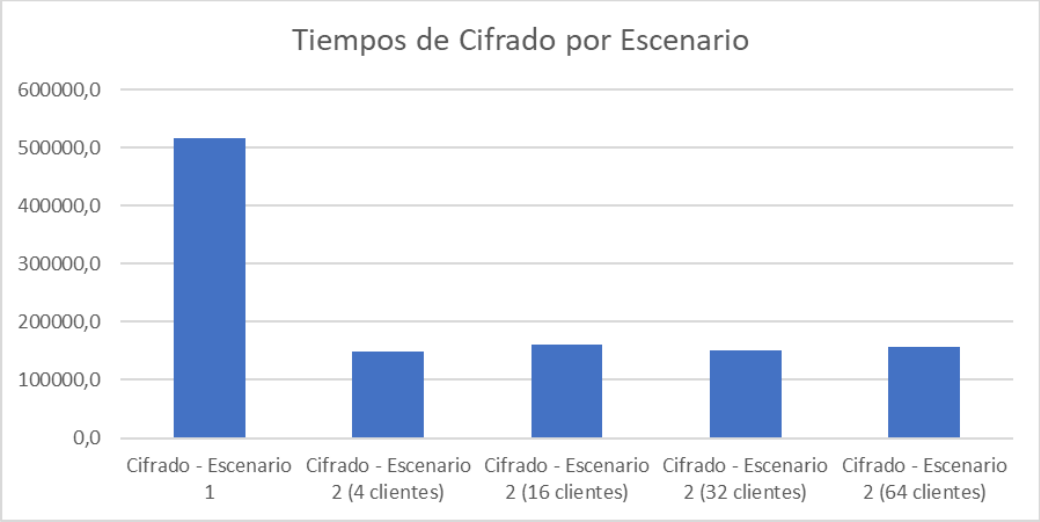
Las gráficas y su debida construcción se encuentran en el archivo “Caso3.xslx” adjunto en la carpeta docs del caso, en este documento se realizará el respectivo análisis, favoreciendo la organización del informe.

- **Grafica “Tiempos para firmar en los escenarios”**



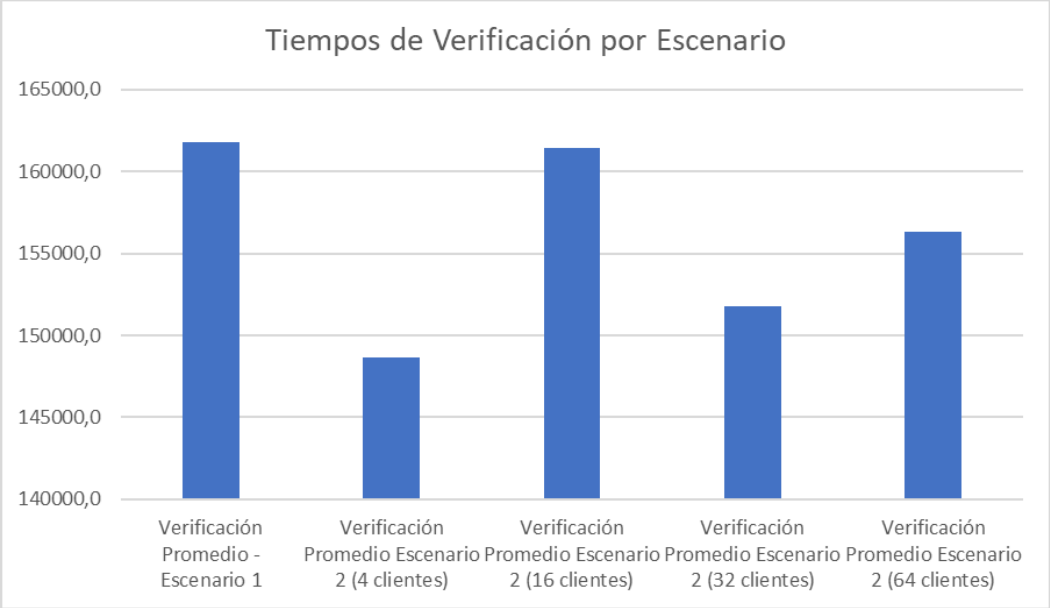
La gráfica muestra los tiempos de firmado en diferentes escenarios de ejecución. Se observa que el Escenario 1 de ejecución iterativa presenta un tiempo de firmado muy superior al de los escenarios concurrentes. A medida que aumenta el número de clientes concurrentes desde 4 a 64, el tiempo promedio de firmado disminuye progresivamente, reflejando una mejor distribución de la carga y aprovechamiento de recursos del servidor. Esto evidencia que la concurrencia favorece una ejecución más eficiente en las operaciones de firmado.

- **Grafica “Tiempos para cifrar en los escenarios”**



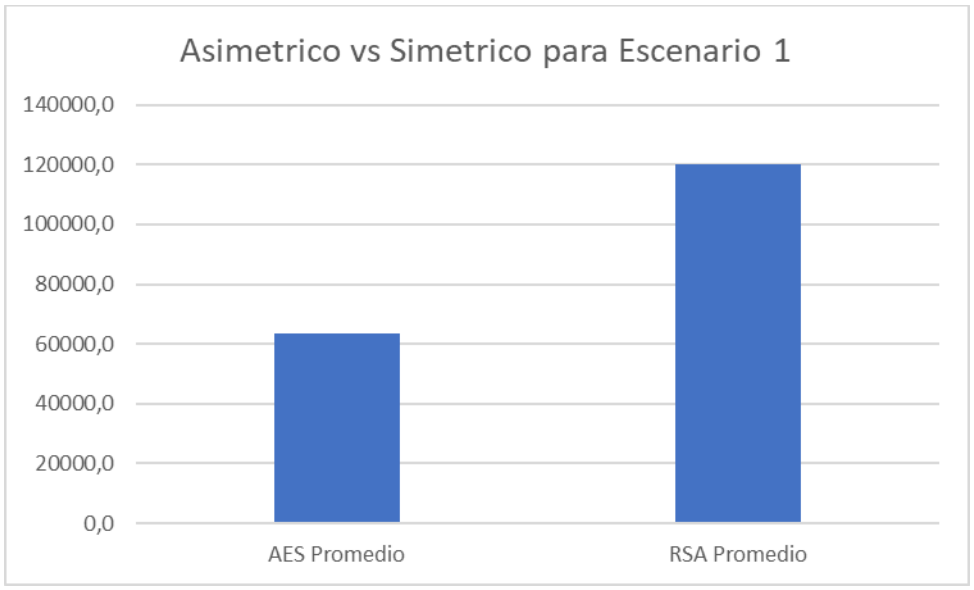
La gráfica muestra los tiempos de cifrado de la tabla de servicios en los diferentes escenarios. Se observa que el Escenario 1 presenta el tiempo de cifrado más alto, debido a su ejecución secuencial. En los escenarios concurrentes, los tiempos de cifrado son considerablemente menores y se mantienen relativamente estables, acá se vuelve a probar que la concurrencia permite un manejo más eficiente de las operaciones de cifrado en el servidor.

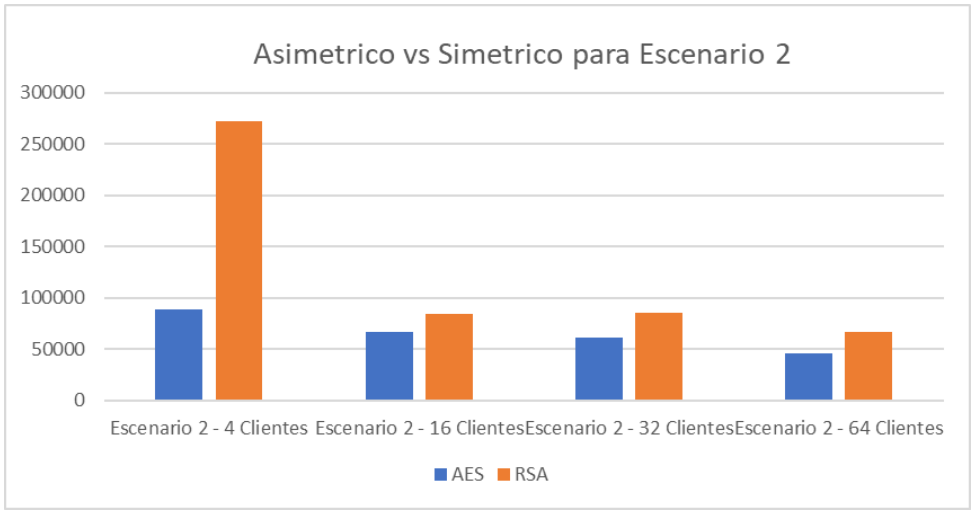
- **Grafica “Tiempos de Verificación en los escenarios”**



La gráfica compara los tiempos promedio de verificación de consultas en los diferentes escenarios. El Escenario 1 y el Escenario 2 con 16 clientes presentan los tiempos de verificación más altos, mientras que el Escenario 2 con 4 clientes muestra el tiempo más bajo. A medida que aumenta el número de clientes concurrentes, los tiempos de verificación tienden a estabilizarse, reflejando que el servidor puede manejar múltiples verificaciones de forma relativamente eficiente, aunque con ligeras variaciones según la carga.

- **Graficas "Simétrico vs Asimétrico en los Escenarios"**





Las gráficas comparan los tiempos de ejecución promedio entre el cifrado simétrico (AES) y asimétrico (RSA) en los diferentes escenarios.
En el Escenario 1, el RSA es considerablemente más lento que AES, lo cual

concuerda con las propiedades de ambos algoritmos.
En el Escenario 2, a medida que aumenta la concurrencia (4, 16, 32 y 64 clientes), se mantiene la tendencia de que RSA siempre requiere más tiempo de ejecución que AES.
Los resultados reflejan la eficiencia superior de AES en operaciones de cifrado de datos, mientras que RSA, debido a su naturaleza matemática más compleja, implica mayores tiempos de procesamiento.

V) **Escenario Definido**

Identificación de la velocidad del procesador:

Procesador: Intel Core i5-10400F.

Velocidad del procesador: 2.9 GHz, que equivale a 2.9 * 10^9 ciclos por segundo.

A continuación, se realiza un desglose de cada tipo de cifrado, con estimaciones de ciclos basadas en las características de estos algoritmos y las operaciones involucradas.

- A. Cifrado Simétrico (AES en CBC)

AES-256 en CBC requiere alrededor de 100 ciclos por byte
Como AES trabaja en bloques de 128 bits (16 bytes), para cifrar un bloque, necesitamos unos 1,000 ciclos. Por lo tanto:

$$\frac{2.9 \cdot 10^9 \frac{ciclos}{segundo}}{1000 \frac{ciclos}{bloque}} = 2.9 \cdot 10^6 \frac{bloques}{segundo}$$

Esto equivale a unas 1.81 millones de operaciones de cifrado simétrico por segundo.

- B. Cifrado Asimétrico (RSA con llave de 1024 bits): Como se requiere hacer

exponenciación de enteros muy grandes, el número de ciclos por cada bloque aumenta. Aproximadamente:

$$\frac{2.9 \cdot 10^9 \frac{ciclos}{segundo}}{10^7 \frac{ciclos}{bloque}} =^{290} \frac{bloques}{segundo}$$

Esto concuerda (al menos parcialmente) con los resultados vistos en la ejecución. El cifrado simétrica toma menos que 1ms, mientras que el asimétrico puede tomar tiempo en el orden de los milisegundos

VI) Referencias

• The Java™ Tutorials - Secure Communication with SSL Oracle. (n.d.). The Java™ Tutorials - Secure Communication with SSL. Retrieved from https://docs.oracle.com/javase/tutorial/security/

• RFC 3447 - Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specification Rivest, R. (2003). RFC 3447 - Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specification. Retrieved from https://tools.ietf.org/html/rfc3447

• Oracle Documentation for Java Oracle. (n.d.). Java Platform, Standard Edition Cryptography. Retrieved from https://docs.oracle.com/javase/8/docs/technotes/guides/security/

• "Java Network Programming" by Elliotte Rusty Harold Harold, E. R. (2004). Java Network Programming (3rd ed.). O'Reilly Media.

• Java Security Algorithms, Keys, and Certificates Sun Microsystems. (n.d.). Java Security Algorithms, Keys, and Certificates. Retrieved from https://docs.oracle.com/javase/8/docs/technotes/guides/security/

Universidad de los Andes
Ingeniería de Sistemas y Computación ISIS 2203 Infraestructura Computacional Semestre 2025-10