UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

COMPARACIÓN DE RENDIMIENTO DE INTERCAMBIO DE DATOS CON JSON ENCRIPTADO

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor
Bach. Vallejos Díaz, Larry Amador
https://orcid.org/0000-0002-6450-7615

Asesor
Mg. Cachay Maco Junior Eugenio
https://orcid.org/0000-0003-4056-3142

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2021

APROBACIÓN DEL JURADO

COMPARACIÓN DE RENDIMIENTO DE INTERCAMBIO DE DATOS CON JSON ENCRIPTADO

_____

Bach. Larry Amador Vallejos Díaz
Autor

_____

Mg. Cachay Maco Junior Eugenio
Asesor

_____ _____

Mg. Diaz Vidarte Miguel
Presidente de Jurado de Tesis

Mg. Cachay Maco Junior
Secretario de Jurado e Tesis

_____

Mg. Celis Bravo Percy Javier
Vocal de Jurado de Tesis

DEDICATORIA

A mis padres quienes me dieron vida, educación, apoyo y consejos. A mis compañeros de estudio, a mis maestros y amigos, quienes sin su ayuda nunca hubiera podido hacer esta tesis. A todos ellos se los agradezco desde el fondo de mi alma. Para todos ellos hago esta dedicatoria.

AGRADECIMIENTO

Gracias a la universidad por haberme formado profesionalmente, gracias a todas las personas que fueron participes de este proceso ya sea de manera directa o indirecta. Gracias a todos ustedes, fueron ustedes los responsables de realizar su pequeño aporte del día de hoy se vería reflejando en la culminación de mi paso por la universidad.

RESUMEN

La presente investigación tiene como objetivo realizar la comparación del rendimiento en el intercambio de datos JSON encriptado mediante una plataforma web con metodología Scrum. El tipo de investigación es descriptiva-comparativa, porque se pretende implementar un algoritmo con una plataforma web para realizar el intercambio de datos y comparar su rendimiento con Json encriptado y por contrastación de hipótesis se realizó una investigación cuasi experimental, ya que se realiza un pre test - post test, mediante el uso de herramientas de comparación de rendimiento, porque se realizará un control de los indicadores antes y después de implementar el algoritmo. Con la aplicación web se pudo analizar la forma de envío de datos para demostrar el nivel de seguridad, además de la fortaleza de la clave y el tiempo de velocidad en cifrado y descifrado en datos Json sin encriptar y datos Json encriptados con RSA.

Palabras Clave: Encriptación, Transmisión, Comparación, JSON, Rendimiento de Datos, Velocidad de Datos, RSA.

ABSTRACT

The objective of this research is to perform the performance comparison in the encrypted JSON data exchange through a web platform with Scrum methodology. The type of research is descriptive-comparative, because it is intended to implement an algorithm with a web platform to perform the exchange of data and compare its performance with encrypted Json and by hypothesis testing a quasi-experimental research was carried out, since a pre test - post test, through the use of performance comparison tools, because the indicators are checked before and after implementing the algorithm. With the web application, the way data was sent could be analyzed to demonstrate the level of security, in addition to the strength of the key and the speed of encryption and decryption speed in Json data without encryption and Json data encrypted with RSA.

Key Words: Encryption, Transmission, Compare, JSON, Data Performance, Data Rate, RSA.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

I. INTRODUCCIÓN

1.1. Realidad Problemática.

El intercambio de información es una acción del día a día, desde que nos levantamos hasta que nos acostamos, solemos compartir con nuestro entorno información privada e información pública propia y ajena, es por ello que, vivimos en una sociedad donde la información es un activo de las empresas y organizaciones y genera movimientos de intereses y económicos, ya que esta se puede utilizar, comprar, vender, consumir (Gutiérrez & Tena Ayuso, 2003), por eso mismo, Areitio (2008) indica que, el que un sistema exponga o pierda información genera la preocupación de los responsables, debido a que se pierde la confianza de los clientes, la reputación de la empresa en el mercado, problemas legales y su extremo, pérdida del negocio.

En internet existen millones de personas que se encuentran en comunicación constante mediante las redes, esta forma de intercambio y transmisión de información viene acompañada con amenazas de seguridad, como robo, destrucción de la información, suplantación de identidades, hacking, cracking y estafas (Marrero Travieso, 2017). Y a pesar de saber todo esto, Gutiérrez & Tena Ayuso (2003) también mencionan que, los sistemas de información siguen siendo una incógnita y muchas veces son tan complejos que tienen errores de seguridad sin descubrir o que los vuelve vulnerables y esto puede ser aprovechado por aquellos que asechan y esperan el mínimo acceso para infiltrarse en la información.

La seguridad de la información genera un impacto económico tanto para las empresas usuarias como para las desarrolladoras de Software, lo cual depende de los recursos asignados y de lo grave de la información (De pablos Heredero, y otros, 2008). Es por ello que, al momento que la empresa desarrolladora considera la realización de un proyecto de Software debe tomar en cuenta en la implementación de la seguridad del software, el rendimiento en el intercambio de datos, debido a que, el envío de grandes cantidades de datos puede generar

una carga excesiva en tiempos de respuesta, lo que puede ocasionar fracasos en términos de rendimiento, además de generar una vulnerabilidad a la seguridad e integridad de los datos mediante ficheros de fácil lectura. Es por esto que Areitio (2008) afirma que, gestionar la seguridad de la información requiere de medidas técnicas de seguridad y también de procedimientos físicos y lógicos.

La mayoría de dichas arquitecturas hacen uso de técnicas criptográficas creadas en el cifrado de la información (Petticrew & Roberts, 2006), puesto no existe un único mecanismo capaz de proveer todos los servicios requeridos. Siendo diversas las técnicas criptográficas para lograr las comunicaciones, tomando en cuenta la autenticación (avala la idoneidad de mensajes y la procedencia), integridad (análogo a la función de checksum, y responde ante intercepción y alteración de los mensajes) y la confidencialidad (Garantiza el cifrado de mensaje para evitar su lectura por alguien no autorizado). Ya son 900 millones de web site (Stats, 2019) el uso de HTTPS es mínimo como protocolo de transferencia de datos, siendo pocas páginas que utilizan seguridad al transferir datos.

Así mismo, cuando se desarrolla software, muchas empresas consideran si implementarlo o no con ficheros Json, tomando en cuenta el garantizar la seguridad de sus datos y factores respecto a la vulnerabilidad, acceso y velocidad. Una de ellas, es una entidad de la Región Lambayeque, la cual cuenta con una división de Tecnologías de la Información (Anexo N 02 Entrevista de Software) responsable el desarrollo y monitoreo del software que funciona en la entidad, de los cuales, de 12 sistemas, 7 cuentan con ficheros Json, los que, bajo cálculo, llegan a un promedio de 421 ficheros entre todos.

Con la rápida progresión del intercambio de datos digitales, la seguridad de la información se ha convertido en un tema importante en la comunicación de datos. Los algoritmos criptográficos juegan un papel importante en los sistemas de seguridad de la información. Estos algoritmos utilizan técnicas que mejoran la seguridad y privacidad de sus datos al hacer que la información que solo

puede ser cifrada o descifrada por el propietario de la clave asociada sea indescifrable. Pero al mismo tiempo, estos algoritmos consumen grandes cantidades de recursos informáticos, como tiempo de CPU, memoria y energía de la batería. Por lo tanto, se debe evaluar el rendimiento de varios algoritmos criptográficos para encontrar el mejor algoritmo para uso futuro.

A pesar de una serie de propuestas de diseño sobre aceleradores criptográficos flexibles, la flexibilidad durante el diseño arquitectónico se limita a la determinación de operadores comunes y rara vez se explora desde perspectiva de los diseñadores de algoritmos (Shahzad, Khalid, Rákossy, Paul, & Chattopadhyay, 2013). La seguridad de los datos es una preocupación principal para todos los sistemas de comunicación, para lo cual, en la actualidad, se han propuesto e implementado varios algoritmos de criptografía, pero a pesar de que se utilizan varios algoritmos para la seguridad de los datos, comprometen la seguridad en cierto período (Harini, Gowri, Pavithra, & Selvarani, A novel security mechanism using hybrid cryptography algorithms, 2017).

Según Panda (2019), solo el 23% de las empresas consideran que la seguridad es un factor relevante al momento de considerar un proveedor de software, mientras que el 51% de las empresas no utilizan el cifrado para la protección de sus datos, a pesar de que el 48% de datos empresariales se encuentran en la nube.

1.2. Antecedentes de Estudio.

Abood (2017) describe en su problemática que En las tendencias recientes de la tecnología el desafío de mejorar la seguridad de la información es una necesidad importante al enviar y recibir datos en los campos de comunicación de datos y redes. Para resolver estos problemas hay varios métodos utilizados para proteger los datos de acceso no autorizado durante transmisión. Muchas técnicas se utilizan para proteger los datos del usuario. La técnica más eficiente es el uso de criptografía y estenografía. La criptografía y la estenografía son las

áreas maestras que tienen una oportunidad en el ocultamiento de la información y la seguridad. Empleó el método de verificación de la confidencialidad de la imagen en escala de grises que hace uso de pixel shuffling y RC4 cifrado de flujo para criptografía y Hash-LSB para estenografía. La función principal del arrastre de píxeles es que no implica ninguna modificación en los valores de bits y ninguna expansión de píxeles al final del cifrado y el procedimiento de descifrado. Aquí, los valores de los píxeles se vuelven a dibujar, se combinan moviéndolos desde una posición particular y luego se intercambian para darle a la imagen una codificación reconocible. El objetivo de utilizar cifrado de flujo RC4 es mejorar la confidencialidad del cifrado, obteniendo como resultado un concepto novedoso, que combina el algoritmo RC4 y el desplazamiento de píxeles con H-LSB para la imagen en escala de grises para mejorar la seguridad y privacidad. El método de calidad eficaz se debe a las siguientes razones. (i) La simplicidad del algoritmo de RC4 y pixel shuffling (ii) RC4 requiere sólo manipulaciones de bytes de longitud por lo que es adecuado para sistemas empotrados, (iii) Aunque RC4 tiene vulnerabilidades, lo combinamos con barajar para que sea casi imposible de romper (iv) En píxeles arrastrando todas las características de una imagen permanecen sin cambios durante el proceso de cifrado y descifrado. v) HLSB es más eficiente que LSB simplemente porque usa una función hash para seleccionar el bit de inserción LSB. Finalmente, los autores concluyen que el sistema propuesto es una de las mejores formas de ocultar la seguridad de los datos transmitidos entre el remitente y el receptor de una red de intrusión insegura. Tecnología de encriptación y parcheo RC Se implementa un algoritmo de encriptación con el propósito de encriptar imágenes secretas (jpg, png, gif, bmp) y luego integrarlas en la imagen de portada RGB (jpg, png, gif, bmp). Detección de cifrado. Encriptación de imagen usando el cifrado RC4 y Shuffling tiene una seguridad considerable.

El factor de calidad que implica las distribuciones de intensidad para las imágenes originales y la imagen mutilada son distintivas. Cuando nosotros considerar el histograma de imagen cifrada nos damos cuenta de que tener una distribución uniforme El menos significativo basado en hash La estenografía de bits (H-LSB) se ha implementado para incrustando la imagen encriptada en la

imagen de portada. El propuesto La técnica HLSB es el desarrollo de una mejora estenografía al ocultar datos en una imagen con menos Se ha hecho una variedad de bits de imagen que hace propuesta de algoritmo seguro y más efectivo y puede tener el Módulo de autenticación en línea con técnicas de cifrado. Se tomó en consideración esta investigación ya que aborda el tema de la criptografía y la estenografía se utilizan para garantizar la seguridad de los datos transmitidos. RC4 y algoritmo de cifrado de mezcla de píxeles se utiliza para cifrar la imagen secreta y Hash-LSB es una imagen cifrada incrustada en la imagen seleccionada Mínimos bits significativos de la imagen RGB y luego enviado. En el lado receptor la imagen se reconstruye desde la imagen RGB stego y utilizar RC4 y algoritmos de desencriptación de píxeles para obtener la imagen original.

Centeno, Chhabra, Fianza, Montes-Austria, & Ocampo (2018) describe en su problemática que la mayoría de los relojes inteligentes actuales se comunican de manera insegura con el teléfono móvil del usuario a través de Bluetooth Low Energy, lo que deja la información confidencial personal vulnerable a los ataques de seguridad. Debido a las limitaciones de tamaño y recursos en los relojes inteligentes, las características de seguridad se han descuidado en gran medida. Empleó cuatro algoritmos de cifrado diferentes (AES, Twofish, RSA y ElGamal) en un reloj inteligente y evaluó sus efectos sobre la vida útil de la batería, la memoria, la utilización de la CPU y la latencia de la comunicación del dispositivo, además probó estos algoritmos en varios tipos de datos intercambiados entre dispositivos portátiles y teléfonos móviles: mensajes simples, archivos pequeños y datos sensoriales, obteniendo como resultado que aplicar cualquiera de los algoritmos de cifrado que evaluamos no tiene un impacto negativo estadísticamente significativo en el rendimiento del reloj inteligente y Además, el costo de rendimiento que introducen estos algoritmos pasa desapercibido para los usuarios.. Finalmente, los autores concluyeron que el cifrado en dispositivos con especificaciones similares no afectará estadísticamente significativamente el rendimiento de los relojes inteligentes. Estas observaciones respaldan nuestra afirmación de que el cifrado debe usarse para todo el tráfico entrante y saliente hacia y desde el reloj. Además,

observó que el algoritmo de cifrado AES-Rijndael introdujo una sobrecarga de rendimiento muy mínima, lo que sugiere que entre los cuatro algoritmos que estudiamos, AES-Rijndael puede ser el mejor para usar en una plataforma de reloj inteligente. Es recomendable buscar un criptosistema que integre RSA AES, lo que mejora el nivel de seguridad introducido por el algoritmo al tiempo que pone la menor carga en el dispositivo. También se ha observado subjetivamente que los costos incurridos por el cifrado no son visibles para el usuario, y recomendamos que esto se verifique en futuras investigaciones. Se tomó en consideración esta investigación ya que aborda el tema del evalúa del rendimiento de algoritmos de cifrado en relojes inteligentes, ya que a medida que los relojes inteligentes ganan más utilidad en la vida diaria, manejan más tipos de datos confidenciales.

Sasikumar & Karthigaikumar (2018), describe en su problemática que, hoy en día, la mayoría de moderno NVIDIA GPU las arquitecturas son (Maxwell, Pascal, y Volta). Streaming Los flujos de multiprocesador son un componente importante de la arquitectura Pascal. Empleó el método de técnicas de optimización de transferencia de dato como, Optimización de kernel y Estrategia de Granularidad, obteniendo como resultado que, al utilizar a GPU Geforce 1080 con generación Pascal para objetar los efectos de su enfoque AES. GPU velocidad de rendimiento real. 1) estrategia del mecanismo de flujos: aclara la realidad velocidad de rendimiento (Gbps) de AES utilizando diferentes datos aleatorios archivos (MB) y utilizando diferentes mecanismos de número de secuencias (4, 8, 12, 16, 20, 24, 28 y 32). Usando La técnica de 32 secuencias proporciona el mejor rendimiento real en todos los archivos tamaños con velocidad de rendimiento máxima igual a 80 Gbps usando 512 Mbytes. Además, el uso de 24 transmisiones brinda un mejor rendimiento con velocidad de rendimiento máxima igual a 71 Gbps usando 256 Mbytes La velocidad de rendimiento de otras secuencias varía de un tamaño de archivo a otro. 2) estrategia de memoria unificada: la Figura 7 aclara la realidad velocidad de rendimiento (Gbps) de AES utilizando diferentes datos aleatorios archivos (MB) de usar la técnica de 32 secuencias en comparación con utilizando la técnica de memoria unificada. La figura aclara que usando La técnica de 32

transmisiones brinda la mejor velocidad de rendimiento real que usar memoria unificada en todos los tamaños de archivo con un máximo velocidad de rendimiento igual a 80 Gbps. En general, en todos los tamaños de archivo, la velocidad de rendimiento real con 32 transmisiones es aproximadamente dos veces que usar una estrategia de memoria unificada. Finalmente, los autores concluyeron que, la velocidad de rendimiento real AES utilizando la técnica de transmisión 32, proporciona un rendimiento máximo de 80 Gbps, que es el doble de rendimiento que usar la técnica mejorada de memoria unificada Pascal. Aunque la velocidad de rendimiento del kernel AES hace significativo el avance de rendimiento, el ancho de banda del bus PCI express es principales restricciones Hace que el rendimiento real de AES sea grande. Velocidad de rendimiento del kernel (Gbps) de AES usando diferentes archivos de datos aleatorios (MB) y usando 512 hilo por organización de bloque, asignación de memoria compartida para teclas redondas y 32 Bytes por técnica de granularidad de subprocesos en comparación con la mejor velocidad de rendimiento real utilizando 32 flujos. Aclarano que la velocidad de rendimiento del kernel da la máxima velocidad de rendimiento de 280 Gbps a 512 Mbytes, que es aproximadamente triple rendimiento que el rendimiento real con 32 transmisiones. Se tomó en consideración esta investigación ya que aborda el tema del Estándar de cifrado avanzado (AES) es fuertemente utilizado en diferentes niveles de seguridad de comunicación de datos ya que tiene mayor eficiencia y mayor seguridad en comparación con otros algoritmos de cifrado. Unidad de procesamiento de gráficos (GPU) es una de las plataformas más importantes utilizadas para mejorar Algoritmo AES Rendimiento. Desafortunadamente, el AES actual el rendimiento sobre GPU difícilmente puede mejorar debido a la CPU-GPU sobrecarga de transferencia de datos. En este artículo, el algoritmo AES-ECB se implementa en NVIDIA GTX 1080 (arquitectura Pascal).

Pushpa (2020) describe en su problemática que los datos de atención médica almacenados en la nube se consideran un registro altamente sensible, que debe ocultarse a accesos no autorizados para proteger la información sobre el paciente. El modelo propuesto se presenta mediante la integración de la técnica

de transformación de wavelet discreta 2D con un esquema de cifrado híbrido propuesto., obteniendo como resultado garantizar mediante el uso de diferentes medidas de rendimiento. Es interesante que el modelo propuesto haya alcanzado resultados supremos con un MSE mínimo de 0.08 y un PSNR máximo de 58.96dB. Finalmente, el autor concluyó que este estudio ha presentado una nueva hibridación del modelo de cifrado de datos para salvaguardar los datos de diagnóstico en imágenes médicas. El modelo propuesto se presenta mediante la integración del proceso 2D DWT con una hibridación de algoritmos de encriptación Blowfish y Two fish. El modelo presentado comienza con el cifrado de datos secretos y luego oculta el resultado mediante el uso del resultado en una imagen de portada mediante el uso de DWT 2D 1L y 2L. Se tomó en consideración esta investigación ya que presenta una nueva hibridación del modelo de cifrado de datos para proteger los datos de diagnóstico en imágenes médicas.

Harini, Gowri, Pavithra, & Selvarani (2017) describe en su problemática que la seguridad de los datos es una preocupación principal para todos los sistemas de comunicación, para lo cual, en la actualidad, se han propuesto e implementado varios algoritmos de criptografía, pero a pesar de que se utilizan varios algoritmos para la seguridad de los datos, comprometen la seguridad en cierto período. Empleó el método de combinar varios algoritmos seguros para proporcionar un entorno altamente seguro para la transmisión de datos. Los algoritmos que se combinarán son el algoritmo criptográfico simétrico AES, el algoritmo asimétrico RSA y el algoritmo hash MD5. Con estos tres algoritmos, podemos garantizar tres primitivas de criptografía: confidencialidad, autenticación e integridad de los datos, obteniendo como resultado una metodología con una mejor seguridad, esto se debe a que se integró algoritmos AES en la estructura RSA, además de haber agregado el algoritmo MD5. Finalmente, los autores concluyeron que un nuevo protocolo de seguridad está diseñado para mejorar la seguridad. El algoritmo híbrido propuesto es más seguro contra ataques fundamentales que se enfrentaron cuando el algoritmo AES se usó solo. La metodología propuesta mejora la seguridad al agregar el algoritmo MD5 e integrar el algoritmo AES en la estructura RSA con hashing.

Se tomó en consideración esta investigación ya que aborda el tema de la combinación de algoritmos seguros (AES, RSA y MD5), para crear un algoritmo altamente seguro para la transmisión de datos.

Liu & Wen (2018) describe en su problemática que en algunas áreas donde la seguridad o la privacidad están involucradas, la aplicación en el mercado puede ser fácilmente interceptada por terceros al transferir archivos, y la información transmitida está altamente amenazada y la seguridad no está garantizada. Empleó el método de aplicación de cifrado dual, con el fin de resolver la clave para transmitir en texto claro; y luego analice el principio del canal secreto, antes de la transmisión del archivo, a través del canal secreto de la transmisión secreta de la información clave de doble cifrado para lograr la transmisión de información importante en el oculto, obteniendo como resultado el mismo carácter que el cifrado DES de "República Popular de China" con 5 milisegundos, mientras que RSA tarda 13 milisegundos, lo cual describe que el cifrado DES es más rápido que RSA. Al analizar la complejidad de la clave, el cifrado RSA es más seguro que DES. Entonces, cuando transfiere una gran cantidad de información, seleccione DES para la encriptación, se puede usar una pequeña cantidad de información importante para la encriptación RSA. Finalmente, los autores concluyeron que la implementación del algoritmo de cifrado dual en canal oculto no solo combina las ventajas de DES y RSA para mejorar el efecto de cifrado, sino que también UTILIZA canales ocultos para realizar la transmisión secreta de la clave. El establecimiento de canales encubiertos también se puede diseñar de acuerdo con las necesidades reales. Se tomó en consideración esta investigación ya que aborda el tema de analizar el cifrado simétrico de DES representativo y cifrado asimétrico con el representante de los principios básicos del algoritmo RSA, el uso de estas dos características del algoritmo de cifrado.

Solichin, Putra, & Diniari (2018) describe en su problemática que el creciente número de usuarios de dispositivos móviles está impulsando el crecimiento del número de aplicaciones móviles y servicios de intercambio de servicios de datos basados en la web. Con la tecnología del servicio web, el intercambio de datos

entre sistemas se hace más fácil a pesar de que tiene una plataforma diferente. Empleó el método del algoritmo de compresión y cifrado para optimizar el servicio web RESTful. Propusimos la compresión LZW para reducir el espacio de almacenamiento necesario para almacenar datos en el servidor. También propusimos un algoritmo Blowfish y AES-128 para asegurar el intercambio de datos a través de la red de Internet, obteniendo como resultado un cambio significativo en el tamaño de los datos después de la compresión. La tasa de compresión promedio de todas las pruebas fue de 1.78, y el ahorro de capacidad de almacenamiento resultante fue de 38%. De los resultados de la prueba también se puede concluir que cuanto mayor es el tamaño del archivo, mayor es la relación de compresión y más ahorro de espacio de almacenamiento. La Figura 5 presenta una comparación del tamaño de los datos originales con los datos comprimidos. Mientras tanto, la Figura 6 muestra la comparación de tiempo para almacenar datos entre datos comprimidos y no comprimidos para cada dato de prueba. Finalmente, los autores concluyeron que esta investigación produjo un servicio web con varios servicios. Además, este estudio también produjo un prototipo de aplicaciones móviles. Al probar el proceso de cifrado de los datos, todos los datos probados se intercambiaron con éxito entre el servicio web y el cliente. Las pruebas de servicios web mostraron que todos los servicios podrían funcionar bien. Se tomó en consideración esta investigación ya que analiza el uso de un algoritmo de compresión y cifrado para optimizar el servicio web RESTful.

Santoso, Muin, & Mahmudi (2019) describe en su problemática que el intercambio de datos es un tema inevitable en esta era digital. Especialmente datos del sistema en la nube. Los datos deben protegerse contra el uso indebido por parte de personas no autorizadas. Los algoritmos de protección de datos son bien conocidos. La seguridad de los datos es más segura si combina dos o más algoritmos de cifrado. Se aplicó un método que combina dos algoritmos de cifrado., a saber, AES y Twofish que produjeron (Hybrid AES y Twofish). Para el proceso de cifrado, el primer Plaintext se cifra con AES y produce cifrado de nuevo con el segundo algoritmo es Twofish utilizando la clave SHA 265, obteniendo como resultado proteger los datos personales y los datos

organizativos, porque los datos se cifran cuando se transmiten o utilizan los datos, así como cuando se encuentran en la ubicación de almacenamiento. Beneficios del almacenamiento seguro en la nube de Multi – Tenancy inquilinos. Cifre los datos del sistema en la nube y proteja a los usuarios y a las personas no autorizadas del uso de claves de cifrado para acceder a los datos del servicio en la nube.Proporcionar puerto seguro de notificación de infracción, si se produce una violación de datos en una persona, se pierde información identificable, la parte infractora debe notificar a todas las personas afectadas. Asegúrese de que su almacenamiento en la nube esté mejor protegido contra amenazas. Finalmente, los autores concluyen que el algoritmo híbrido propuesto es más seguro porque utiliza la clave generada por SHA256 para cifrar y descifrar el mensaje. El algoritmo recién creado es una combinación de cifrado que utiliza dos algoritmos simétricos, AES y Twofish. Los algoritmos coincidentes son relativamente seguros y fáciles de implementar. Se tomó en consideración esta investigación ya que aborda el tema de incorporar dos algoritmos criptográficos, a saber, AES (Advanced Encryption Standard) y Twofish con clave de 256 bits generada por la función HASH SHA 256.

Kaur & Soni (2017) describe en su problemática que en redes utilizamos cable coaxial, cables de par trenzado para transferir datos desde el origen al destino. Estos medios de transferencia de datos son de naturaleza más lenta y los datos que se transfieren pueden filtrarse o perderse durante la transmisión y tampoco hay seguridad de los datos. Para superar estos problemas durante la transmisión, utilizamos fibra óptica. Empleó el método de introducir el cifrado en fibra óptica utilizando imágenes, también han propuesto varios algoritmos de cifrado para hacer que los datos del medio óptico sean seguros y también se han hecho comparaciones entre los algoritmos AES, DES y RSA para encontrar el mejor algoritmo de seguridad, que debe usarse en medio óptico para hacer que los datos del medio óptico sean seguros y no sean pirateados por atacantes, obteniendo como resultado mejores en términos de velocidad y encriptación. La simulación se realiza en MATLAB. Finalmente, los autores concluyeron que los mecanismos de seguridad que se analizan en este documento son eficientes. Los algoritmos de cifrado juegan un papel importante

en la seguridad de los datos en el medio óptico y, en comparación con los diferentes parámetros utilizados en los algoritmos, se ha descubierto que el algoritmo AES usa menos tiempo para ejecutar los datos del medio óptico. El algoritmo Blowfish tiene menos requisitos de memoria. El algoritmo DES consume menos tiempo de cifrado. RSA consume el mayor tamaño de memoria y el tiempo de cifrado. Se tomó en consideración esta investigación ya que aborda el tema de la transferencia de datos que será a través de fibra óptica por ser más rápida, así como también utiliza algunos mecanismos de seguridad para la transmisión segura de datos.

P. Chaudhury et al. (2017) El problema explica que el sistema de cifrado de clave compartida utiliza exactamente la misma clave para el cifrado y descifrado, por lo que la clave privada debe compartirse. Por lo tanto, un intruso puede descubrir la clave privada mientras envía los datos. El algoritmo RSA tiene muchas limitaciones. Pueden ocurrir varios tipos de ataques con el algoritmo RSA, incluidos los ataques de búsqueda directa y los ataques de forma común. Otra desventaja de utilizar el cifrado RSA para la criptografía es el problema del análisis factorial y la velocidad de cálculo. Empleó el método de utilizar un algoritmo criptográfico basado en clave asimétrica que utiliza cuatro números primos llamados ACAFP para manejar cuatro números primos y proporcionar seguridad. Cuatro números primos no se desintegran fácilmente y aumentan la efectividad en todas las redes, obteniendo como resultado que, el algoritmo propuesto ACAFP trata con cuatro números primos de pequeño tamaño y es por eso que es muy fácil calcular el factor. Por lo tanto, dicho método requiere menos memoria y energía que RSA. La memoria requerida para RSA es mucho mayor que la de ACAFP. Finalmente, los autores concluyeron que ACAFP es un algoritmo de modificación RSA. El programa es muy sofisticado en términos de consumo de memoria y velocidad de computación porque ACAFP puede resolver el problema de factorización RSA al incluir números primos más pequeños. Obtenga más información sobre los programas ACAFP que pueden proteger datos como RSA. En el futuro, analizaremos la resistencia de este programa a varios ciberataques**.** Se tomó en consideración esta investigación ya que aborda el tema de usar un algoritmo

de criptosistema RSA modificado llamado "Algoritmo criptográfico basado en clave asimétrica que usa cuatro números primos para asegurar la comunicación de mensajes (ACAFP)" para manejar cuatro números primos y proporcionar seguridad.

1.3. Teorías Relacionadas al Tema.

1.3.1 Procedimientos de Seguridad Lógicos.

Existen diversas técnicas para luchas contra amenazas como robo, interceptación de mensajes, sabotajes, Etc. (Sampalo de la Torre, Leyva Cortés, Garzón Villar, & Prieto Tinoco, 2003).

a. Cifrado de datos.

   También llamado encriptado de datos, es el método más confiable de protección, que transforma los datos para que no sean entendible, realizado a través de elementos lógicos o físicos, aunque su desventaja es el consumo de muchos recursos.

b. Firma digital y certificación.

   Permite que el receptor compruebe la autenticidad del mensaje mediante la aplicación de la función hash, garantizando al usuario la afirmación del envío de algún documento electrónico, el cual toma carácter de certificado una vez firmado.

c. Control de acceso.

   Medidas basadas en características físicas del usuario, posesión de un objeto, conocimiento de un secreto y sistemas expertos que se usan para verificar el acceso a ciertos datos.

d. Diccionario de seguridad.

   Base de datos cifrada, al que solo tiene acceso el administrador, con información de usuarios y sus privilegios que sirve para verificación y validación de acceso.

e. Antivirus.

   Programa especializado para detectar y eliminar programes que alteren el funcionamiento del equipo, es decir software maligno.

1.3.2 Criptografía.

Pabón Cadavid (2010) y Aguilera López (2010) indican que, proviene del griego “Kryptos” que significa “escondido” y del griego “Graphein” que significa “escritura”, llamada también escritura secreta, parte de la criptología, quien también contiene al criptoanálisis, el cual tiene como finalidad el descifrado de la información procesada por un criptosistema. La criptografía es la ciencia responsable del cifrado y descifrado de la información para garantizar una transmisión segura entre el emisor y el receptor.

Tipos de Cifrado y Descifrado.

a. Criptografía Simétrica.

Aguilera (2010) menciona que, se usa la misma clave para cifrar y descifrar información, donde la clave es conocida por el emisor y receptor, comunicada por un conducto seguro, aunque una desventaja es que la clave única puede ser interceptada por terceros. Usa las funciones hash y algoritmos IDEA, REC2, Bowfish y 3DES.Las formas de encriptación son las siguientes (S.A., 2002):

i. Encriptación de Clave Simétrica.

Es la clave secreta, proceso donde el texto enviado se encripta y desencripta por una misma clave, lo cual requiere un intercambio previo de la clave entre el emisor y el receptor mediante un canal seguro.

b. Criptografía Asimétrica.

Se usa claves diferentes para el cifrado y descifrado, las cuales son públicas y privadas respectivamente. Los algoritmos usados son DSA y RSA (Aguilera Lopez, 2010).Las formas de

encriptación son las siguientes (S.A., 2002):

i. **Encriptación de Clave Privada.**

   Una de las más usadas en internet, es la encriptación de un texto mediante una clave que solo conocen el emisor y el receptor.

ii. **Encriptación de Clave Pública.**

   Proceso donde se involucran la clave pública y simétrica, donde la primera es puesta a disposición menos al emisor y receptor del mensaje, mientras que la segunda solo será conocida por el receptor, el mismo que se encargará de compartir la clave pública con la que se encripta o codifica el mensaje y la clave privada para desencriptarlo.

Herramientas de Cifrado.

Según Aguilera (2010), las herramientas de encriptado son las siguientes:

a. **TrueCrypt.**

   Se usa cuando se requiere proteger los datos almacenados en un equipo personal, ya que permite la creación de una partición o unidad encriptada y oculta en el lugar deseado.

b. **CryptoForge.**

   Se usa para encriptar un archivo que luego se quiere enviar y/o transportar en un dispositivo externo, sin temor de robo o pérdida.

c. **Hash.**

   Se usa para representar el contenido de los archivos o documentos mediante claves generadas, su resultado de su ejecución se Hashing y realiza la comparación de

dos archivos.

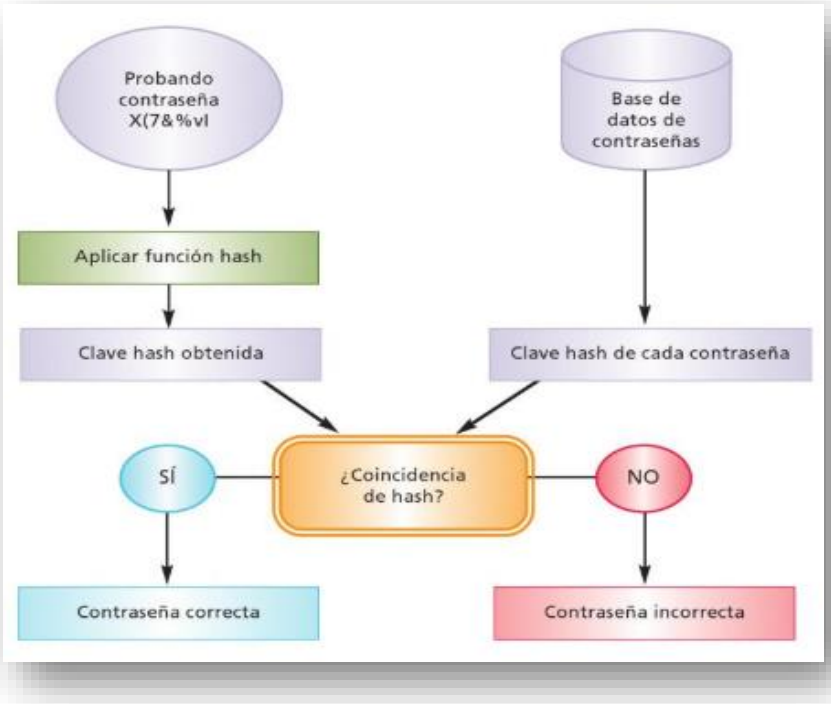


Figura 1. Comprobación de Contraseñas Mediante Claves Hash. Fuente: Aguilera (2010)

d. MD5.

Muy actualizado en la actualidad respecto a descargadas de internet, es un algoritmo de reducción criptográfica que protege a los archivos contra el malware.

Algoritmo RSA.

Llamado así por las iniciales de sus descubridores (Rivest, Shamir, Adleman). Esquema para el cifrado en bloque de clave pública, robusto y seguro, pero para una buena seguridad se requiere 1024 bits como mínimo, lo que ocasiona cierta lentitud en el encriptado. (tanenbaum, 2003) Su método se basa en la teoría de números:

1. Elegir dos números primos grandes, p y q. Ejemplo. cada uno de 1024 bits.
2. Calcular n = pq y z =(p-1) (q-1)
3. Elegir e (con e < n) que no tenga factores comunes con z (e y z son primos relativos)
4. Encontrar un número d, tal que ed-1 sea divisible de forma exacta entre z (ed mod z=1)
5. La clave pública es (n,e) y la clave privada es (n,d)

Oriol & Jordi (2020) el algoritmo RSA es el más indicado para las sesiones que requieren un intercambio de datos de alto rendimiento y con fuertes medidas de encriptación que no hayan sido descifradas, que ofrezcan un alto nivel de confidencialidad por lo cual recomienda el uso de claves públicas de 1024 bits.

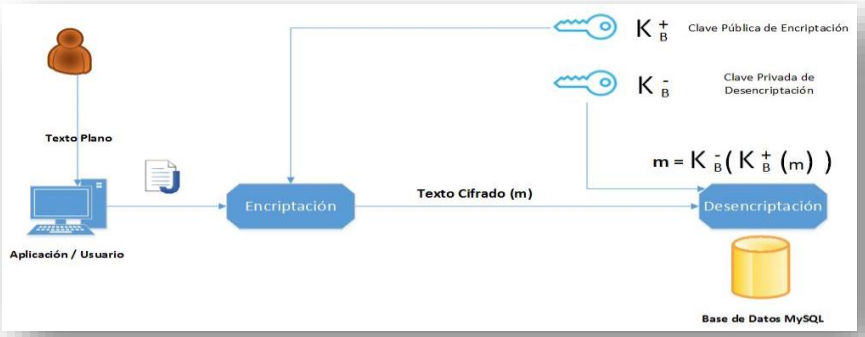


Figura 1. Diagrama de Encriptación RSA

Fuente: Oriol & Jordi (2020)

Cada sesión en el intercambio de datos es encriptada por el algoritmo asimétrico RSA que forma parte de la public class Tabla_Visual.java de la aplicación, la cual previamente realiza una conexión segura hacia el servidor de base de datos MySQL. La implementación del algoritmo RSA en la aplicación contiene el siguiente código fuente:

```
ResultSet rsa = null;
PreparedStatement psa = null;

try {
    psa = conec.getConnection().prepareStatement(sqla);
    rsa = psa.executeQuery();
    while (rsa.next()) {
        nid = rsa.getInt(1);
    }
} catch (SQLException ex) {

} finally {
    try {
        psa.close();
        rsa.close();
        //conec.desconectar();
    } catch (Exception ex) {
    }
}
```

Figura 2. Fragmento de Código Fuente de Implementación RSA. Fuente: Fuente: Oriol & Jordi (2020).

RSA vs DSA.

Ambos son criptografía asimétrica, pero tienen diferencias respecto a su desempeño (Sawakinome, 2018) (Seguridad, 2011):

DSA.

- Basado en logaritmos discreto
- Más rápido para la generación de claves y descifrado
- Puede generar firmas rápidas, pero con una lenta Validación.
- No se encuentra soportado ampliamente y aunque OpenSSL permite a sus claves alcanzar más bits, solo llegan hasta 1024.

RSA.

- Basado en factorización de enteros grandes.

- Tiene mayor velocidad en el momento del cifrado, lo cual lo vuelve conveniente al momento de cifrar y verificar una firma digital.
- Tiene una mayor ampliación en sus certificados comerciales.
- Para la creación de firmas cuenta con su propio generador aleatorio de números.
- Sus claves pueden alcanzar hasta 4096 bits.

1.3.3 Encriptación.

Es una técnica realizada mediante algoritmos, en el que un texto plano se transforma en un texto cifrado y no puede ser leído por cualquier usuario a menos que, conozca los mecanismos de encriptación (S.A., 2002). Según Sampalo et al. (2003), es un nivel de seguridad de sistemas de protección estándar que resulta en un mensaje cifrado, pero aun así puede estar propenso a intrusos, los cuales son pasivos cuando solo estudian el mensaje, y son activos cuando alteran el mensaje ya que conocen el cifrado.

Lawrence Lessig señala que, “Las tecnologías de la encriptación constituyen el avance tecnológico más importante de los últimos mil años. Ningún otro descubrimiento tecnológico… tendrá un impacto más significativo en la vida social y política de la humanidad…”, mencionado por Pabón Cadavid (2010)

Sampalo et al. (2003) menciona que los elementos necesarios para método de encriptación son los siguientes:

a. Texto en claro, es el texto original.
b. Método de cifrado, compuesto por el algoritmo y la clave.
c. Texto cifrado, resultado del método descifrado.

Métodos de Encriptación.

a. Codificación por Sustitución.

Método sencillo y fácil de descifrar, denominado “cifrado de César”, donde las letras o palabras son sustituidas por otras letras o palabras. Se clasifica en Sustitución por desplazamiento, Sustitución Monoalfabética y Cifrado Polialfabético.

b. Cifrados por Transposición.

Método que modifica la posición de las letras que componen el mensaje, está compuesto por una matriz de cifrado, una palabra clave y el texto en claro.

c. Cifrado DES.

Método creado por IBM, es usado por Linux, quien encripta ficheros que contiene claves de acceso de usuarios. Tiene un funcionamiento complejo que se compone de 19 etapas, el cual funciona en bloques de bits, donde un texto claro se parte en fracciones de 64 bits y se codifica bloque a bloque.

1.3.4 Json.

Aguirre (2020) y JSON (2020) señalan que, es un estándar para intercambio de datos entre servidor y navegador, formato de texto que utiliza arreglos de lenguajes conocidos en un formato de texto plano e independiente.

Está conformado por estructura de pares en colección de nombre y valor, donde, el objeto comienza y finaliza con llaves, el nombre es seguido por dos puntos y los pares son separados por comas. Un arreglo inicia y finaliza con corchetes y también son separados por comas.

Var fruta = ‘{“nombre”: “naranja”, “color”: “naranja”}’

Puede trabajar con múltiples lenguajes como Ruby, Python, Scala, ASP, C++, Cobol, PHP, Java, MatLab, JavaScript, los lenguajes soportan Json mediante librerías externas o soporte nativo. En muchos casos se usan APIs para integrar los datos con otros servicios y aplicaciones. Entiéndase como API a una interfaz de programación de aplicaciones, es decir un servicio web basado en un protocolo de envío de datos.

Estándares de Comunicación.

Los estándares de comunicación son los siguientes (Unifica, 2020):

a. Json Web Algorithms – JWA. - Usado para enumerar e identificar algoritmos criptográficos que serán usados en diversos estándares.
b. Json web key – JWK. - Usado para describir el formato y manejos de claves criptográficas.
c. Json Web Signature – JWS. - Usado para describir el manejo y producción de mensajes firmados mediante algoritmos criptográficos.
d. Json Web Encryption – JWE. - Usado para describir la producción y manejo de mensajes encriptados.
e. Json Web token – JWT. - Usado para describir la representación de pares codificados en Json.

Terminología de Especificación.

La terminología de especificación para JWS y JWE son las siguientes (Jones, 2012):

a. Términos de JWS.

- **i.** Firma JSON Web Signature, la cual consta de encabezado, carga útil y valor de la firma.
- **ii.** Objeto de texto Json representada por una cadena codificada

UTF-8.

iii. Encabezado JWS, descripción de la firma digital mediante objeto Json.

iv. Carga útil JWS, mensaje compuesto de una secuencia arbitraria de bytes a proteger.

v. Firma JWS, protección del encabezado y carga útil mediante una matriz de bytes.

vi. Codificación Base64url.

vii. Encabezado JWS codificado mediante Base64url.

viii. Firma JWS codificada mediante Base64url.

ix. Entrada protegida JWS.

x. Espacio de nombres resistente a colisiones.

b. Términos de JWE

i. Cifrado JWE, que consta de encabezado, clave cifrada, vector de inicialización, texto cifrado y valor de integridad.

ii. Mensaje o texto plano sin formato basado en una secuencia arbitraria de bytes.

iii. Texto cifrado sin formato.

iv. Clave simétrica con cifrado de contenido para la generación del texto cifrado generado por el destinatario.

v. Clave de integridad del texto cifrado y parámetros utilizados.

vi. Clave maestra de contenido.

vii. Objeto de texto mediante una cadena codificada en UTF-8.

viii. Encabezado mediante objeto de texto que describe operaciones de cifrado.

ix. Clave maestra cifrada con clave del destinatario mediante encapsulado.

x. Vector de inicialización contenido en una matriz de bytes.

xi. Texto cifrado contenido en una matriz de bytes.

xii. Valor de integridad. Encabezado codificado en Base 64.

xiii. encabezado cifrado y codificado en Base64.

xiv. Clave cifrada y codificada en Base64.

xv. Vector de inicialización codificado en Base64.

xvi. Texto cifrado y codificado en Base64.

xvii. Valor de integridad codificado en base64.

xviii. Algoritmo de cifrado autenticado.

1.4. Formulación del Problema.

¿Cuál será el resultado de la comparación del rendimiento entre formatos JSON encriptados y sin encriptar, para el intercambio de datos?

1.5. Justificación e Importancia del Estudio.

1.5.1. Justificación Social.

La investigación se realiza porque existe la necesidad de mejorar la seguridad en los proyectos de software mediante el cifrado de datos y así poder proteger información sensible y resguardar la privacidad de los usuarios.

1.5.2. Justificación Teórica.

La investigación se realiza con la finalidad de contribuir al conocimiento sobre seguridad mediante el cifrado de datos y medir el rendimiento en el intercambio de los mismos. Donde el resultado podrá ser una propuesta que será incorporada como conocimiento en el desarrollo de software y así mejorar el nivel del producto.

1.5.3. Justificación Práctica.

La investigación se realiza debido a la necesidad de conocer el rendimiento de intercambio de datos mediante el uso de la encriptación Json.

1.6. Hipótesis.

Los formatos JSON encriptados tendrán un mejor rendimiento en el intercambio de datos.

1.7. Objetivos.

1.7.1. Objetivo General.

Realizar la comparación del rendimiento en el intercambio de datos con JSON y JSON Encriptado.

1.7.2. Objetivos Específicos.

a. Realizar el análisis de las estructuras algorítmicas de Json y Json encriptado.

b. Establecer el escenario para realizar las pruebas.

c. Estimar el rendimiento en el intercambio de datos.

d. Comparar el rendimiento de las estructuras algorítmicas.

II. MATERIAL Y MÉTODOS

2.1 Tipo y Diseño de Investigación.

2.1.1 Tipo de Investigación.

Descriptiva – Comparativa, porque se pretende implementar un algoritmo con una plataforma web para realizar el intercambio de datos y comparar su rendimiento con Json encriptado.

2.1.2 Diseño de Investigación.

Cuasi experimental, ya que se realiza un pre test - post test, mediante el uso de herramientas de comparación de rendimiento, porque se realizará un control de los indicadores antes y después de implementar el algoritmo.

G: 01 X 02

X: Variable Experimental: Integración de Json Encriptado.

O1: Aplicación del Pre – Test: Análisis comparativo para medir el rendimiento de intercambio de datos.

O2: Aplicación del Post – Test: Análisis comparativo para medir el rendimiento de intercambio de datos.

2.2 Población y Muestra.

2.2.1 Población.

En base al promedio de ficheros por sistemas de 7 sistemas explicados en la situación problemática, se ha considerado 421 ficheros Json.

2.2.2 Muestra.

La muestra se ha determinado por conveniencia y está constituida por dos formatos de ficheros JSON.

2.2.3 Variables, Operacionalización

Las variables usadas como elementos básicos en el desarrollo de la hipótesis se han identificado de la siguiente manera:

2.2.3.1. Variables

A. Variable Independiente

Integración del formato JSON encriptado

B. Variable Dependiente

Rendimiento en el intercambio de datos

2.2.3.2. Indicadores.

Tabla 1

Indicadores de Variables.

Tipo	Variable	Indicador	Descripción	Fórmulas
Independiente	Integración de Json	Grado de Seguridad	Capacidad de almacenamiento: capacidad del algoritmo (unidad de medida en Kilobytes)	∑ (CCGB) CCGB = Total de Caracteres Generados por Bit
		Fortaleza de clave	Número de combinaciones del algoritmo criptográfico	TCD = DC - TCS DC: Dimensión de Clave TCS: Total de cifras similares TCD: Total de cifras diferentes
Dependiente	Rendimiento en el intercambio de datos	Tiempo de velocidad en cifrado y descifrado	Velocidad que posee el algoritmo criptográfico para realizar una tarea, es necesario tener el tiempo que demora en cifrar o descifrar.	∑ (HDD - HDC) /TP HDD: Hora de Descifrado HDC: Hora de Cifrado TP: Total Paquetes

2.3 Técnicas e Instrumentos de Recolección de Datos.

En la tabla mostrada a continuación se puede observar las técnicas e instrumento usadas para la recolección de datos.

Tabla 2.

Técnica e Instrumento de Recolección de Datos.

Técnicas	Instrumentos	Elementos de la población	Propósito
Control de tiempos	Observación	Ficheros Json	Conocer la velocidad de cifrado y descifrado, almacenamiento y tiempo de transferencia.

2.4 Procedimiento para el Análisis de los Datos.

A. Análisis bibliográfico de las arquitecturas de JSON, y los algoritmos criptográficos para la encriptación de JSON.
B. Creación de 1 fichero de JSON ENCRIPTADO utilizando el algoritmo criptográfico RSA y 1 fichero JSON sin encriptación.
C. Construcción de una aplicación para subida los ficheros utilizando metodología Scrum.
D. Medición del tiempo de velocidad de cifrado y descifrado del fichero JSON encriptado.
E. Medición de la capacidad de almacenamiento de los Ficheros JSON.
F. Medición del tiempo de transferencia de datos de los ficheros JSON.
G. Comparación mediante tablas dinámicas y estadísticas de los resultados.
H. Análisis de los resultados.

2.4.1 Criterios Éticos.

A. Objetividad: Mediante la indicación de hipótesis y determinación de conclusiones se identificarán los mejores resultados respecto de la idea y los datos materia de análisis.

B. Originalidad: En esta parte la interpretación sustentada en el diverso material bibliográfico planteará nuestro particular punto de vista respecto al objeto de estudio.

C. Veracidad: En la misma línea de la objetividad y como consecuencia del análisis con una posición original, es posible argumentar que lo esgrimido en la investigación es íntegramente verás.

2.4.2 Criterios de Rigor Científico.

Martínez (2011) explica que, los criterios de rigor científico para una investigación cuantitativa son los siguientes:

A. Valor de verdad – validez interna

   Igualdad entre los datos recolectados y la realidad estudiada.

B. Aplicabilidad – validez externa

   Nivel en el que se puede aplicar descubrimientos de una investigación en otro contexto.

A. Consistencia – fiabilidad interna

   Nivel en que diversos instrumentos pueden tener las mismas medidas y circunstancias.

B. Neutralidad – fiabilidad externa

   Nivel en el que la perspectiva del investigador no influye en la investigación.

III. RESULTADOS.

3.1. Discusión de Resultados.

3.1.1. Grado de Seguridad.

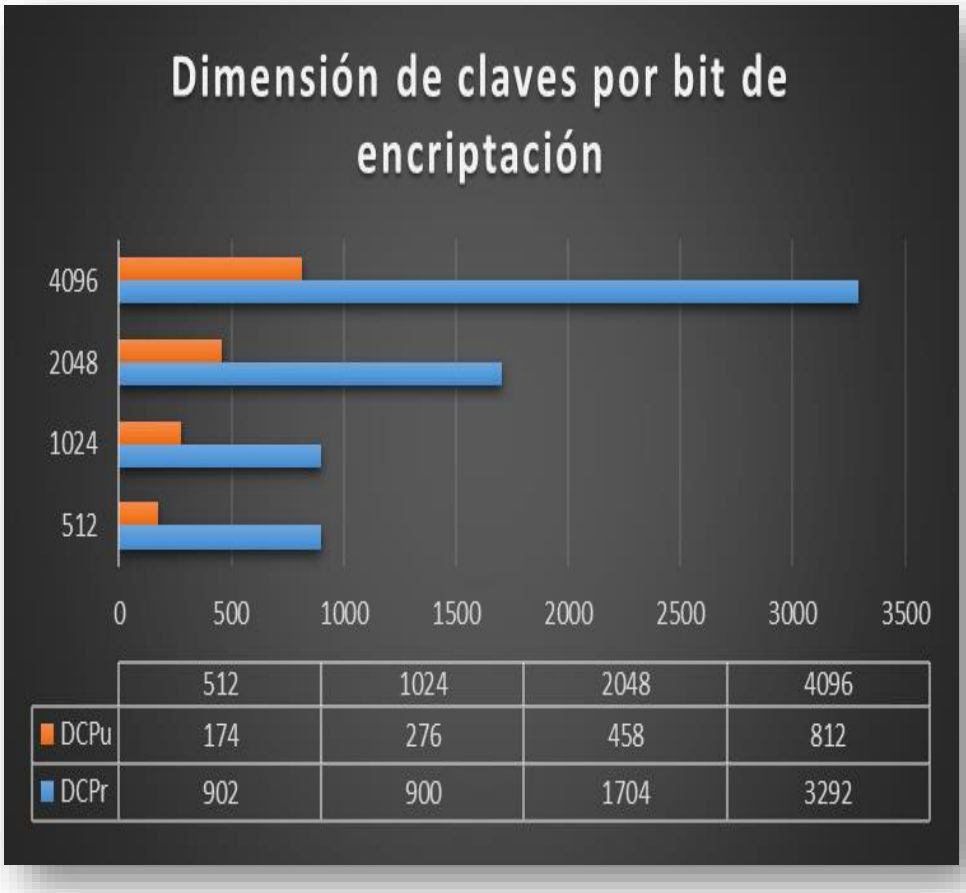
Para la medición del grado de seguridad entre tramas de datos JSON encriptadas y sin encriptar, se realizó un análisis de forma cuantitativa de diversos parámetros que se consideran esenciales.

A. JSON sin Encriptar.

Para el análisis de JSON SIN ENCRIPTAR se verifico que es compatible con la mayoría de los dispositivos existentes en el mercado actual. Por el contrario, al ser un envío JSON sin ningún tipo de cifrado, es inexistente el manejo de claves para encriptación punto a punto, la trama es completamente vulnerable y la robustez queda bastante deficiente.

B. JSON con Encriptación RSA.

Para el análisis de JSON ENCRIPTADO RSA, se verifico que es compatible con la mayoría de los dispositivos del mercado actual. Además de poder manejar completamente claves públicas y privadas para la encriptación punto a punto, generar una trama invulnerable a posibles ataques o manipulaciones y presentar una robustez que satisface muy a detalle los diferentes tipos de tramas de datos. Pero se notó que esto generó un mayor consumo de recursos tanto en el dispositivo como en el servidor.



	512	1024	2048	4096
DCPu	174	276	458	812
DCPr	902	900	1704	3292

Figura 3. Dimensión de Claves por Bit de Encriptación.

Asimismo se calculó las medidas de las dimensiones de las claves públicas y privadas por bit (Figura 3) obteniendo que a mayor bit, mayor es la dimensión de la clave, además, se puede observar (Tabla 3) que, por cada bit soportado, varía la dimensión de las claves, tanto privada como pública, lo que genera que a mayor bit, mayor es la seguridad del fichero.

Tabla 3.

Dimensión de Claves por Bit.

BIT	CLAVE PRIVADA	DIMENSIÓN DE CARACTERES	CLAVE PÚBLICA	DIMENSIÓN DE CARACTERES
512	-----BEGIN RSA PRIVATE KEY----- MIICXQIBAAKBgQDUuWbRWvjEX++jcq+GaUlvsZEvlYGRw3/fbbk1rAPZanhKQHA/ 4lTIIZV2mSJbZchOsQy7BcCgIvolCCODS9iH7g/Y1udykwd9UbJdjLWvAjkzqEww 1cg93964AvrkgRyYFiQEmUPM8mU1NVWXWAzE0719r9n+XkMdw1/FzpeRFQIDAQAB AoGBALhIsZj2rYI59jlr1kGkHpljg/psE6Z6beHahVPI5EDKqtDwiA3f/kCuT3RB 73CseV4qpXOz5Q8VngzMwy8wWrC6C7p9IVIKoydnQ8sqR7bZ6pLJQ/KjL1r	902	-----BEGIN PUBLIC KEY----- MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBALC004OAFa8vibepsSGapV01yHV/q+Uz PhYAXbN3n94bkWHqEnWuSoVw5z4zUsqFT8towU+ENJ8JRjqI014RTz0CAwEAAQ== -----END PUBLIC KEY-----	184

WvJXr

4Yd+aweJhuByCdMGrMieGxpiWQM0
Qm4/ymUXs9tY0D0RY6KVAkEA+1Qf
O3JK3C50

v7WEJ+jBTBznsEKUPiGMSl69OXI1w
uPudBcU25jK2E/Ax2WeaNs6HshZJbq
Eh9nk

HdiJvdEDUwJBANitl2E/OM28FwB0FX
6GaUq8M6OP54ubicqZzxMbDYzSlT2
dVpQi

Vt1lzeldPxfAHDm38H2oN5qgRpIp8qot
tPcCQAXk7iqVvp21VTWdyzCoa8geM
WzX

0uQ2IzUnTabIP+yHLfN5CxsvVSqHjEz
g3Ds3UQCh0D1rR2psOMZearSR71M
CQQDF

zmu1HpdH0l8ZTeBVenVjAb8OoY/bN
moraBDPu1G6boBKGHnykArG4WqCk
VrI3tul

oZNN06S3Z3wlcWAUFdfvAkAdPYBzh
35Hj0u57cSlqlOGeBFSpOzjRnzS/Ibpq
HUJ

pEYpEhj3BoRpY7aGSUnTk4vLF4Gkt1

	f9M8a05XJr5/QQ -----END RSA PRIVATE KEY-----			
1024	-----BEGIN RSA PRIVATE KEY----- MIICXAIBAAKBgQDD5S0xzOrcS7mPtpMFkVDAElwVujK84ZZzHqVvFS3i+ewKGzVA 4fHEKr3x/jUOZocCzGUPn2Q3m31L3NfFKZVbHHiat3kKVMKhwoCGHi/w/lEMz5Tk 9rH6its/EFYWwlfZWqbUc5gGfguHODTpeeKQmsyVoj4ZSPxwAWouBBitYQIDAQAB AoGAV3ILParSwGoROGTwtlCK8mJ/trc6hPRSm6r3gGG87RBRXKWOnRr/OGk4BzlJ 59a+t/ITvuiQ7P287ct0ii+o2ozIXP02YcxA/+nP9Rh+FKjqzsnyt9sv91ZWNqxZ c6h5dN99mfUb38yUqMXESfB9XWKQjnKX3jXH90Kck2xmdV0CQQD71hQfraEnoIuS MGp3wuxKWGnMzcA6d+ON78L6ZVt	900	-----BEGIN PUBLIC KEY----- MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDD5S0xzOrcS7mPtpMFkVDAElwV ujK84ZZzHqVvFS3i+ewKGzVA4fHEKr3x/jUOZocCzGUPn2Q3m31L3NfFKZVbHHia t3kKVMKhwoCGHi/w/lEMz5Tk9rH6its/EFYWwlfZWqbUc5gGfguHODTpeeKQmsyV oj4ZSPxwAWouBBitYQIDAQAB -----END PUBLIC KEY-----	276

q9jesaR1MT/ctbI1NHyBP4nDEMCrfSb
JQSojZ

/YlLtldPAkEAxyJSddj7pdCkm2eGiQW
sBic10o60HYbfQKLhbATyGn5QKP2so
QNb

/ijiHOJsDyaemckq7tzWpnRyY58ZbYG
ETwJBAJrWBs1rTZg1p1qclflISUt0jKg1

IKlSKQCNl42oX8ql05kXj8HpsImnIkF2j
8FN8hNNoi5i+1VOXlHZBl8cnxUCQA6
f

Cob0o6WNKEjw1QQCkprxDhaju73fR
U0tqHeR9WTUmXYpCh/Ya0zCG7gbN
+v03fhU

uP5I4MngCvYdnHRZ6ikCQCgwPFvEx
vRSH+B2t6smsykJlQNrNR9HxO2RCg
AkiYr/

50U8fwbuSS+UcGjAZ7N8lCrkRZqsIZ
Yo/dxT38/Vvns=

-----END RSA PRIVATE KEY-----

2048	-----BEGIN RSA PRIVATE KEY----- MIIEpAIBAAKCAQEAlIw0ZFxv9hJfN+M3rRHwAsQ9cwyQPkvRvS1cy0r7/6txJOwo FfBQCwl9LtsNeO5qv1kAvRjiMuJ87Z4UmdUxCTZ24G8PvuOZSForYi9Sqx8CSd6b 3IXHu5HNldhLOce3NJaeOiHg0Dlz8U0oRfK37mXwY/3inzZkU3duMlhnOvHU2g7V uGSVP3ppKCc6UrV5FD8r1Tat+eLPqK0WJZh3gTXV6kYKKVJrCzUoMTYicsShzB8m oaz6CdaReJhnY1uKnY2a0CXyCtXLlOSsOsV/YxW4GYL5Nr1SI001GciUPBvB/n4/ gd2omiT9b0nEiwMQo4+bxobmgw6fkM1hmk2tDwIDAQABAoIBAES6gdsVca/WXDvN YwsuS2/cP9olK+/GwKnYfSCJ9wIZ3mWI5hsQbthMcaDmNNuI49bYZSEYjD+izs5Z 34o1O1TZ4DKC8DQPtfHl1kC9SHBoD	1704	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAOCAQ8AMIIBCgKCAQEAlIw0ZFxv9hJfN+M3rRHw AsQ9cwyQPkvRvS1cy0r7/6txJOwoFfBQCwl9LtsNeO5qv1kAvRjiMuJ87Z4UmdUx CTZ24G8PvuOZSForYi9Sqx8CSd6b3IXHu5HNldhLOce3NJaeOiHg0Dlz8U0oRfK3 7mXwY/3inzZkU3duMlhnOvHU2g7VuGSVP3ppKCc6UrV5FD8r1Tat+eLPqK0WJZh3 gTXV6kYKKVJrCzUoMTYicsShzB8moaz6CdaReJhnY1uKnY2a0CXyCtXLlOSsOsV/ YxW4GYL5Nr1SI001GciUPBvB/n4/gd2omiT9b0nEiwMQo4+bxobmgw6fkM1hmk2t DwIDAQAB -----END PUBLIC KEY-----	458

nUDC/YB56F/GjGEO3nmT3uDXH3lAt
EUY/J3

EnUakcM5YvnAk8O6NLzxewjMIicA29
b9TrEHl2blxVNVHgm91eKjr3aLgcG7P
TSF

Dm9+H8Yv13zQuoAWT1sVOJNOZ0c
+MOwigJkGOML7SQJVsJw51Xry4kR
2oNnbAy3k

H/8vmq7devhtjS/NBzY/A5RGl9tNZAdj
UA3cbDpkEXWZfRBxofWMdcmzv3lA2
g+d

D9B5xpECgYEA0i/ScyKV046u8Tkbuh
F43updLxn1WzRuV8+Vdt2cF4/3mvXt
1n1N

gXc8TT4gVufIfui0vrRg/tlhm4TrmUn/A
UaHrYVs3c8Q5/wBXBlOwTuqEbXMN
2cH

GNAeq5hBzKEmf/8lLuqCRf2RBludGh
DB8MMAPOtYMb9Zr8CrUiTf60MCgY
EAtOz9

vis9xlAmSxf4sikxZLebFZ5fdhXN5wHA
C9mROdzzNgkHAW5QH/RIHk0TgrgS
z+B3

Lpra72BPTmr29TkrfJ5ca7OkNuUqNb

QUUMLkVJDAHL2Blo5uUTUBcAgnDh
pUyMmK

l3njS5cFKpSJPBTIsbk690NAUxc4KM
N+RsvRbEUCgYEAryLJ6MXYsf8iJj2/r
LsC

TDxciPLITsbRarm9gN4HQlrjpmtlUmP
2QPppdLNfnwpjNdwRPxZt++yGOF/5
MDFI

Gnf6dcvHesse+y4ws90ikkB05WzmTQ
O8FiDUbYbgzPE6c0zWXtFo24j9Xp7X
uPBd

qY/SkvBG/KzEvn18Q2YWbqECgYEAr
vDMtXnIRJ4xOMEYBuUPzow+V5wIP/
z7WBf4

Iz0fUGzVenUmIfsRXKRuq2XprKbiAcD
QXO79LSLZ7EA/RIQgCWJyiO1tpmrV
8pGK

Z0DbWO2R5PhUekiEYN844aPwZ5G
MU42UlZMA9ZE0PKTdQYvzNu2dF4M
einlUIw5M

ZpYjjGUCgYBv2128ipdVIvVNJn5uNW/
48k4/vjPxx8nwJrT1UagF6oHlJ7irAvYF

La7YbNvpORmUMvykntocLjeL2Gwny
Qrj4D2sJwJqeIHRI75U+wAi2/DDDkn/

	FQge qvl1QmshByb886cW6aaoxTmrHesyM X509oFDEqYpXTEfxKvPpu+hFw== -----END RSA PRIVATE KEY-----			
4096	-----BEGIN RSA PRIVATE KEY----- MIIJJwIBAAKCAgEAgz4SjGH232DBrq ifVqbLF/2HqMTKvk6RDuKRN4Dgovzb eaTF tUl5WpGZEJ5DkkphGbubwCTM81q8F HduLRci3bQ+42rgZeXPo/SCNyAQihD MkxUJ 5uHEIe7DYTPMpWXFQAmdiMQKYa Q3kkFFhs3941kAISELvfdaE/OSSYrg/ 6u51Tjk QTlg2L7XOCN0qpVvOvymXrzSG84T0 F3wz/OszhWiUa38DCIvhwh5N/oyGitq 5IRb 02NyUi5B/IOpzXq95GeUIPTov8JjtOm NIOLMDIIEAZL0r3C/9wadNuARdgF9D	3292	-----BEGIN PUBLIC KEY----- MIICIjANBgkqhkiG9w0BAQEFAAO CAg8AMIICCgKCAgEAgz4SjGH23 2DBrqifVqbL F/2HqMTKvk6RDuKRN4Dgovzbea TFtUl5WpGZEJ5DkkphGbubwCT M81q8FHduLRci 3bQ+42rgZeXPo/SCNyAQihDMkx UJ5uHEIe7DYTPMpWXFQAmdiM QKYaQ3kkFFhs39 41kAISELvfdaE/OSSYrg/6u51TjkQ Tlg2L7XOCN0qpVvOvymXrzSG84 T0F3wz/Os zhWiUa38DCIvhwh5N/oyGitq5IRb 02NyUi5B/IOpzXq95GeUIPTov8Jjt	812

u9w

OUTroOVuxL3qSpCXo5dIRavA+YIIm
HqWs9FQvYRV+JRYVUUwJV8KktHQ
4DoJHJGt

veAaO1piOSG07U1TlGeo0MuE/gwO0
h/WRecdcrBacieu6jG3xU7V0Jg82Exx
TsR1

+3jDzYF5nXh3iPc900vD+DEnHp2ghP
vcajCcwVKbgaPmx5n68WT64hwkeWx
cJkU6

t/UIH03LrqG8OS681thc/K7zzpFvqIKJ
Wf8NaZyMXsGIAGtxYdHGQ+BQqYch
mOAi

gg+zQuD/+AqnborFS6x1OFkM8J1zBf/
Ww/jasy5JtlxiMFOAVvFowTX+9BqDH
EvU

4Y8JZnx2vnVq4FtwH4x9eo9Oyy9HbU
MuaD//nsQh25DQ2FyDc9gXcz7yAzU
CAwEA

AQKCAgALw92/zexmm/Lg6GbYSr18G
hM2MuYF518jmXXxJtF8avR11CqRFN
QXJWQ2

VCm0jIxl1SaiM+pXinO/1fD/g8pxW34jv
un82Hwjs4CU1oAFEHp2WPV5SBXgA
OmNIOLM

DIIEAZL0r3C/9wadNuARdgF9Du9
wOUTroOVuxL3qSpCXo5dIRavA+
YIImHqWs9FQ

vYRV+JRYVUUwJV8KktHQ4DoJH
JGtveAaO1piOSG07U1TlGeo0Mu
E/gwO0h/WRecd

crBacieu6jG3xU7V0Jg82ExxTsR1
+3jDzYF5nXh3iPc900vD+DEnHp2
ghPvcajCc

wVKbgaPmx5n68WT64hwkeWxcJ
kU6t/UIH03LrqG8OS681thc/K7zzp
FvqIKJWf8N

aZyMXsGIAGtxYdHGQ+BQqYchm
OAigg+zQuD/+AqnborFS6x1OFkM
8J1zBf/Ww/ja

sy5JtlxiMFOAVvFowTX+9BqDHEv
U4Y8JZnx2vnVq4FtwH4x9eo9Oyy
9HbUMuaD//

nsQh25DQ2FyDc9gXcz7yAzUCA
wEAAQ==

-----END PUBLIC KEY-----

Nps

9UrMLPJgyjvgDcrsc98zcjx8WC0BW3IihibaUbk2fqhHhXakFM3k6mOz4SPoajs6

4hrMS/cbr5hud7jSG0g1bqZuzeZY1yGk9tbYc5wVG9JaDUi+hdPgRmlA6vJNoa3y

qA7px64MMqjZx3qx7ggoKKOpJguqt2EKjxT2QUJEL8iaKb4G+R0v4JEL4TzKQKcM

ZgYBxzKAzMgCzuHOvDRvDtZBqmr1X59bq+uiqgjadWtBvy6p0xpy9VxNh15AyTP+

br3IjRGyhEfhSPXILrnKAcuG36T0PY8vq0t60UT/mvprDDvLbM91cIoK4iVgC3AQ

kRjRKw2UNk/rV1hQDv+AAd8R4yEvo7YXmPQfwP4J4I16PdlMODywsJFnl5ktSgC6

yCGO4lJ8VjH6EoPuG3//XEUQ1M4l+bDH3O2A3Xf1t+q6Rz9x6risWuWrU+JUhuQz

EqLZd7FHFualJED5M9GuW6M3zxnrCaBuzj83uMNtL8/9vGRphKv+ZS1V3Qk

PF3m5

I3N24OULwTcO7HqjMGOBVzm927m
bh3XpruvHE37h0xgvhIixoQKCAQEA/1
CgIjK4

k3tVApZz/kNemYzDH/yhwMIy9/o5qQb
YjOGNC1NkY5dPupj0+LimCNeekSKz
NqdJ

T6wEf8iJLE0XoX37Nq5Qz/uLO/5RF7
RSfxkiI/7ceqHaVBHC2rIqfncLh8cHQE
hm

cC80FXbYx+2NbbQoHdprX+zLG8jMi
MUk95ZRxLai/bTBKCxz/sVQ/IzfLQxZy
EEf

R4nIEzUfHs6NnvFv+PPzfuozLRpY/a4
o2He9Xu2XmjjlU4rjoH4Xg8jfkcSwBiko

W7k7iTyqVhzNKwuiPKwKVOWMv5Yq
EjEO56tQc88xon21NYL5uXjU0aCaYJ
dAibTC

DJZ8x1Gv3F/gaQKCAQEAg5g44sYH
Hg2vMGmGS+5fQb4wDoFFGMYNM4
ea9ObCOmZI

HQCwporCu4WbHK2b5UCPpOK43Rw
agqA3gbTHMnf7NJt4KA5enRfRSnzpE

JGdDJzQ

tUAiUyRtO9nWPqldtcLDN7v5zIO5hve
UwrYRozEJTAxkPM+JrHkE1qKBf74V
OQ03

yiBCk5JtQZrtNxmGHiEUI0yIqyHDVidX
hD6xMo1uYt48h8CyJEzE4C43yU5TO
Xtn

3MVIr5J8+8hnguDH5TLZmddRbfhH5
HfCl3yBAVeW1Lu2JBZGt8/rqs1LACei
HV0p

HCGe1JvD4VXfARmINlBom22fTijaxw
9f/m1eFizy7QKCAQBUeqQRzMCgJB3
z1j6Z

Mkd5/UaPLfODqX5d6RDtKdO7geI+jiB
Yh7ec4YGlTsu5f0b7dQThn6a1UxDGC
1/A

Jd4c1GesJy+cMfpXJoZElqj5XyKZ0Hm
5phZs3Nehr3Q6Fj/ddHCkx5AFrMjKF
WIT

fTnfw83GWwBOZDasgduToc/ikH2L3n
a+BfInzdWvxjBO8HnkGQMYLNwhuqh
WGqyq

b+QSZIcNVAMMyoeWc/JgNiFTxwMF
Bw6zvAG9i+ZXs4EqXOwE+pTX5QRy

nS7i8QzX

o34LsTTEaJSelLnmptRS4qKw61GakE
M3Jeswwp7IzP92QXLzZPi41/OhYKX2
+cN7

4smJAoIBADmuIVEVJ81+hgmcV2c05r
/0UzAobVb8iJDLoZozA3OBnZ5IzSVS
sSRs

3Z5mYmzMLapCTnFsfho/i9Tonl96w7s
9Yh98nPASnoQd2DKRANukaL7pi6IR
12di

hkwVBKRE6DinxkhkfI+yOZxXHcgAG
GkjDVSyS1A0A3Do7gZGFRtatdd86H
Gyvv0A

oXWDVIwkR6VpSIZqWCqNOAkZSaY
04+umKbX/OpY6qBjzaTRthoEtnEriloe
OmAXs

w6JUUIogDc0m2eRv7WMxMzQfOJSU
BeLdHGsY8IplMpE9RuCcfqyMRxazje
D3HZbJ

guQCgvf7qr5mAGvL8llMLfFhktn5fckC
ggEAa1Xx36mK1MlmCKkOZngnK/Fjs
9Ki

81aTQ/A2lo/1taNv4Tz8yTtTlNa2NrRJ
C9IUGgHbSJhBN2Ek328w8mSrmhKc

1pus

/jT5iK/LdoBWz1//NiSedxYCWmDPgHI
GeKlVygvM/nWMJCTZc3bksmdrqYU9
eawn

lNu1urj1siUk1iCOxYn97nv5sfLECF0T2
5cN70GlLZEiTkc35LArbAN1LAaqW33
Y

90NxANJJ8wT2lhaUtR+n2MUqIuQ5i3
3iS5NL6aBwP3tOc/yte0OnaMflGcoeE
mW4

jKiFTn/2x4VUxxPnMHzJWYtt58xjH5uF
yaeiJbNXHpXCUkFks92WegxKPQ==

-----END RSA PRIVATE KEY-----

3.1.2. Fortaleza de Clave.

Para la medición de la fortaleza de la clave entre tramas de datos JSON encriptadas y sin encriptar, se realizó un análisis de forma cuantitativa de diversos parámetros que se consideran esenciales.

A. JSON sin Encriptar.

Para el análisis de JSON SIN ENCRIPTAR se verificó que el formato de manera independiente no genera una clave de seguridad.

B. JSON Encriptado Con RSA.

Para el análisis de JSON ENCRIPTADO se verifico que es soporta encriptación de 512, 1024, 2048 y 4096 bits, pudiendo de esta forma generar 2 claves, privada y pública, las cuales varían en cada solicitud del fichero y son mucho más extensas y de esta manera poder soportas mayor cantidad de caracteres el momento de realizar la encriptación.

Tabla 4.

Similaridad y Diferencia de Cifras en Claves por Bit.

Bits	Dimensión de Clave Publica	Cantidad de cifras		Dimensión de Clave Privada	Cantidad de cifras	
		Similares	Diferentes		Similares	Diferentes
512	424	5	419	128	2	126
1024	812	4	808	216	3	213
2048	1592	2	1590	416	5	411
4096	3128	5	3123	756	5	751

En la Tabla 4 se puede observar los valores de la dimensión por de las claves tanto pública como privada por cada bit de cifrado, además de la cantidad de cifras similares y diferentes de las mismas.

Figura 4. Variación de Claves por Bit.

Como se puede observar, la variación de las claves por bit es mayor al 98% (Figura 04) y las claves varían por petición.

Tabla 5.

Claves Generadas por Petición y Bit.

BIT	N° PETICION	CLAVE PRIVADA	CLAVE PUBLICA
512	1	-----BEGIN RSA PRIVATE KEY----- MIIBOQIBAAJBAIimrk/rBoiotAeDBliwi0hT4Sb1MBhrnEJSCB+ixHM6aw0FDaIIax3gzsvO4rIA6yBk6jukOObDlJ6z0DDujSECAwEAAQJAIgMar9mXrrwO0y/RjJGkmUQHbmLoratbIUa5ruYcb3Flw4gCemVyQExTEAJbqeqjjymdktVH23n2ZoSo7p8KAQIhAL7OSMSx4O4SYNyijWnYASm9TbKT6D4F3r5WmiyoEgEpAiEAt1eBDkSGtZQf WAqJOuMw4sFspNC+sy0zZykqmM+sUzkCIFiMRvzxR990K/t4fIQYOEH0ljA/r9HIy5wteJMs1F4xAiBmQs5e8OXaHZYl+ypzyAiaUZyRiUT0l3+WSKXTufG6MQIgVTVy q2EipbGEkLZMY+yQhIpXj28fkldprZrGeaWrD0A= -----END RSA PRIVATE KEY-----	-----BEGIN PUBLIC KEY----- MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAIimrk/rBoiotAeDBliwi0hT4Sb1MBhr nEJSCB+ixHM6aw0FDaIIax3gzsvO4rIA6yBk6jukOObDlJ6z0DDujSECAwEAAQ== -----END PUBLIC KEY-----
1024	1	-----BEGIN RSA PRIVATE KEY----- MIICWwIBAAKBgHJnTnSGxnF6Ybp29lqKJfm+G8ODvLofQI14PxILqRfm0ZIcQjQkdObXYzL55auShKfRKUu8fvNCNLx7utEu/blab3o0p234+vQ2InnsY76zYJWWk+IOU5AygazDuTZz5/XS0yoLnqlz2G2phAXpi8Cwzp5ndFDF4RvKp03QYPgTAgMBAAECgYBIGKyLqOfkWcC+9vt3wXn/MnxisWG9VbajsPTmVWzhwDewQZjDLXpyg6hjVrMTdgmdjKXesdK7UKzVX5AzBc8lKK/0qeme1dnTrCxqon7pHNbkwt0RQVLNJQI973zdmdjKXesdK7UKzVX5AzBc8lKK/0qeme1dnTrCxqon7pHNbkwt0RQVLNJQI973z -----END RSA PRIVATE KEY-----	-----BEGIN PUBLIC KEY----- MIGeMA0GCSqGSIb3DQEBAQUAA4GMADCBiAKBgHJnTnSGxnF6Ybp29lqKJfm+G8OD vLofQI14PxILqRfm0ZIcQjQkdObXYzL55auShKfRKUu8fvNCNLx7utEu/blab3o0 p234+vQ2InnsY76zYJWWk+IOU5AygazDuTZz5/XS0yoLnqlz2G2phAXpi8Cwzp5n dFDF4RvKp03QYPgTAgMBAAE= -----END PUBLIC KEY-----

		6 Xcl+xpQvch1g8kqq/GLoX5lAwTCHHHa6NjrgazyAxii66QJBALtdXHYcjOqYa6c7 HnwrjDJCswHC1oBRtUXVJkw9FtnaEqTydJKF+ThAKLukT8rMBGqiym/Z/j/1ILE+ RRen6Y8CQQCcT9AXc1bUOjjVnU/0CZZ3q5uS8IUVknE5c3m+O+DB93IbiwwAN Din xYjQTuS/XTjdhzJbP88l+4KQhOh/ih89AkBrU5aUXdRmybrZB6d1z2g2njh2KMrA pI1nHCsc/t1pG18Ut4IJMM9K5o2utveyHl3lS60Ul1XqAxugDCB95YSHAkEAhTeI U9Qa1jjgoz21HHjnYgpBRtPxuFbcZGhEA5sojGcFOvAgRDQ9bM7vMQlr7qPWiQZ m JlqVgvt9ZQnA1KG0yQJAIOgXaPySj00y0eYORsRBUs9YBkiummdiNr2V83YRzR7 + U8Id28RZjjoeim0pEZcK/FPVk/bzBi/chRWuAyLaIw== -----END RSA PRIVATE KEY-----	
2048	1	-----BEGIN RSA PRIVATE KEY----- MIIEpAIBAAKCAQEAy05y1Aa3giopWkjwaM/FK1+1OuuD09WwByn8kjUQCkloUZ 14 BArxQIoZoS1ORvv+YpiPnGE3tzxDUwTrzdKXj2jbFGE9D3LJLIZ2Io8MZkvJEOKI WWzk3tpoWml6Xzwd5qW9K/5WoVFzUtoRhwkhnvywYCjGH3MfF60nWRWH+iW9 P5Sw 9yVPC+FqxNNTqPxUt8K+1v/qOjl/4FXDoUQwJOfNJHM7Mzi1oznWKxvFUU5d6da o fYWJMlOSBb51rAhAV4Fr0YoTyJXcYyTAVgYVjAFcWJ4f4A3qOTrfoFLM2j4rJBK6 l6fQ6twvtK1BzQmjOU0L2er9U99CRnmTAcoYcQIDAQABAoIBAAzDJe4tZcOkBei	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAy05y1Aa3giopWkjw aM/F K1+1OuuD09WwByn8kjUQCkloUZ14BArxQIoZoS1ORvv+YpiPnGE3tzxDUwTr zdKX j2jbFGE9D3LJLIZ2Io8MZkvJEOKIWWzk3tpoWml6Xzwd5qW9K/5WoVFzUtoRh wkh nvywYCjGH3MfF60nWRWH+iW9P5Sw9yVPC+FqxNNTqPxUt8K+1v/qOjl/4FXD oUQw JOfNJHM7Mzi1oznWKxvFUU5d6daofYWJMlOSBb51rAhAV4Fr0YoTyJXcYyTA

G

/6B0unnYjjXepu3bUiuA9TvwXMtohEXrTqdiUHautKEriOhn1TY3QXydlbPI8qj5

fBv1hlIYUPhcikq39+z1m+DbffnF9B89OTXz4gIjT72tNWNZEL9z1AY49xM1W7nt

un/xkMy8LA49QQKkZWVFF10+PsTzELNcuvcfoTgPeRBvLDMEt+w57HmBiUrlS+o5

qa2zxCJymtgtqJuXgyJGL2gNfIIzgOHxH/lzPa/1yF7DfjXxMfowpY6c+JfBeMRi

BEM2e5rXGOIIZ8/tpHCjekAdQyXxa8TeaRDcUhHrHdy6/EC/Nxjy3b7+0O2h5chP

fIF+AykCgYEA/+U+01W8SA5DoTOWSjpqir32RBA8YilF6ke/FuzS0naz/bIJaa77

fwbEr3Wi/vapsjZ9R7/M+Ch1vq72tkqTSwYvPk+cPXk233iVd7DzwVe4Jiwwg7gb

QSLbhHwP8cK5hFav74F1w1B8MxJmVjws9xtPgYKH51gzkJcbOmXdwZsCgYEAy2O0

bgFTmZ2etogjfAqTmOHpzpGfJDpbk6JyVx7igQMIi3JhzfQ9SukJ5Hwq9OtByZqV

Ht6gGlwpuKwKBTvotGQy9K20I+xtRHxetPeJoOjS37uX7wViRaHD29n+NCAR7Hpd

UknB3O1Psngovcima6PneA4sdKn6YoUZmOPzBOMCgYEAlADyimcPJBnj8cuVhE8v

Khne4tUFJ705a9ydYFQPR51SU0xy90Y2veybO+RtJsxhr+qmISaFFY7JM0Wc+ig9

8umbz41aD5AZNb1XrceR5GSo5KMwWC7WzKCtmH3s9uaqSCmfToZxJ0tY2m0s9XmO

hEmufXiSxDKWeG19WRRr4zsCgYBFeGPboyZb9QOOE0gTiUVK1fdXcFPsOuamsbje

WxVAvbmAb+xKPZ9l491WJrdBes24v80pgx8qgcpaYqD1krX0T7k768z47DuSRezP

YTGoPX4xn1DQ+oTLhepxQK9cPIwCuvT/M2Q9Qu2S5wn7cPqvD4fzcNvsaVa1qg

VgYV

jAFcWJ4f4A3qOTrfoFLM2j4rJBK6l6fQ6twvtK1BzQmjOU0L2er9U99CRnmTAcoY

cQIDAQAB

-----END PUBLIC KEY-----

		LJ Pyr1wwKBgQDox4MN+/I0co1S/cjbJKLj+ewryV5subPSXtTAZIZEgjgjFbIv2R36 EnvJyPoTvWfkAZpKp7GFixq8B6ElCFfppQxG7EN0k12jkFoNJn4imdhflq8Z0/6d umqDrxsjuT2D8G0weCyXUz55+tjrOtSjYGhrYBcC4p8F3T+pC/Anpg== -----END RSA PRIVATE KEY-----	
4096	1	-----BEGIN RSA PRIVATE KEY----- MIIJJgIBAAKCAgB4F8uNK3v/pNgA0N7m+FdC1G4G55DaA84Zn78rl0z51VNuRDgS TpcMK99Kswj2WN8MwOyl3FMJqh4deXBQO59Lvfo4I54nGm7qmrBrlAy+Vevbmm0d GKKJ1rf7MYrc3aaQo8fcZJtuMbLFeH643T6XgVYOMLmykQRaJW2CHMenPu2BR67M dr1DHa4pJRQ8UEEUs829Jj8wvwL5YPQKRaq760qVxzDaoMTaITeEmcU4LAiUI6ya 0lS4ayegpczwKwxiflOlfQ3tPfGalsBJaDiLXxMr8glRC4hubf6M+awMVqcqSe2K RBErxMThHBQ1yTzgam0sNLA+4Mq4z0UVx8Dfp9SbR4+CStqjVNjMgGpKxvlBknil xdgbZCvRyUNW3a9LCDb74AoKCCRrjrVALPM7gBl0SBIbESPQfhFIo6ZwbG81d9j8 sHFuYWoWP6VA3KQlSBU6hsWWn2kpCcIPX3e8FbxUTQsvZtR0J4uuhC6QyHWItIts 7wOzyvcRyTRz8TMnDEyuI4qh8yrzkDqP3OGEWzgPQPrUEWdVqXPE3e+rNL+MKfZg isWucnYD6YRCLG6t3RrAHEQ9iAt7YYi59tik/qfBMYEmBEYVoTS5jl+pYSHekQN6	-----BEGIN PUBLIC KEY----- MIICITANBgkqhkiG9w0BAQEFAAOCAg4AMIICCQKCAgB4F8uNK3v/pNgA0N7m+FdC 1G4G55DaA84Zn78rl0z51VNuRDgSTpcMK99Kswj2WN8MwOyl3FMJqh4deXBQO59L vfo4I54nGm7qmrBrlAy+Vevbmm0dGKKJ1rf7MYrc3aaQo8fcZJtuMbLFeH643T6X gVYOMLmykQRaJW2CHMenPu2BR67Mdr1DHa4pJRQ8UEEUs829Jj8wvwL5YPQKRaq7 60qVxzDaoMTaITeEmcU4LAiUI6ya0lS4ayegpczwKwxiflOlfQ3tPfGalsBJaDiL XxMr8glRC4hubf6M+awMVqcqSe2KRBErxMThHBQ1yTzgam0sNLA+4Mq4z0UVx8Df p9SbR4+CStqjVNjMgGpKxvlBknilxdgbZCvRyUNW3a9LCDb74AoKCCRrjrVALPM7 gBl0SBIbESPQfhFIo6ZwbG81d9j8sHFuYWoWP6VA3KQlSBU6hsWWn2kpCcIPX3e8 FbxUTQsvZtR0J4uuhC6QyHWItIts7wOzyvcRyTRz8TMnDEyuI4qh8yrzkDqP3OGE WzgPQPrUEWdVqXPE3e+rNL+MKfZgisWucnYD6YRCLG6t3RrAHEQ9iAt7YYi

NB58o58VjG+Dnzyca3yx+lIz1s51dMTTkAM5dK9X5R7zy46rIiqeKVE7bQIDAQAB

AoICAD8Qv+QyobtuZcCH74Z7g9ekluMpV/glZ2YtbccGfXX1gQq26Hs0qj3KDEi+

HEbCBOA9sAip2zGIFKiFzWVU6iJscappifQe/YBWG69i4A5UMLfyMLrG5RRdneBb

9ABopb/vQlWKv9TuZGMvE90g0m3djcOF8v7DCPYOEaPyWSaK9U6exjy5XkqQm5Uf

07CYtJSal8BtBWIj7Pvo3j9GtpJYzaI6KH2nT7j9GJqZ5efJt/eThGEJc1QlcXn7

l9PHx4kxrduKAHZFwUrdP71M7eTC2N6LDDlzRQI44SRHOQKBvueGSygA9sN1Ya6V

gEd+lmmDejFsueFLSyoiLrVxYOdVSUpsMVPHhbLPPoBvommW2L7a5q8phCpqigBj

TajdChryS1BtknruY/6Y3v+wp0zEmF9oHinlNscV6QbIQOOKJLHNXrkLHQOaW6L0

7Ecee5I0Kv2Ksi1Or4uB3c/s8PSNN+BYF3czvJOcUcnZuQw1evIOCyojb0l6Mc0k

g0KVrQvhWwN7fUSHawo7ANrZsNhOtVh74sxpONM4maTU+pQUb8+dFoFCbSxdDhwl

9qCo8Ce4tPlgoBbmBn0VGcfO/IE5xBUDG5nom7uLQEm1M0yQ6KtCoLwnrx7W4qPd

FW4QaMdmPSm3h4fTld61YT90VfFprpIGV2Cfpe+xrGoIuWIBAoIBAQDfAcuoEX9s

c2mQRGeJLrtNvybiL+vs6HyV/xNGsclk4bZIZX8WleSCAvBA7JGtm62C8Oy158p5

I/WLxs10TsjX6h7ZwGhuSSOtLswhlsOdNmN1KbbGI1U58Rmine7mONRLHD8CwOxk

SRMP4P3aNgclhasfH56SNF/KkIW1d629HkFgibVVwLpw8D/GtBoNZGYfIFrF8Tdc

4Qje/H1kF1YcmP3Kwhh8f0krj3CPwfmblOIZxT+xHO7OlwMx6lS5znfnNYwNN/WH

MobVouJ0S64dNGD6s6qksBmL/MXaHuTpwYR5G/g2tTJ3Y6kYZhEaACTrlXgQex

59tik

/qfBMYEmBEYVoTS5jl+pYSHekQN6NB58o58VjG+Dnzyca3yx+lIz1s51dMTTkAM5

dK9X5R7zy46rIiqeKVE7bQIDAQAB

-----END PUBLIC KEY-----

qL

xJT5a6x+/iWZAoIBAQCJ3Dcf8L3GSrRQ402nHWGViPfoxlSQjO2aQGtWySkuoo60

0onP+lTinOwIp4gtaHlPmNlzAGgiK4+ijOp0MdbdpwmhL70+x6lieW9dYMH1Q0ON

MsqkjGqRpZEePCeI1H7VmemctX++M7rh1QNpj/+1sCMFbzJYVhNOY9/Q96Ix0ra
s

ET9UvtX6jltu/eg5UxdV0THpoxQ5qax3xjRxoy9BTPe5A7xiqvjHW7x/0qh+oq/3

fKRRu6mbIC8k2zwc/idzhheSm5HPoWRfxgz5fYi9/18JEkoVHplRdsqFfxJY9KsO

T/HnbWeF/rdUonFe4hIJ/mn4RkBh5dJiI8f42UD1AoIBAAHLEucsFNk7aFrnoemB

fbIXWmmxNvV1vSK7pV0JKvRruDqGzmnRFMBS47mx7U33FYMbCtw/m+ozdtMD
6z0H

MShDUrIvkOdqISZQXQXcH4CRImWby0N2n0XTAYDEQlqJlHNyK7OK95uq1sNm
7MOn

cZFErOaYK39Deo23ccvOxY/iwWXUVaf138nutKSVaCw5ZXTUnsY+tVJC4P5iC2Q
V

p/Sc6lWKEAZYQ69c/Ou7BJXSsarBYky7TdRNdJIfL0U7Os2aKaEllQzcRCdNowKK

ZUEI3crGTRX78IimpLSnUPHsQ3Nx3SIbfOdm3sI6d+K8o49daVTAmKsNDqGe0wl
G

Y+kCggEAKO7VA0vU3UKjJX1IOj10ExVXHqt8wmrFxc4o0NzruGgV3UTCHJmkiK
C1

SKCNw4XIt0SYFBhj0vmCLW6P0q5VsPx+1Hg7RZ+9AHs0ANLQ5ETzksIo/xCeN9
6k

JwkPaAZIHp4Tii8t3ci4TYOU/sv9zfGdhGiO10G/e8AV3ZMZepepG5AHcN80469P

ILaNNDAitp1lp4x/XduuVOXp/m/IgSgcSbf9QIKpWXA4mO9wjiE6BE7i7028P04g

EV5zC9y+1ANVjndYpQCzr+g7Q/ztFBT2H9hpXLn4JgMSgO7JANasz5QXLHMEkP

mf

b6bH6LdRSdxblJcvWKIHhJJBNvcy3QKCAQAIsIgHdjOihtRdFvl9TT3L6/uIoDEl

t6uNvoaRj1FlN8au3X6LXFd3nIBHFLKiUga42e94dxrbv+vCGmoFNofehd0R5BLI

Va2vxxzXkR1/fgYE8e/aROcRnoK8GPONuvJlCa1RkxtVG5TZq03bKXGxgudhiFjs

OqA4C6uKWOuC0kA3O6cEVCXl38lYbHtm1ljz7wLlhiGNjfkdFYVMOhKCXhbFXm
o1

HZwNSAbMyWqHB1jC9QCDZM+o/0KGJvEuvcCBz+R5uJ3lzhrP/Lyb1oBtaRwylQ
Jb

LNetB0eW9T94VhimXPfmWkIpBXnosFYF1p5wXh8jiS0WlAeNrV1gcKwi

-----END RSA PRIVATE KEY-----

3.1.3. Tiempo de Velocidad en Cifrado y Descifrado.

A. Tiempo de Ejecución en JSON sin Encriptación.

Para el análisis del Tiempo de velocidad en cifrado y descifrado se llevaron a cabo pruebas que consistieron en los siguientes criterios:

i. Comprobar los tiempos empleados para envío y devolución de una respuesta por parte del servidor NodeJS. Esta prueba tiene como intervalo de tiempo 10 minutos, en los que se realizaron peticiones consecutivas con texto aleatorio, generado con LOREM IPSUM.

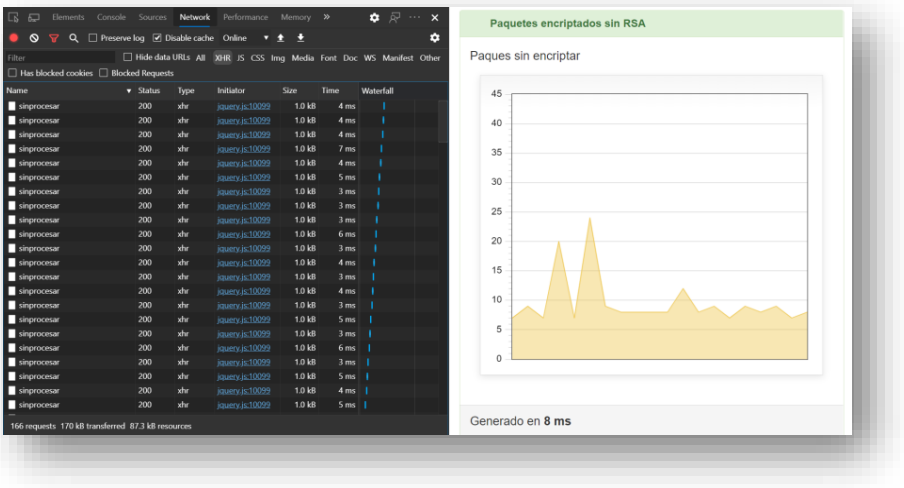


Figura 5. Tiempos Empleados Para Envío y Devolución de una respuesta.

En el cuadro se puede observar en envío secuencial de paquetes Json sin procesar realizado una petición Ajax con contenido generado de manera aleatoria. Luego en el cuadro subsecuente se muestra 10 paquetes escogidos al azar para realizar una estimación del tiempo promedio de envío del paquete desde el cliente y el tiempo de confirmación por parte del servidor.

Tabla 6

Envío Secuencial de Paquetes Json sin Procesar.

NRO. PAQUETE	HORA DE CIFRADO	HORA DE DESCIFRADO	RESULTADO (MS)
1	1:35:14.300	1:35:14.307	7
2	1:35:33.560	1:35:33.564	4
3	1:36:45.432	1:36:45.439	7
4	*1:38:10.567*	1:38:10.573	6
5	*1:39:23.212*	1:39:23.218	6
6	*1:40:45.887*	1:40:45.893	6
7	*1:41:23.346*	1:41:23.354	8
8	*1:42:32.778*	1:42:32.787	9
9	*1:43:44.078*	1:43:44.089	11
10	*1:45:12.192*	1:45:12.199	7
		Total	**71**
		Promedio	**7**

B. Tiempo de Ejecución en JSON con Encriptación RSA.

Para el análisis del Tiempo de velocidad en cifrado y descifrado se llevaron a cabo pruebas que consistieron en los siguientes criterios.

i. Comprobar los tiempos empleados para el cifrado, envío, descifrado y devolución de una respuesta por parte del servidor NodeJS. Esta prueba tiene como intervalo de tiempo 10 minutos, en los que se realizaron peticiones consecutivas con texto aleatorio, generado con LOREM IPSUM.

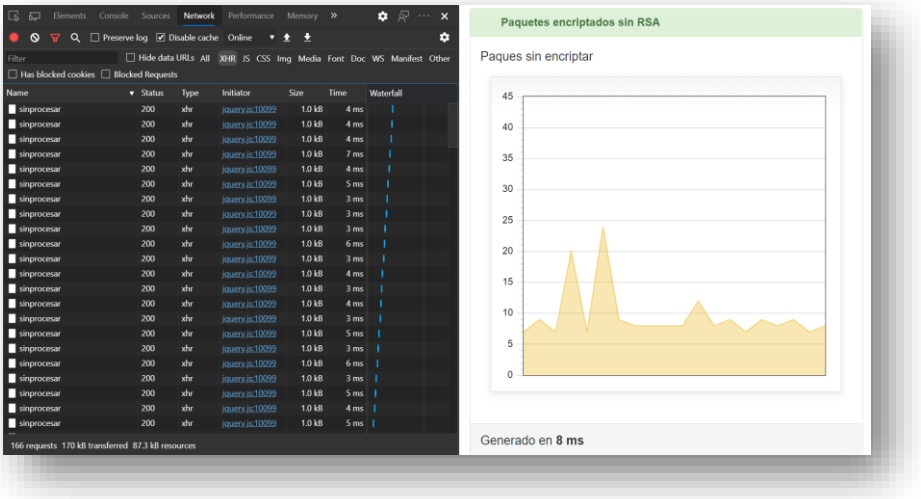


Figura 6. Tiempos Empleados Para el Cifrado, Envío, Descifrado y Devolución de una Respuesta.

En el cuadro se puede observar en envío secuencial de paquetes Json sin procesar realizado una petición Ajax con contenido generado de manera aleatoria. Luego en el cuadro subsecuente se muestra 10 paquetes escogidos al azar para realizar una estimación del tiempo promedio de envío del paquete desde el cliente y el tiempo de confirmación por parte del servidor.

Tabla 7.

Envío Secuencial de Paquetes Json Procesado.

NRO. PAQUETE	HORA DE CIFRADO	HORA DE DESCIFRADO	RESULTADO (MS)
1	2:15:14.670	2:15:15.490	820
2	2:15:33.663	2:15:34.286	623
3	2:16:51.421	2:16:51.422	768
4	2:18:11.731	2:18:12.531	800
5	2:19:45.275	2:19:45.787	512
6	2:20:25.187	2:20:25.832	645
7	2:21:23.346	2:21:24.058	712
8	2:22:35.118	2:22:35.719	601
9	2:23:87.224	2:23:87.813	589
10	2:25:14.218	2:25:14.645	427
		Total	**6497**
		Promedio	**650**

Según cuadro estadístico podemos observar lo siguiente:

	1	2	3	4	5	6	7	8	9	10
TECE	820	623	768	800	512	645	712	601	589	427
TESE	7	4	7	6	6	6	8	9	11	7

Figura 7 . Tiempo de Ejecución en Ms.

En el gráfico (Figura 7) se analiza la diferencia de los promedios obtenidos con las formas de envío de datos para poder demostrar un aumento de tiempo en el cifrado y descifrado en un JSON encriptado con RSA, lo que nos conlleva a lo siguiente:

Tabla 8.

Intercambio de Datos con Json.

JSON SIN ENCRIPTAR	JSON CON ENCRIPTACION RSA	DIFERENCIA
7 ms	650 ms	643 ms

Se pu9ede apreciar en la tabla (Tabla) que el intercambio de datos con JSON sin encriptar tiene un tiempo promedio de 7 ms (milisegundos) mientras que en el JSON

encriptado con RSA muestra un tiempo en promedio de 650 ms (milisegundos)

3.1.4. Resumen comparativo de Seguridad, Fortaleza y Tiempo de los formatos JSON encriptados y sin encriptar

Tabla 9.

Comparativa de factores críticos

Tipo Formato	Seguridad	Fortaleza	Tiempo
JSON sin encriptar	No hay manejo de claves para encriptación punto a punto. La trama es completamente vulnerable y la robustez queda bastante deficiente	Formato de manera independiente no genera una clave de seguridad	Los tiempos empleados para el cifrado, envío, descifrado y devolución de una respuesta por parte del servidor NodeJS tardó 7 ms.
JSON Encriptado	Maneja claves públicas y privadas para la encriptación punto a punto. Generar trama invulnerable a posibles ataques o manipulaciones. Robustez que satisface muy a detalle los diferentes tipos de tramas de datos	Soporta encriptación de 512, 1024, 2048 y 4096 bits, pudiendo de esta forma generar 2 claves, privada y pública.	Los tiempos empleados para el cifrado, envío, descifrado y devolución de una respuesta por parte del servidor NodeJS tardó 650 ms. A mayor seguridad y fortaleza, un mayor tiempo de respuesta.

3.2. Aporte Práctico.

3.2.1. Realizar el Análisis de las Estructuras Algorítmicas de Json y Json Encriptado.

La presente investigación ha analizado las estructuras de Json (Figura 8) y Json encriptado (Figura 9) con la finalidad de conocer el resultado de codificar y decodificar una trama de datos.

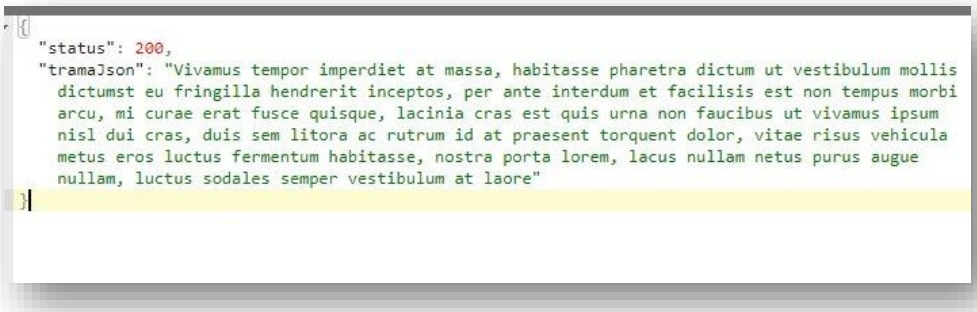


Figura 8. Formato Json sin Encriptar.

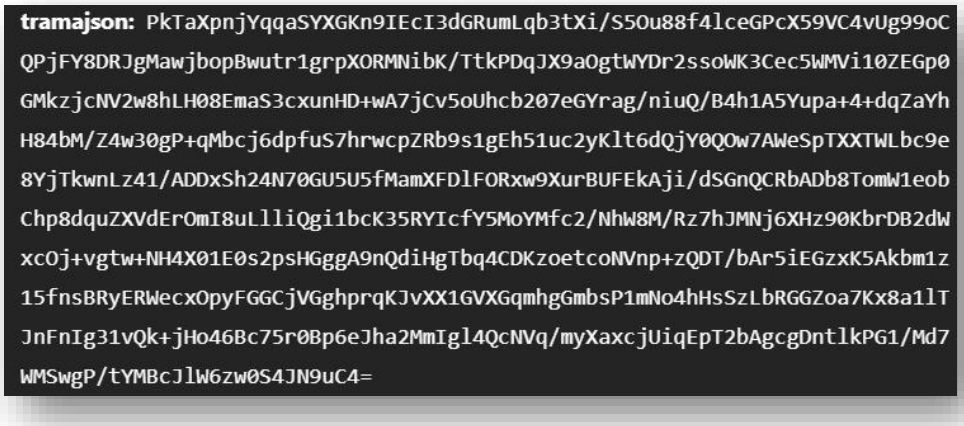


Figura 9. Formato Json con Encriptación.

3.2.2. Establecer del Escenario para Realizar las Pruebas.

Product Backlog.

Elaboración del Product Backlog del proyecto junto a su historia de usuario.

Tabla 10.

Product Backlog del Escenario.

PRODUCT BACKLOG	HISTORIA
PLATAFORMA WEB	Construir una plataforma web para comparar e rendimiento de datos con Json encriptado.

Construyendo el Product Backlog

Listado de requerimientos, construidos y solicitados por el Product Owner, además de mencionar los requerimientos no funcionales.

Tabla 11.

Requerimientos funcionales.

PRODUCT BACKLOG
Gestionar claves
Gestionar tiempo de cifrado

Requerimientos no funcionales.

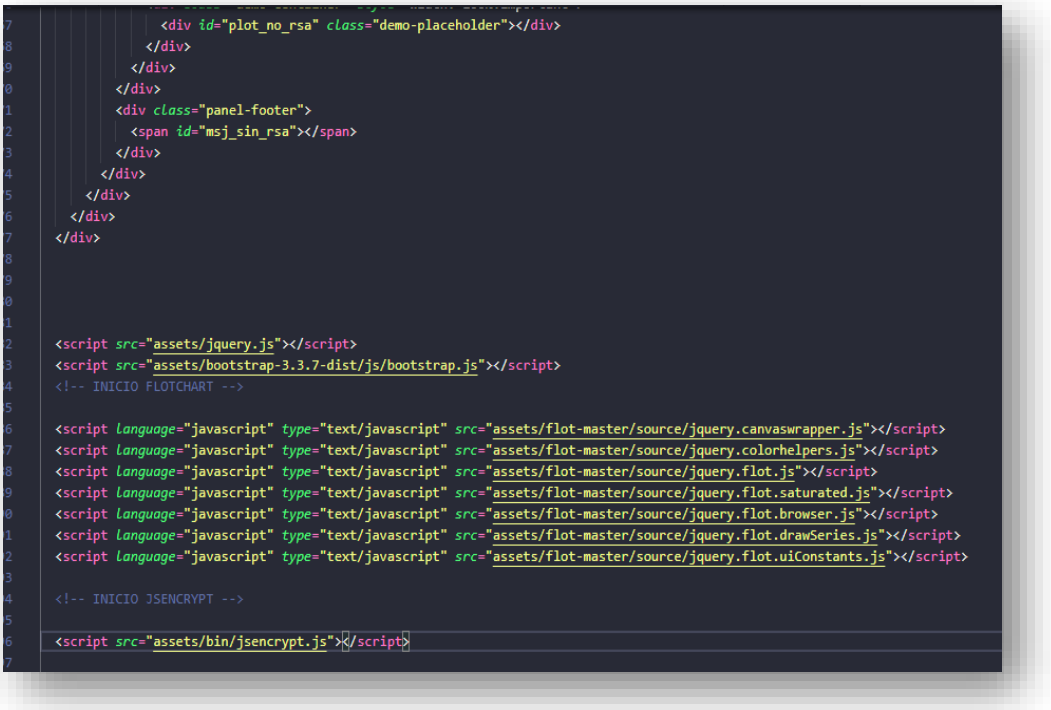
- El sistema web será desarrollado en Node y JS.
- La aplicación podrá ser abierta desde cualquier navegador.

Una vez capturados los requerimientos se procede al desarrollo del sistema que nos permitirá realizar las comparaciones correspondientes.

A. Implementación de Servidor y cliente.

a. Cliente.

Se implementa directamente en los assets, permite generar tanto clave privada como pública, se invoca generando una nueva instancia dentro de archivo JavaScript



```
                <div id="plot_no_rsa" class="demo-placeholder"></div>
              </div>
            </div>
          </div>
          <div class="panel-footer">
            <span id="msj_sin_rsa"></span>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>




<script src="assets/jquery.js"></script>
<script src="assets/bootstrap-3.3.7-dist/js/bootstrap.js"></script>
<!-- INICIO FLOTCHART -->

<script language="javascript" type="text/javascript" src="assets/flot-master/source/jquery.canvaswrapper.js"></script>
<script language="javascript" type="text/javascript" src="assets/flot-master/source/jquery.colorhelpers.js"></script>
<script language="javascript" type="text/javascript" src="assets/flot-master/source/jquery.flot.js"></script>
<script language="javascript" type="text/javascript" src="assets/flot-master/source/jquery.flot.saturated.js"></script>
<script language="javascript" type="text/javascript" src="assets/flot-master/source/jquery.flot.browser.js"></script>
<script language="javascript" type="text/javascript" src="assets/flot-master/source/jquery.flot.drawSeries.js"></script>
<script language="javascript" type="text/javascript" src="assets/flot-master/source/jquery.flot.uiConstants.js"></script>

<!-- INICIO JSENCRYPT -->

<script src="assets/bin/jsencrypt.js"></script>
```

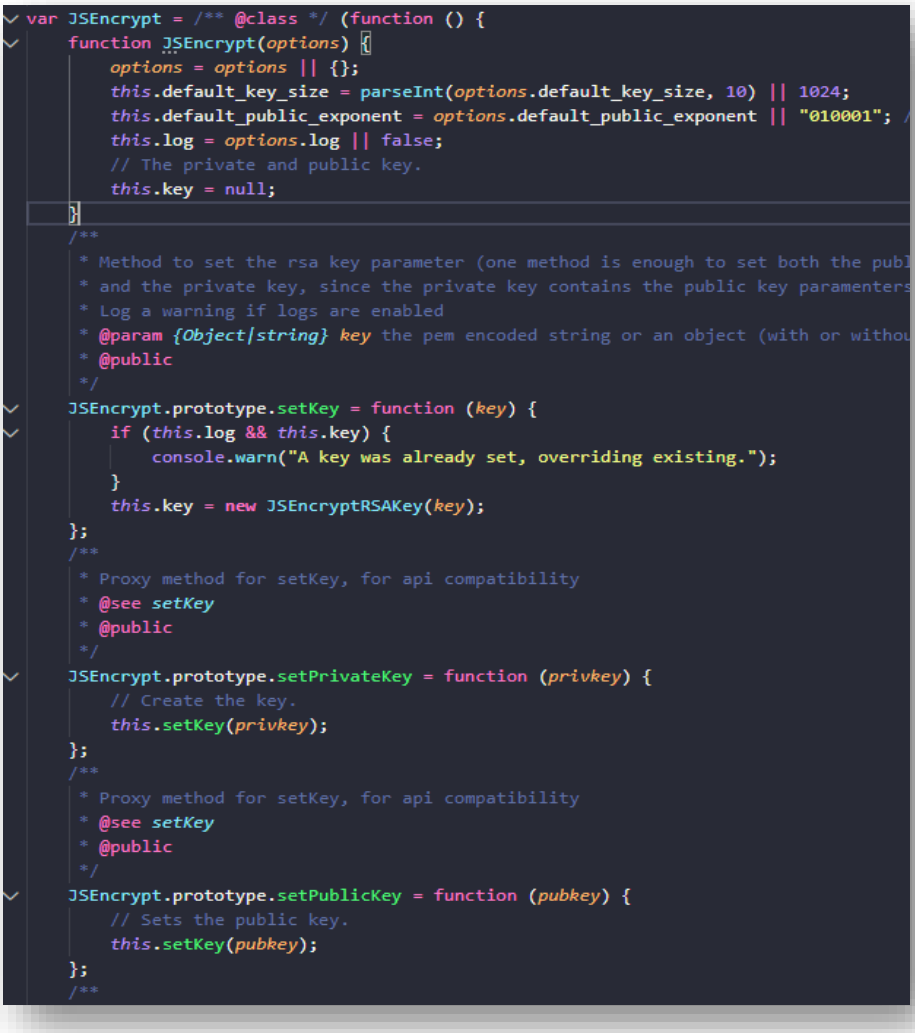
Figura 10. Invocación de Script de los Assets.

Se invoca una nueva instancia del algoritmo base y se indica que tamaño tomara la para generación de claves privadas y públicas.

```
var generateKeys = function () {
  var sKeySize = $('#key-size').attr('data-value');
  var keySize = parseInt(sKeySize);
  var crypt = new JSEncrypt({default_key_size: keySize});
  var async = $('#async-ck').is(':checked');
  var dt = new Date();
  var time = -(dt.getTime());
  if (async) {
    $('#time-report').text('.');
    var load = setInterval(function () {
      var text = $('#time-report').text();
      $('#time-report').text(text + '.');
    }, 500);
    crypt.getKey(function () {
      clearInterval(load);
      dt = new Date();
      time += (dt.getTime());
      $('#time-report').text('Generated in ' + time + ' ms');
      $('#privkey').val(crypt.getPrivateKey());
      $('#pubkey').val(crypt.getPublicKey());
    });
    return;
  }
  crypt.getKey();
```

Figura 11. Invocación de Estancia del Algoritmo Base.

La manera de generar las claves, dentro de plugin, se recepciona el tamaño asignado, en caos no asignar uno, toma por defecto 1024.esta se envía en el state del algoritmo, para poder generar todas las operaciones que se van a realiza (Figura 12).



```
var JSEncrypt = /** @class */ (function () {
    function JSEncrypt(options) {
        options = options || {};
        this.default_key_size = parseInt(options.default_key_size, 10) || 1024;
        this.default_public_exponent = options.default_public_exponent || "010001"; /
        this.log = options.log || false;
        // The private and public key.
        this.key = null;
    }
    /**
     * Method to set the rsa key parameter (one method is enough to set both the publ
     * and the private key, since the private key contains the public key paramenters
     * Log a warning if logs are enabled
     * @param {Object|string} key the pem encoded string or an object (with or withou
     * @public
     */
    JSEncrypt.prototype.setKey = function (key) {
        if (this.log && this.key) {
            console.warn("A key was already set, overriding existing.");
        }
        this.key = new JSEncryptRSAKey(key);
    };
    /**
     * Proxy method for setKey, for api compatibility
     * @see setKey
     * @public
     */
    JSEncrypt.prototype.setPrivateKey = function (privkey) {
        // Create the key.
        this.setKey(privkey);
    };
    /**
     * Proxy method for setKey, for api compatibility
     * @see setKey
     * @public
     */
    JSEncrypt.prototype.setPublicKey = function (pubkey) {
        // Sets the public key.
        this.setKey(pubkey);
    };
    /**
```

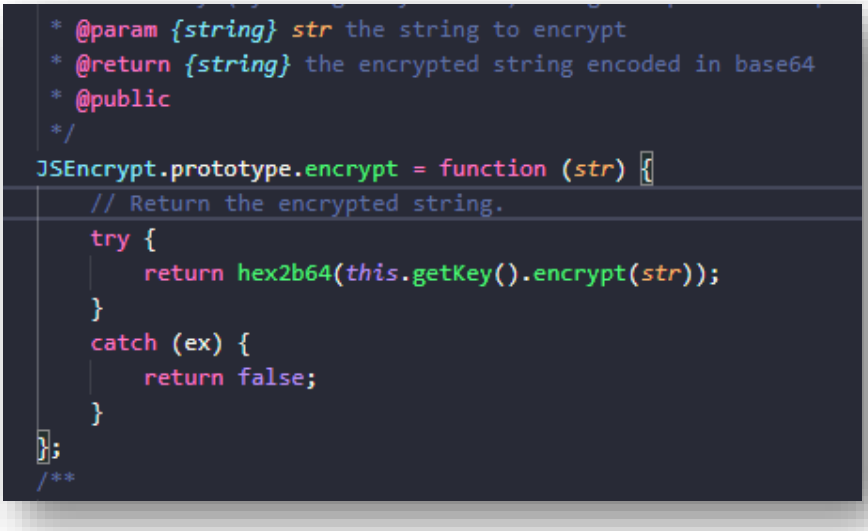
Figura 12. Generación de claves.

Una vez instanciadas las claves se codifican en BASE64 de tal manera que sea legible al navegador y de esta manera sea más sencillo trabajar (Figura 13).

```
 */
JSEncryptRSAKey.prototype.getPrivateKey = function () {
    var key = "-----BEGIN RSA PRIVATE KEY-----\n";
    key += JSEncryptRSAKey.wordwrap(this.getPrivateBaseKeyB64()) + "\n";
    key += "-----END RSA PRIVATE KEY-----";
    return key;
};
/**
 * Retrieve the pem encoded public key
 * @returns {string} the pem encoded public key with header/footer
 * @public
 */
JSEncryptRSAKey.prototype.getPublicKey = function () {
    var key = "-----BEGIN PUBLIC KEY-----\n";
    key += JSEncryptRSAKey.wordwrap(this.getPublicBaseKeyB64()) + "\n";
    key += "-----END PUBLIC KEY-----";
    return key;
};
/**
```

Figura 13. Codificación en Base 64.

Después de generar las claves, se toma el contenido en Json y en conjunto con la clave pública se genera la encriptación de todo el texto a través de una función HEX2B64, que permite convertir la encriptación generada en Hexadecimal a una codificación base 64 (Figura 14). Una vez hecho esto, la trama queda lista para poder ser enviada y decriptada en el servidor.



```
 * @param {string} str the string to encrypt
 * @return {string} the encrypted string encoded in base64
 * @public
 */
JSEncrypt.prototype.encrypt = function (str) {
    // Return the encrypted string.
    try {
        return hex2b64(this.getKey().encrypt(str));
    }
    catch (ex) {
        return false;
    }
};
/**
```

Figura 14. Generación de Encriptación.

B. Servidor.

La implementación en el Servidor se genera a través de NodeJS, este motor permite generar Apis con JavaScript, como primer paso se genera una instancia de algoritmo a través de NPM en NODE_MODULES.

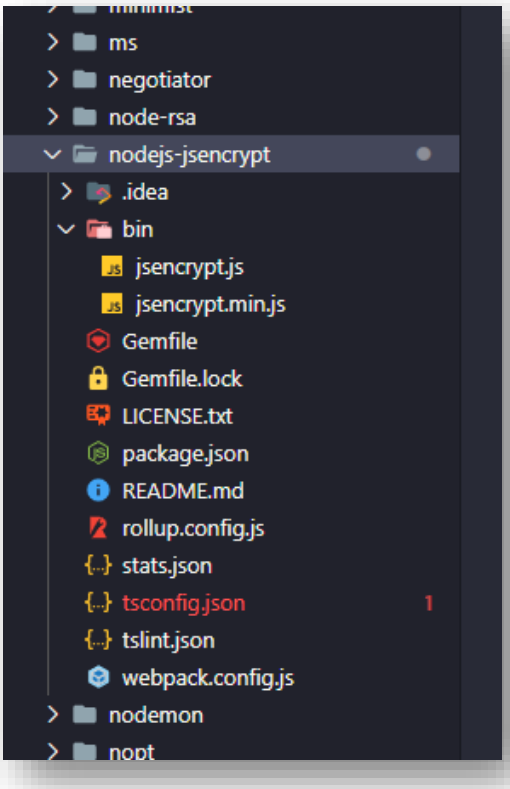
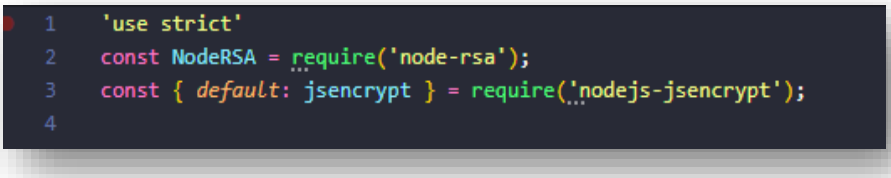


Figura 15. Implementación del Servidor.

Después de esto, se genera un controlador que permita exportar el algoritmo como una CONSTANTE y de esta forma reutilizarla según sea necesario.



```
'use strict'
const NodeRSA = require('node-rsa');
const { default: jsencrypt } = require('nodejs-jsencrypt');

```

Figura 16. Generación del Controlador.

Una vez realizado se toma la clave privada compartida entre la aplicación del lado del cliente como el API, cabe resaltar que la única clave que se comparte es la privada, debido a que esta es la que usa para decryptar.

```
var static_private_key = `-----BEGIN RSA PRIVATE KEY-----
MIIJJwIBAAKCAgEAgUTmM6h8ZdVUQAc5Qh82/hrBEykMwbmipnuwsOPyzboCnQld
5o8XCfjKARBvq7qJONopE1F71PEK46hru+eaf3PEMOp06OcAu9WD4XxWy0s8iVcK
DQVtgFZfMZiLh0KBTne9GCHkVkXZ741NJVJsrxjzsn9xufQJntqy88Yd0QWMW0nr
3gJeRg3FKoDhT4qPjgmawnSarDuMXG6b/oTxj2x7dFHnWWsLzV8XSbhKpldlvdc8
Q3q9Wg2K/6FgWYb2jYGEbTkeBjGkiQ3RGo3uJ0ZRNwVxrfXt5365Dy6542CrFMj4
zHwkfliEC/uWqorbve1ddkAmUWgcH5WoXoTzNHevbd3QEOVCIJGKztBodKWlJoAO
e0NpqjNLqWCTlKG7/Yb9vMi3qc9oDK2wMONO2GfTJN4OLgGpGE5k3GjR4sMF8Q4M
geSheN4QlOZPBiPQ5KpfXgY0dRTsM0TwPTritrXSWJV347MZkTW8CAFsQwgQdcsR
m/IHKArqYKvhh6c05v58XrkDSnt+KKlj0Mz1sZj1mTIdwo9RkG6Ozjze0yZxW8yz
pTWEYJzBDXw7lO0nvfD8EIyJ6MDyY1/3Zc0WL0qN94W8RhCt24Xx40YCio+d+2tX
RQjDPV/B8QhNI/qwwDBjBOw6BmDtiGGZcL6oEVe+nQVlBc/ZcyXLVMLU4BcCAwEA
AQKCAgB6pUcPrgRQjaVAllnx7TG8xt92cVUEHD9AW0xZl/wOyjOekHuBuzLtqBjo
B5MbyIBW601bYczuTL+7HVkRz3ya/dgfXOMM8daVX8+fSNwER16pI2aK0iKBR/FL
WeLPsv6B10MWOVmtrNpH2LVf+qLZQgA804OwHORfzjlI5if136C6IBlj6rVNG9V8
```

Figura 17. Clave Privada.

Una vez tomada la clave privada, se instancia en la constante del algoritmo y se invoca a la función decrypt, que, usando la clave privada asignada, descifrará el Json encriptado decepcionado a través de la petición en el API.

```
encrypt.setPrivateKey(static_private_key);

var decryptedString = encrypt.decrypt(tramajson);
console.log(decryptedString);

res.send({'status' :200, 'tramaJson' : decryptedString})
```

Figura 18. Instancia de la Constante del Algoritmo.

Por último se implementa la interfaz de la plataforma web para realizar la generación de claves y realización de la encriptación (Figura 19), además del monitoreo del envío de paquetes con y sin encriptación (

Figura 20).

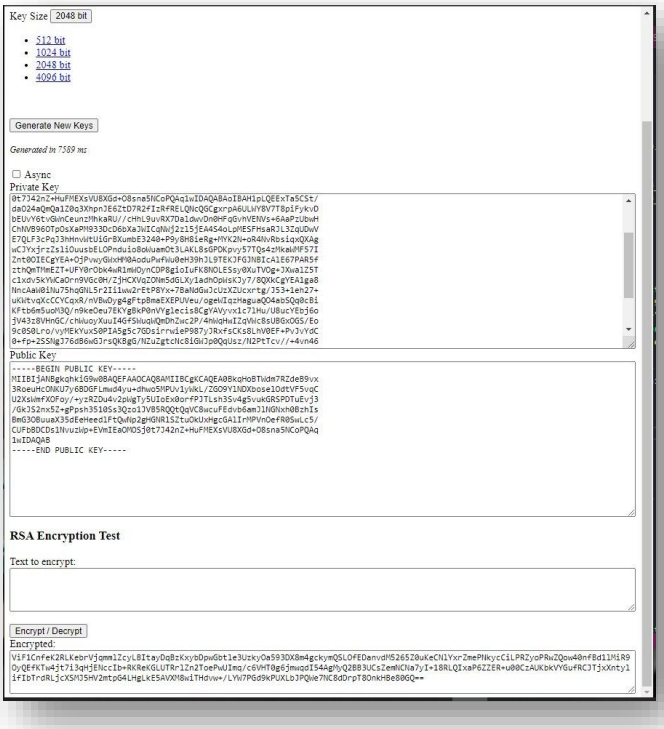


Figura 19. Interfaz Para la Generación Claves y Encriptado.

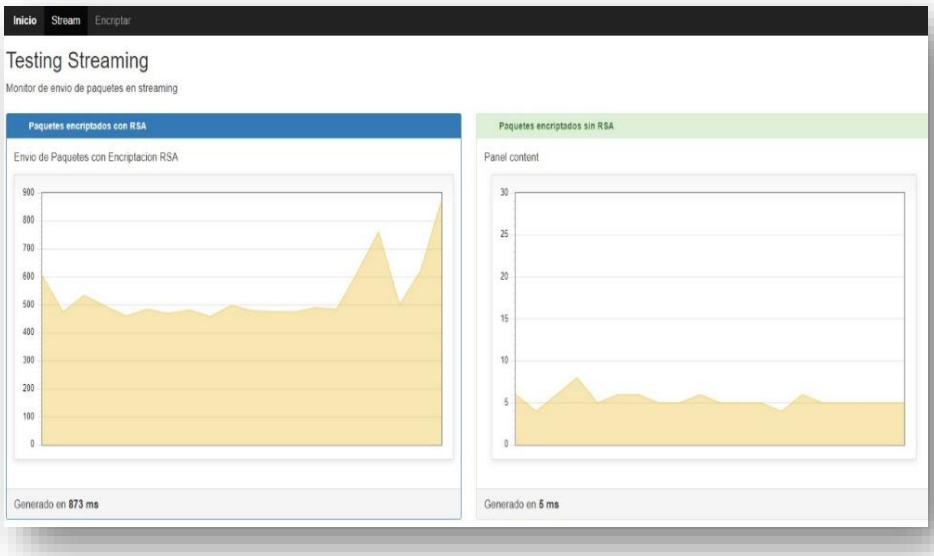


Figura 20. Interfaz Para el Monitoreo de Paquetes

3.2.3. Estimar del rendimiento en el intercambio de datos.

Respecto al rendimiento en el intercambio de datos, en base a lo

investigado en el apartado (1.3) y el software realizado, se estimó una diferencia de 643ms (Tabla 8.

Intercambio de Datos con JW) que es más del 100%, en el envío de paquetes con una clave privada y pública de 4098 bits.

Tabla 12.

Intercambio de Datos con JSON.

JSON SIN ENCRIPTAR	JSON CON ENCRIPTACION RSA	DIFERENCIA
7 ms	650 ms	643 ms

Se puede apreciar en el cuadro que el intercambio de datos con JSON sin encriptar tiene un tiempo promedio de 7 ms (milisegundos) mientras que en el JSON encriptado con RSA muestra un tiempo en promedio de 650 ms (milisegundos)

3.2.4. Comparar del Rendimiento de las Estructuras Algorítmicas.

El software permite realizar la comparación del rendimiento en el envío de paquetes en streaming, donde se puede visualizar mediante gráfica el tiempo de generación con encriptación RSA (Figura 21) y sin encriptación RSA (Figura 22).

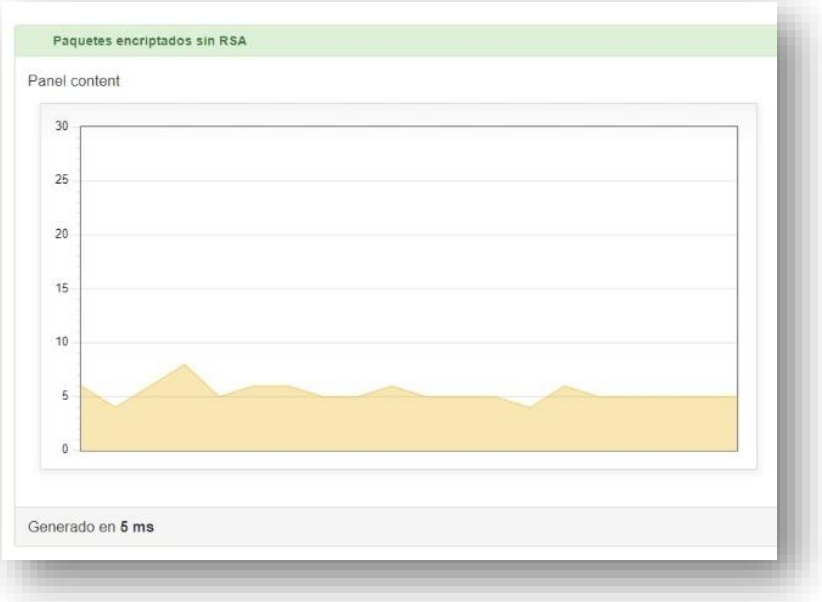


Figura 21. Gráfico de Rendimiento de Paquete sin Encriptación RSA.

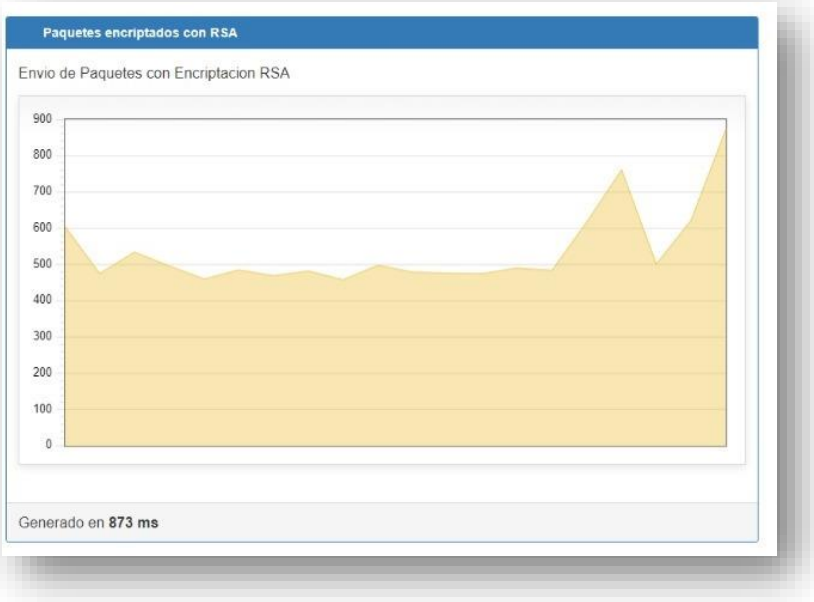


Figura 22. Gráfico de Rendimiento de Paquete con Encriptación RSA.

3.2.5. Comparación entre RSA, MD5 y HASH.

En la siguiente tabla comparamos las diferencias entre los algoritmos RSA, MD5 y Hash, y por qué motivo elegí RSA.

Tabla 13.

Cuadro Comparativo.

Criterio Comparativo	RSA	MD5	Hash
Definición	Es un algoritmo de cifrado asimétrico	Algoritmo que genera un código asociado a un archivo de texto	Código que es generado por un algoritmo, de tipo MD5 o SHA
Tipo de Encriptación	Encriptación con cifrado asimétrico	No encripta	No encripta
Modo de Operación	Genera 2 claves, pública y privada, añadiendo una capa de protección.	Compacta los datos y genera una salida compleja.	Compacta los datos y genera una salida compleja.
Descifrado	Revierte el cifrado con la clave privada	No revierte cifrado	No revierte cifrado
Seguridad	Es un algoritmo seguro y confiable	Actualmente está roto, tiene fallos de seguridad	Si proviene de MD5, tiene graves fallos de seguridad
Fortaleza	Las claves publicas y privadas ofrecen una capa de fortaleza durante el cifrado y descifrado	El cifrado no es simétrico ni asimétrico, solo genera un código, lo que hace que su fortaleza sea menor	El cifrado no es simétrico ni asimétrico, solo genera un código, lo que hace que su fortaleza sea menor
Tiempo de Respuesta	Es mayor en milisegundos debido a la complejidad del cifrado y descifrado. Actualmente no representa un problema por la	Su velocidad es mayor en milisegundos ante una misma petición.	Su velocidad es mayor en milisegundos ante una misma petición.

	velocidad actual de los procesadores y memoria RAM utilizados		
Ventajas	Al ser asimétrico, utiliza un algoritmo sumamente complejo. Puede cifrarse y descifrarse.	El valor de salida siempre tiene la misma longitud (40 caracteres)	El valor de salida siempre tiene la misma longitud
Desventajas	El tiempo de respuesta es mayor.	El tiempo de respuesta es menor. Su fortaleza y seguridad es menor a RSA	El tiempo de respuesta es menor. Su fortaleza y seguridad es menor a RSA.

El motivo de utilizar el algoritmo asimétrico RSA es por su fortaleza y seguridad que brinda durante la encriptación. Esto es compensable ante el tiempo en milisegundos tardío que tiene una respuesta.

Actualmente MD5 tiene fallos de seguridad ya que se filtró el método para poder descifrar el hash generado, lo cual lo hace un método inseguro.

Debido a las altas prestaciones a nivel de hardware de los servidores actuales, el tiempo consumido en el cifrado no tiene mucha inferencia en el rendimiento global.

IV. CONCLUSIONES.

4.1. Se realizó el análisis de las estructuras algorítmicas de Json y Json encriptado y se determinó al algoritmo RSA conveniente para la realización de las pruebas de rendimiento.

4.2. Se estableció el escenario para realizar pruebas mediante una plataforma web desarrollada en Node y JS, en la cual se puede encriptar datos, además generar claves y realizar la encriptación y monitoreo del envío de paquetes con y sin encriptación.

4.3. Se estimó el rendimiento en el intercambio de datos basado en el grado de seguridad, la fortaleza de la clave y el tiempo de velocidad en el cifrado y descifrado, mediante la definición de tamaño y generación de keys. El intercambio de datos sin encriptar nos dio un tiempo de respuesta promedio de 7ms y el encriptado de 650ms, tiendo una diferencia de 643 ms en el envío de paquetes con claves de 4098 bits.

4.4. Se comparó el rendimiento de las estructuras algorítmicas de Json y Json encriptado con RSA, determinando que Json encriptado con RSA tiene mayor seguridad, mayor fortaleza en la clave, pero mayor tiempo de ejecución en el cifrado.

REFERENCIAS

Abood, M. H. (2017). *An efficient image cryptography using hash-LSB steganography with RC4 and pixel shuffling encryption algorithms. Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), 86-90.*

Acosta Mayorga, O. E. (2015). *Aplicación móvil bajo la plataforma Android para la Gestión de calificaciones en la Unidad Educativa "Augusto Nicolas MArtinez". Ambato: Universidad Técnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/bitstream/123456789/10999/1/Tesis_t1014si.pdf*

Aguilera Lopez, P. (2010). *Seguridad Informática. Madrid: Editex.*

Aguirre, S. (2020). JSON. Buenos aires: Plandos.

Areitio, B. J. (2008). Seguridad de la Información. España: Paraninfo S.A.

Atencio Flores, D., & Mamani Machaca, D. (2017). Interconectividad basado en API Rest en aplicacones de la Municipalidad Provincial de Lapmpa. Puno: Universidad Nacional del Altiplano. Obtenido de http://repositorio.unap.edu.pe/bitstream/handle/UNAP/6163/Atencio_Flores_Dilmerd_Mamani_Machaca_David.pdf?sequence=1&isAllowed=y

Centeno, J. K., Chhabra, P. S., Fianza, C. L., Montes-Austria, I., & Ocampo, R. (2018). *Performance Analysis of Encryption Algorithms on Smartwatches. TENCON 2018 - 2018 IEEE Region 10 Conference, 0162-0166.*

Chaudhury, P., Dhang, S., Roy, M., Deb, S., Saha, J., Mallik, A., . . . Das, R. (2017). *ACAFP: Asymmetric key based cryptographic algorithm using four prime numbers to secure message communication. A review on RSA algorithm. 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), 332-337.*

Davila Torres, J. C. (2017). *Análisis y optimización del algoritmo de encriptación RIJNDAEL en el que se basa en estándar de ecriptación avanzada AES. Puno: Universidad Nacional del Altiplano. Obtenido de http://repositorio.unap.edu.pe/bitstream/handle/UNAP/4795/Davila_Torres_Juan_Carlos.pdf?sequence=1&isAllowed=y*

De pablos Heredero, C., Lopez Hermoso, J., Romo Romero, S., Medina Salgado, S., Montero Navarro, A., & Nájera Sánchez, J. (2008). *Dirección y Gestión de los Sistemas de Información en la empresa: Una Visión Integración. Madrid: ESIC Editorial.*

Gutiérrez, J., & Tena Ayuso, J. G. (2003). *protoclos criptográficos y seguridad en redes. Santander: Universidad de Cantabria.*

Harini, M., Gowri, K. P., Pavithra, C., & Selvarani, M. P. (2017). *A novel security mechanism using hybrid cryptography algorithms. IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), 1-4.*

Jones, M. (28 de 12 de 2012). *IETF Tools. Obtenido de JSON Web Algorithms (JWA): https://tools.ietf.org/id/draft-ietf-jose-json-web-algorithms-08.html*

JSON. (01 de 06 de 2020). *Obtenido de Introducing JSON: https://www.json.org/json-en.html*

Justiniano Alvarez, C. A. (2015). *Modelo de encriptación para la seguridad en redes inalámbricas Bluetooth. La Paz: Universidad Mayor de San Andrés. Obtenido de https://repositorio.umsa.bo/bitstream/handle/123456789/8752/T.3094.pdf?sequence=1&isAllowed=y*

Kaur, A., & Soni, G. (2017). *Optical stegnography to enhance speed of analog*

transmission with security enhancement through image encryption. 9th International Conference on Computational Intelligence and Communication Networks (CICN), 165-168.

Liu, T., & Wen, Y. (2018). *Studied on Application of Double Encryption Algorithm in Covert Channel Transmission. International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 210-213.*

Marrero Travieso, Y. (2017). *La Criptografía como elemento de la seguridad informática. ACIMED.*

Martínez Bohórquez, L. E. (2017). *Algoritmo para la encriptación y desencriptación entre archivos digitales de audio e imágenes. Bogotá: Universidad de San Buenaventura. Obtenido de http://bibliotecadigital.usbcali.edu.co/bitstream/10819/7198/1/Algoritmo_encriptacion_desencriptacion_Martinez_2017.pdf*

Martínez, A. (2011). *Técnicas de evaluación cualitativa y mixta. España: Universidad de Valladolid.*

Moya Caza, J. B., & Escobar Erazo, F. A. (2015). *Desarrollo de una aplicación para encriptar información en la transmisión de datos en un aplicativo de mensajería web. Quito: Pontifica Uiversidad Católica del Ecuador. Obtenido de http://repositorio.puce.edu.ec/bitstream/handle/22000/8335/Disertacion_MoyaCazaJohannaBeatriz_EscobarErazoFranklinAndres.pdf?sequence=1&isAllowed=y*

Oriol, F., & Jordi, R.-G. (01 de 06 de 2017). *EBSCO. Obtenido de Provably secure public-key encryption with conjunctive and subset keyword search.: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=138478069&lang=es&site=ehost-live.*

Oriol, F., & Jordi, R.-G. (01 de 06 de 2020). *EBSCO. Obtenido de Provably secure public-key encryption with conjunctive and subset keyword search.: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=138478069&lang=es&site=ehost-live.*

Pabón Cadavid, J. A. (2010). *La criptografía y la protección a la información digital. Revista la Propiedad Inmaterial, 59-90.*

PUSHPA, B. (2020). *Hybrid Data Encryption Algorithm for Secure Medical Data Transmission in Cloud Environment. Fourth International Conference on Computing Methodologies and Communication (ICCMC), 329-334.*

S.A. (2002). *Diccionario de Internet. Madrid: Editorial Complutense.*

Samaniego Zanabris, A. L. (2018). *Evaluación de Algoritmos Criptográficos para mejorar la Seguridad en la Comunicación y Almacenamiento de la Información. Lima: Universidad Ricardo Palma. Obtenido de http://repositorio.urp.edu.pe/bitstream/handle/URP/1509/ALSAMANIEGOZ.pdf?sequence=1&isAllowed=y*

Sampalo de la Torre, M., Leyva Cortés, E., Garzón Villar, M. L., & Prieto Tinoco, J. I. (2003). *Informática. España: MAD, S.L.*

Sanchez Vallejos, J. M. (2017). *Técnicas de encriptacon para mejorar la seguridad en la transferencia de archivos en un entorno confiable. Chiclayo: Universidad Señor de Sipán. Obtenido de http://repositorio.uss.edu.pe/bitstream/handle/uss/4060/INFORME%20DE%20TESIS_S%C3%81NCHEZ%20VALLEJOS%20JHONNYMOISES.pdf?sequence=1&isAllowed=y*

Santoso, K. I., Muin, M. A., & Mahmudi, M. A. (2019). *Implementation of AES cryptography and twofish hybrid algorithms for cloud. Journal of Physics: Conference Series, 1-6.*

Sasikumar, S., & Karthigaikumar, P. (2018). *VLSI IMPLEMENTATION OF DNA CRYPTOGRAPHY USING QUANTUM KEY EXCHANGE. 3rd International Conference on System Reliability and Safety (ICSRS), 97-103.*

Sawakinome. (2018). Sawakinome. Obtenido de Diferencia entre RSA y DSA: https://es.sawakinome.com/articles/protocols--formats/unassigned-2690.html

Seguridad, L. (julio de 2011). La seguridad. Obtenido de RSA vs. DSA para las claves de autenticación SSH: https://laseguridad.online/questions/648/rsa-vs-dsa-para-las-claves-de-autenticacion-ssh

Solichin, A., Putra, M. A., & Diniari, K. (2018). *RESTful Web Service Optimization with Compression and Encryption Algorithm. International Seminar on Application for Technology of Information and Communication, 333-337.*

Solís, F., Pinto, D., & Solís, S. (2017). *Seguridad de la información en el intercambio de datos entre dispositivos móviles con sistema Android utilizando el método de encriptación RSA. Enfoque UTE, 160-171. Obtenido de http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-65422017000100160&lng=es&nrm=iso*

tanenbaum, A. S. (2003). *Redes de computadoras. México: Pearson Educación.*

Unifica. (01 de 06 de 2020). *Unifica. Obtenido de Normativa de securización y comunicación entre aplicativos: https://ws001.sspa.juntadeandalucia.es/unifica/web/gobernanza/normativa-de-securizacion-y-comunicacion-entre-aplicativos*

ANEXOS

ANEXO Nº 01: RESOLUCIÓN DE APROBACIÓN DEL PROYECTO DE INVESTIGACIÓN

UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°1172-2021/FIAU-USS

Pimentel, 7 de diciembre de 2021

VISTO:
El oficio N° 0391-2021/FIAU-IS-USS de fecha 7 de diciembre de 2021, de la Dirección de Escuela profesional de INGENIERÍA DE SISTEMAS con el que remite el Acta de reunión N°0612-2021 del Comité de investigación de la referida Escuela profesional, acerca de la Tesis presentada por egresados del Programa de estudios de INGENIERÍA DE SISTEMAS, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y *tesis* son *aprobados por el Comité de Investigación* y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El *periodo de vigencia de los mismos será de dos años*, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; *es individual o en pares para obtener un título profesional*. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.".

Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar la ampliación de la vigencia de las tesis que se detallan en el Acta de reunión N°0612-2021, a cargo de egresados del Programa de estudios INGENIERÍA DE SISTEMAS, hasta el 3 de diciembre de 2022.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO ÚNICO: AMPLIAR VIGENCIA, de la Tesis a cargo de los estudiantes del Programa de estudios de **INGENIERÍA DE SISTEMAS** que se detallan en el anexo de la presente Resolución, hasta el 3 de diciembre de 2022.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE

Mg. Víctor Alexci Tuesta Montaza
Decano (e) / Facultad De Ingeniería, Arquitectura Y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

MBA. María Noelia Staler Rivera
Secretaria Académica / Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Cc: Interesado, Archivo

UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°1172-2021/FIAU-USS

Pimentel, 7 de diciembre de 2021

ANEXO

N°	APELLIDOS Y NOMBRES	TEMA DE TESIS	FECHA RESOLUCIÓN DE APROBACIÓN/ MODIFICACIÓN TEMA DE TESIS/AMPLIACIÓN DE VIGENCIA
1	CHAPOÑAN SANTISTEBAN DAVID	ANALISIS COMPARATIVO DE HONEYPOT HONEYD Y KFSENSOR IMPLEMENTADOS VIRTUALMENTE	Ampliación previa hasta 31 julio 2021
2	VALLEJOS DIAZ LARRY AMADOR	COMPARACIÓN DE RENDIMIENTO DE INTERCAMBIO DE DATOS CON JSON ENCRIPTADO	18 febrero 2019
3	ARROYO CABALLERO JIMMY ANDRÉ	MODELO DE GESTIÓN DE PROYECTOS Y CONTROL DE CALIDAD EN SOFTWARE PARA PYME'S PERUANAS DESARROLLADORAS DE SOFTWARE	2 julio 2019

Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

ANEXO Nº 02: METODOLOGIA SCRUM PARA EL DESARROLLO DE LA PLATAFORMA.

A. Product Backlog.

Elaboración del Product Backlog del proyecto junto a su historia de usuario.

Tabla 14.

Product Backlog.

PRODUCT BACKLOG	HISTORIA
PLATAFORMA WEB	Construir una plataforma web para comparar e rendimiento de datos con Json encriptado.

B. Construyendo el Product Backlog.

Listado de requerimientos, construidos y solicitados por el Product Owner, además de mencionar los requerimientos no funcionales.

Tabla 15.

Construyendo el Product Backlog.

PRODUCT BACKLOG
Gestionar claves
Gestionar tiempo de cifrado

Requerimientos no funcionales.

- El sistema web será desarrollado en Node y JS.
- La aplicación podrá ser abierta desde cualquier navegador.

C. Priorizando el Product Backlog.

Definición de prioridades del Product Backlog por parte del Product Owner, dando pie al punto de partida del proyecto. Se ha priorizado la gestión de claves ya que, eso permitirá la gestión de tiempo de cifrado.

Tabla 16.

Priorizando el Product Backlog.

PRODUCT BACKLOG	PRIORIDAD
Gestionar claves	1
Gestionar tiempo de cifrado	2

D. Identificando la Complejidad.

Identificación de la complejidad de los User Story para desarrollar historias de usuario agrupados por sprint.

Tabla 17.

Identificando la Complejidad.

PRODUCT BACKLOG	PRIORIDAD	COMPLEJIDAD
Gestionar claves	1	4
Gestionar tiempo de cifrado	2	5

E. Asignando el Valor por cada Story Point para cada User Story.

Mediante la aplicación de Planing Poker se realiza la determinación del User Story y asignación de los valores de esfuerzo, donde el tiempo de cifrado demanda mayor esfuerzo en la implementación debido a sus complejas funcionalidades.

Tabla 18.

Asignando el Valor por cada Story Points para cada User Story.

PRODUCT BACKLOG	PRIORIDAD	ESFUERZO
Gestionar claves	1	8
Gestionar tiempo de cifrado	2	8
TOTAL, STORY POINTS		16

F. Duración en día del Sprint.

Reuniones periódicas de avances y observaciones a presentar durante el desarrollo del producto. Debido a la celeridad del desarrollo el Sprint se presentará cada 15 días.

Tabla 19.

Duración en Días de Sprint.

SPRINT	DÍAS
	15

G. El User Story más Representativo.

Selección en base a la generación de mayor valor al proyecto.

Tabla 20.

User Story más Representativo.

PRODUCT BACKLOG	PRIORIDAD	ESFUERZO
Gestionar claves	1	8
Gestionar tiempo de cifrado	2	8
TOTAL, STORY POINTS		16

H. User Story Atendidos por un Sprint.

En función del User Story más representativo se evalúa la cantidad de User Story atendidos por sprint en base a 15 días.

Tabla 21.

User Story Atendidos por Sprint.

PRODUCT BACKLOG	ESFUERZO
Gestionar claves	8
Gestionar tiempo de cifrado	8
TOTAL, DE ESFUERZO	16

Por lo tanto, el proyecto se realizará en 15 días con una velocidad de 16 User Points.

I. Número Total de Sprint.

Calculo total de Sprint obtenido de la división entre el número total de, User Points y User Point máximo por Sprint.

Tabla 22.

Números de Sprint.

Número Total de Story Points	16
Número máximo de Story Point en un sprint	16
Número total de Sprint	1

J. Tiempo Total de Entrega.

La cantidad estimada para la realización del proyecto en días, se calcula en base al producto del número total de sprint y los días de duración.

Tabla 23.

Tiempo Total de Entrega.

Número total de Sprint	Número de días por Sprint	Total de días estimados para el proyecto
1	15	15

K. Elaboración y Agrupación de Sprint.

Identificación de los User Story por cada sprint a desarrollar.

Tabla 24.

Elaboración y Agrupación de los Sprint.

SPRINT	PRODUCT BACKLOG	ESFUERZO
Sprint 1	Gestionar claves	8
	Gestionar tiempo de cifrado	8

L. Desarrollo de Sprint.

Sprint 1.

I. Etapa de Planificación.

a. Ajustes del Sprint Anterior.

Por ser el primer Sprint no se requiere ajustes, pero si se requiere detallar las herramientas a usar, como Node, JS, etc.

b. Pila el Sprint.

Tabla 25.

Pila de sprint 1.

SPRINT	PRODUCT BACKLOG
Sprint 1	Gestionar claves
	Gestionar tiempo de cifrado

c. Historias de Usuario.

Tabla 26.

Historia de Usuario de Gestionar Claves.

HISTORIA DE USUARIO			
Número	1	**Usuario**	Cliente
Nombre de historia	Gestionar claves	**Riego en desarrollo**	Bajo
Prioridad en negocio	Alta	**Sprint asignada**	1
Puntos estimados	5	**Dependencia para su desarrollo**	Ninguno
Programador responsable	Larry Vallejos		
Descripción	Yo como cliente quiero generar claves para encriptar y desencriptar datos.		
Criterio de aceptación	- Se creará una interfaz gráfica en la web - Se debe mostrar el contenido cifrado en un cuadro de texto. - La desencriptación y muestra de debe mostrar en un cuadro de texto alterno.		

Tabla 27.

Historia de Usuario de Gestionar Tiempo de Cifrado.

HISTORIA DE USUARIO			
Número	2	**Usuario**	Cliente
Nombre de historia	Gestión de tiempo de cifrado	**Riego en desarrollo**	Bajo
Prioridad en negocio	Alta	**Sprint asignada**	1
Puntos estimados	5	**Dependencia para su desarrollo**	Gestión de claves
Programador responsable	Yo como cliente quiero medir el tiempo del envío de la petición y el tiempo de respuesta del servidor con el contenido desencriptado.		
Descripción	Yo como cliente quiero		
Criterio de aceptación	- Se creará una interfaz gráfica en la web. - Se debe mostrar el tiempo de envío de la petición y el tiempo de respuesta del servidor con el contenido desencriptado. - Se debe mostrar la medición en un cuadro gráfico.		

d. Tareas de Usuario.

Tabla 28.

Tarea de Usuario Crear Generate Key, Encrypt y Decrypt.

Nº de Tarea	1.1	**Nº Historia de Usuario**	1
Nombre de la Tarea	Crear generate key, encrypt y decrypt.	**Tiempo Estimado en Horas**	4
Fecha Inicio	25/05/2020	**Fecha Fin**	27/05/2018
Responsable	Larry Vallejos		
Descripción	a. El usuario debe poder usar diversos tamaños de bits para le key. b. El usuario debe poder genera la clave pública y privada. c. El usuario debe poder encriptar y desencriptar texto.		

Tabla 29.

Tarea de Usuario Realizar Interfaz Gráfica de Gestión de Claves.

Nº de Tarea	1.2	**Nº Historia de Usuario**	1
Nombre de la Tarea	Realizar interfaz gráfica de gestión de claves	**Tiempo Estimado en Horas**	4
Fecha Inicio	1/06/2020	**Fecha Fin**	5/062018
Responsable	Larry Vallejos		
Descripción	Consiste en diseñar e implementar el formulario necesario para la gestión adecuada de las claves de encriptación y desencriptación.		

Tabla 30.

Tarea de Usuario Crear Carga de Rendimiento y Envío de Tramas.

Nº de Tarea	2.1	**Nº Historia de Usuario**	2
Nombre de la Tarea	Crear carga de rendimiento y envío de tramas.	**Tiempo Estimado en Horas**	4

Fecha Inicio	8/06/2020	**Fecha Fin**	12/062018
Responsable	Larry Vallejos		
Descripción	a. El usuario debe visualizar le tiempo generado por el envío de paquetes. b. El usuario debe visualizar la gráfica de envío de paquetes encriptados con RSA y sin RSA.		

Tabla 31.

Tarea de Usuario Realizar Interfaz Gráfica de Gestión de Tiempo de Cifrado.

Nº de Tarea	2.2	**Nº Historia de Usuario**	2
Nombre de la Tarea	Realizar interfaz gráfica de gestión de tiempo de cifrado	**Tiempo Estimado en Horas**	4
Fecha Inicio	16/06/2020	**Fecha Fin**	20/062018
Responsable	Larry Vallejos		
Descripción	Consiste en diseñar e implementar el formulario necesario para la gestión adecuada del tiempo de cifrado.		

II. Etapa de Diseño

A. Diseño de Interfaz de Usuario.

- **Gestión de Claves**

  Boceto de interfaz para la gestión de claves.

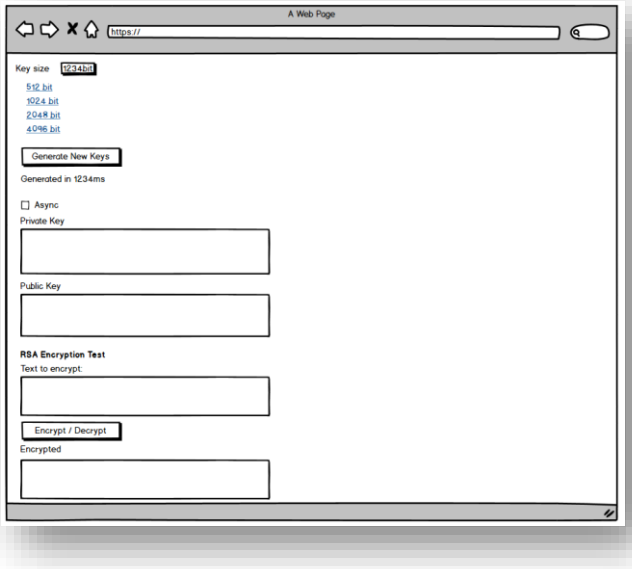


Figura 23. Boceto Interfaz de Gestión de Claves.

- **Gestión de Tiempo de Cifrado.**

  Boceto de interfaz para la gestión de tiempo de cifrado

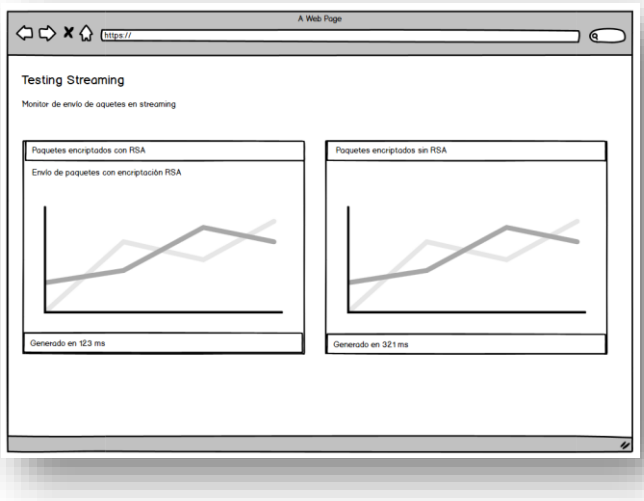


Figura 24. Boceto Interfaz de Gestión de Tiempo de Cifrado.

III. Etapa de Desarrollo.

A. Diseño de Interfaz de Usuario.

- **Gestión de Claves.**

  Con esta interfaz se puede generar las claves de encriptación en base a los bits seleccionados.

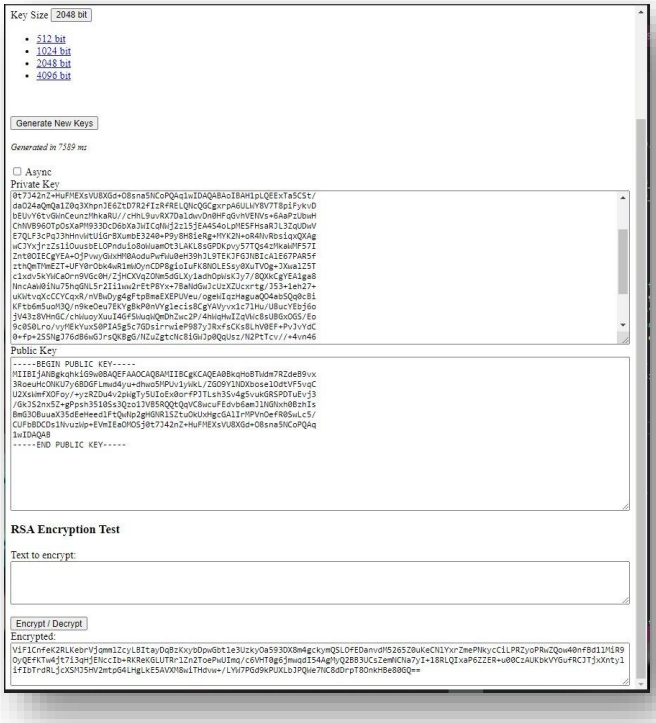


Figura 25. Interfaz de Gestionar Claves*.*

- **Gestión de Tiempo de Cifrado.**

  Con esta interfaz se puede visualizar el monitoreo y comparación de envío de paquetes.

Figura 26. Interfaz de Gestionar Tiempo de Cifrado.

ANEXO Nº 03: ENTREVISTA DE SOFTWARE

OBJETIVO	Conocer cantidad aproximada de ficheros Json por sistema.

UNIVERSIDAD SEÑOR DE SIPÁN

Facultad de Ingeniería, Arquitectura y Urbanismo

Escuela Profesional de Ingeniería de Sistemas

ENTREVISTA DE SOFTWARE

Nombre: Ingeniero del Área de División de Tecnólogas de la Información de una entidad de Lambayeque.

Fecha: 05/06/2020

Cuestionario:

A. ¿Cuál es tu actividad?

Desarrollador de Software

B. ¿Cuántos sistemas han desarrollado en la entidad?

12 hasta hoy, entre propios y terciarizados.

C. ¿Sus sistemas contienen ficheros Json?

Si, solo algunos

D. ¿Hasta cuántos ficheros Json pueden contener aproximadamente sus sistemas?

Nuestros sistemas contienen aproximadamente:

Sistema 1 = 500 ficheros Json.

Sistema 2 = 400 ficheros Json.

Sistema 3 = 300 ficheros Json.

Sistema 4 = 450 ficheros Json.

Sistema 5 = 350 ficheros Json.

Sistema 6 = 600 ficheros Json.

Sistema 7 = 350 ficheros Json.