**Caso 3 – Infraestructura Computacional**

Santiago Rodríguez Cruz - 202013371

**Descripción de la organización de los archivos:**

Dentro del zip encontrará un proyecto de java, a continuación, encontrará una captura de la estructura del directorio:

Una captura de pantalla de un videojuego

Descripción generada automáticamente con confianza mediaDentro de la carpeta src encontrará el código fuente del caso, esta carpeta se divide en dos carpetas: Client y Server, en donde está el código del cliente y el servidor respectivamente. Cada una de esas carpetas contiene dos carpetas más, una para el escenario monothread y otra para el escenario multithread, además en la carpeta Client encontrará la llave publica generada por el servidor.

**Instrucciones para correr el servidor:**

En el archivo .zip viene adjunto la configuración del proyecto para Visual Studio Code, se recomienda que se abra allí pues solo tendría que correr los compounds correspondientes.

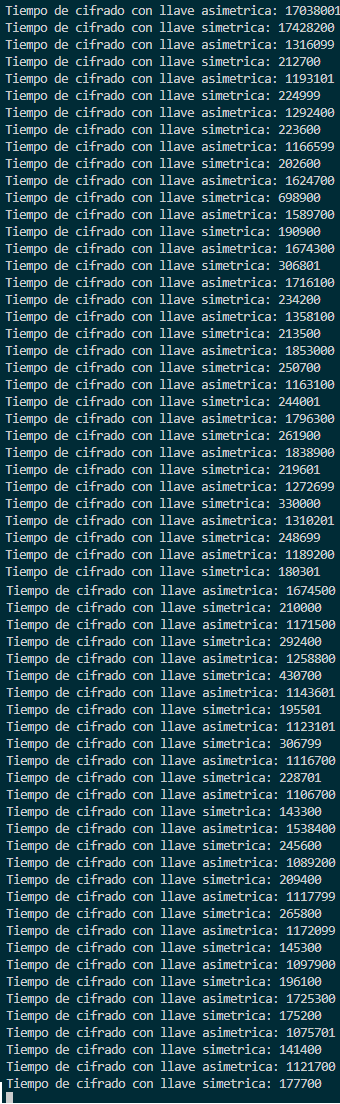
* **Launchear con Visual Studio Code:** Abra el proyecto en Visual Studio Code y diríjase al apartado  del menú, luego despliegue la lista  y elija el caso que va a probar, está el Caso Multithread y Caso Monothread, finalmente dele clic al botón start para desplegar las aplicaciones.
* **Launchear con otro IDE:** Ejecute por separado cada aplicación perteneciente a la misma carpeta, por ejemplo, si quiere ejecutar el caso monothread diríjase a la carpeta Monothread de cada uno de los elementos y ejecute los archivos Servidor.java y Cliente.java, si quiere ejecutar el caso multithread, diríjase a la carpeta Multithread de cada uno de los elementos y ejecute los archivos Server.java y ClientCreator.java.

**Descripción del esquema para generación de llaves:**

Se utilizaron los algoritmos RSA, AES y SHA de acuerdo con lo propuesto en el enunciado. La llave publica asimétrica del servidor se almacena en un archivo llamado p\_key.pub y se ubica en la carpeta Client, las demás llaves no son almacenadas en ningún archivo, sino que se hace referencia a la instancia que las tiene guardadas.

**Tareas:**

1. **Servidor y cliente iterativos con 32 consultas:**



1. **Servidor y cliente delegado:**
2. **4 delegados concurrentes:**

Texto

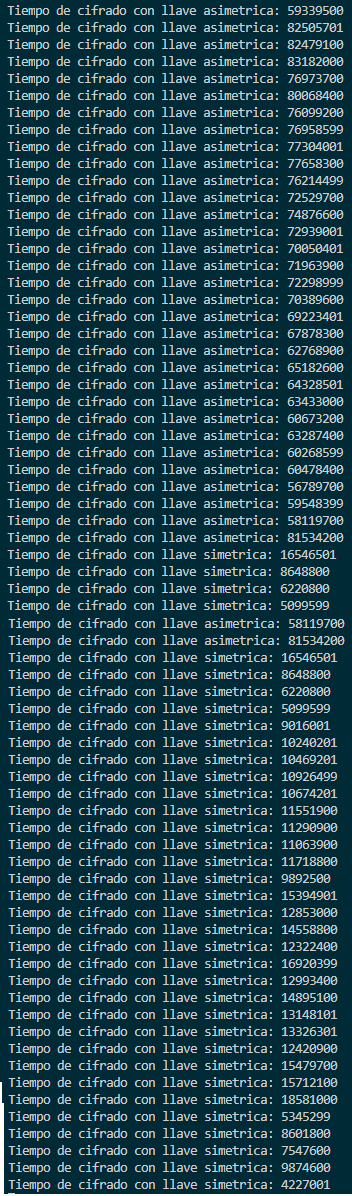
Descripción generada automáticamente

1. **16 delegados concurrentes:**

Calendario

Descripción generada automáticamente con confianza media

1. **32 delegados concurrentes:**



|  |  |
| --- | --- |
| **Servidor y cliente iterativo con 32 consultas** | |
| **Llave simétrica** | **Llave asimétrica** |
| 195618 | 17051600 |
| 302400 | 4455400 |
| 225500 | 1543100 |
| 281100 | 2082299 |
| 315200 | 1532099 |
| 104350 | 2044501 |
| 231800 | 1538400 |
| 215200 | 1818299 |
| 381401 | 1240601 |
| 264299 | 1441500 |
| 246500 | 1884300 |
| 221600 | 2352602 |
| 467000 | 1882899 |
| 299600 | 1729099 |
| 540800 | 1570499 |
| 585801 | 1863600 |
| 405200 | 1970500 |
| 327100 | 1286100 |
| 316800 | 1483300 |
| 298700 | 1603201 |
| 312201 | 1941600 |
| 192301 | 1283000 |
| 290099 | 1950101 |
| 187500 | 2010901 |
| 1106300 | 3467001 |
| 402901 | 2021400 |
| 305100 | 1568300 |
| 239301 | 1779100 |
| 310600 | 1455701 |
| 420000 | 1462401 |
| 315601 | 1445399 |
| 389301 | 1439399 |

|  |  |
| --- | --- |
| **Servidor y cliente delegado: 4 delegados concurrentes** | |
| **Llave simétrica** | **Llave asimétrica** |
| 28440700 | 51896900 |
| 30670700 | 57208899 |
| 27827000 | 53267000 |
| 32905600 | 52905999 |

|  |  |
| --- | --- |
| **Servidor y cliente delegado: 16 delegados concurrentes** | |
| **Llave simétrica** | **Llave asimétrica** |
| 30854900 | 17961300 |
| 871200 | 33669800 |
| 600700 | 33435900 |
| 158900 | 29759500 |
| 175800 | 28628700 |
| 394400 | 27748300 |
| 274900 | 26896600 |
| 353200 | 27423500 |
| 977700 | 27250800 |
| 715100 | 25952500 |
| 1544400 | 25115700 |
| 26076700 | 17016200 |
| 15107800 | 24421000 |
| 27923000 | 22152400 |
| 21170600 | 21552300 |
| 375700 | 18372000 |

|  |  |
| --- | --- |
| **Servidor y cliente delegado: 32 delegados concurrentes** | |
| **Llave simétrica** | **Llave asimétrica** |
| 156000 | 16235700 |
| 293500 | 53132400 |
| 161200 | 54085200 |
| 309000 | 48467000 |
| 801000 | 46706800 |
| 780700 | 41886500 |
| 711000 | 41552100 |
| 1001600 | 40995500 |
| 472500 | 34305300 |
| 188300 | 40127800 |
| 241300 | 30339200 |
| 378900 | 39437800 |
| 435800 | 35405500 |
| 190800 | 33473300 |
| 184400 | 28374500 |
| 150800 | 21391700 |
| 160600 | 28356000 |
| 1901500 | 21076800 |
| 25372600 | 27259600 |
| 27077000 | 19930800 |
| 414200 | 21724600 |
| 511100 | 70974500 |
| 351500 | 76200100 |
| 1055500 | 72926200 |
| 1142900 | 73620200 |
| 548600 | 71081800 |
| 371900 | 71081900 |
| 249400 | 67560000 |
| 703900 | 65363600 |
| 377700 | 60761100 |
| 361900 | 55718600 |
| 655000 | 58866100 |

1. En las graficas se puede evidenciar la diferencia de cifrado simétrico y asimétrico, el desempeño del cifrado simétrico es muchísimo mejor que el desempeño del cifrado asimétrico, esto sucede debido al tamaño de las llaves de cada uno de los cifrados y los algoritmos utilizados para el cifrado. Respecto al desempeño en los servidores, se pudo evidenciar que funcionaba más rápido con los iterativos que con los delegados, esto es debido al manejo de recursos cuando se ejecutan varios threads.
2. La velocidad de mi procesador es de 2.10GHz. Para ver cuantos retos puede cifrar mi maquina por segundo voy a hacer el promedio de todas las cifras que obtuve y convertiré esa cifra a segundos, esa será cuanto se tarda un reto en cifrarse, luego haré una regla de tres para ver cuantos retos puede cifrar mi procesador en un segundo.